



HAL
open science

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

Jérémy Berthomieu, Jean-Charles Faugère, Ludovic Perret

► **To cite this version:**

Jérémy Berthomieu, Jean-Charles Faugère, Ludovic Perret. Polynomial-Time Algorithms for Quadratic Isomorphism of Polynomials: The Regular Case. 2014. hal-00846041v4

HAL Id: hal-00846041

<https://inria.hal.science/hal-00846041v4>

Preprint submitted on 4 Apr 2014 (v4), last revised 21 Apr 2015 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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}

^aSorbonnes Université, UPMC Univ Paris 06, Équipe POLSYS, LIP6, F-75005, Paris, France

^bCNRS, UMR 7606, LIP6, F-75005, Paris, France

^cINRIA, Centre Paris – Rocquencourt, Équipe POLSYS, F-75005, Paris, France

Abstract

Let $\mathbf{f} = (f_1, \dots, f_m)$ and $\mathbf{g} = (g_1, \dots, g_m)$ be two sets of $m \geq 1$ nonlinear polynomials over $\mathbb{K}[x_1, \dots, x_n]$ (\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. The main result is a randomized polynomial-time algorithm for solving IP1S for quadratic instances, a particular case of importance in cryptography and somewhat justifying *a posteriori* the fact that *Graph Isomorphism* reduces to only cubic instances of IP1S (Agrawal and Saxena). 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, which involves to test the orthogonal simultaneous conjugacy of symmetric matrices. 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 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 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 the square root in $\mathbb{K}^{n \times n}$ for various fields (including finite fields). 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 solving IP when $\mathbf{f} = (x_1^d, \dots, x_n^d)$ is presented. From an algorithmic point of view, the problem boils down

[☆]This work has been partly supported by the French ANR-11-BS02-0013 HPAC project.

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

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:

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 varieties of fundamental problem in computer science, ranging – among others topics – from cryptography (*e.g.* Patarin (1996a,b); Tang and Xu (2012, 2014); Yang et al. (2011)), arithmetic complexity (*via* Geometric Complexity Theory (GCT) for instance, *e.g.* 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 (1996a,b), 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 varies 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 for 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 (1996a)). 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 (1996a,b)) as *Isomorphism of Polynomials* (`IP` or `IP2S`). Unlike `IP1S`, there is no lower bound on the computational complexity of `IP`, *i.e.* no known¹ reduction from `GI` to `IP` for instance. `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 (2012) showed that `PolyProj` is NP-Hard. This is maybe 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 $\mathbb{K}[x_1, \dots, x_n]$. The Isomorphism of Polynomials (`IP`) problem, introduced by Patarin (Patarin (1996a,b)), is as follows:

Isomorphism of Polynomials (`IP`)

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

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

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

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),

¹To be more precise, there is no lower bound for `IP` when $m > 1$. If $m = 1$, `IP` degenerates to `PolyEquiv`.

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. 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. We take a first step in this program by considering IP for a specific set of polynomials. In Section 5, we prove the following:

Theorem 1. *Let $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{K}[x_1, \dots, x_n]^n$. Let $d > 0$ be an integer, and define $\mathbf{POW}_{n,d} = (x_1^d, \dots, x_n^d) \in \mathbb{K}[x_1, \dots, x_n]^n$. Assuming \mathbb{K} is big enough, there is a randomized polynomial-time algorithm which recovers – if any – $(A, B) \in \text{GL}_n(\mathbb{K}) \times \text{GL}_n(\mathbb{K})$ 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)).

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 6). 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 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. (2011a), 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 $m > 1$.

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

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

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

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 $(\mathbf{f} = (f_1, \dots, f_m), \mathbf{g} = (g_1, \dots, g_m)) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$ be homogeneous quadratic polynomials. Let H_1, \dots, H_m be the Hessian matrices of f_1, \dots, f_m (resp. H'_1, \dots, H'_m be the Hessian matrices of g_1, \dots, 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 $A \in \text{GL}_n(\mathbb{K})$ such that:

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

Combining two equations of this form yields $H'_j H_i^{-1} = A^{-1} \cdot H_j^{-1} H_i \cdot A$ if H_j is invertible. 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 $\mathbf{f} = (f_1, \dots, f_m) \in \mathbb{K}[x_1, \dots, x_n]$. We assume that a linear combination of the quadratic forms f_1, \dots, f_m is not degenerate².

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 2. *Let \mathbb{K} be a field of 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*

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

²We would like to thank G. Ivanyos for having pointed us this issue in a preliminary version of this paper.

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 $\mathbb{L}[x_1, \dots, x_n]$ such that \mathbb{L} is an algebraic extension of \mathbb{K} of odd degree $O(\log n)$. Any solution on \mathbb{L} on the canonical form can be efficiently mapped to a solution of the initial instance (and conversely).

Note that the success probability of the algorithms presented here will depend of the size of the field. To amplify the success probability over a small field, we will use the fact that matrices are conjugate over \mathbb{K} if, and only if, they are conjugate over an algebraic extension \mathbb{L} (see Pazzis (2010)). Thus, we will 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 $A^T D A = D$ with $D = \text{Diag}(d_1, \dots, d_n)$ and $H'_i = A^T \cdot H_i \cdot A = D A^{-1} D^{-1} \cdot H_i \cdot A$ 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 $\{H_1, \dots, H_m\}$ and $\{H'_1, \dots, H'_m\}$ 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. $X^T D X = D$, with \mathbb{L} being an algebraic extension of \mathbb{K} , such that:

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

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

- i. 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 $Y \in \text{GL}_n(\mathbb{K})$ such that:

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

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

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

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

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

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

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:

$$H_1 \cdot Y = Y \cdot H_1', \dots, H_m \cdot Y = Y \cdot H_m' \text{ and } \det(Y) \neq 0. \quad (4)$$

From now on, we shall denote by $\mathcal{O}_n(\mathbb{L}, D)$ 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 the Edmonds' problem (Edmonds (1967)). Note that, if the representation of the group generated 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 the chosen extension field \mathbb{L} of \mathbb{K} is big enough. As already explained, if the cardinality of \mathbb{L} is too small we can amplify the probability of success by considering a bigger algebraic extension (see 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 $H_1, \dots, H_m, H_1', \dots, H_m'$ to quadratic forms can be chosen as upper triangular. In this context, we show in Section 3.5 that IP1S can still be reduced to a $(H_1 + H_1^T)$ -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 3.6. 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. 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 3. *Let \mathbb{K} be a field of 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, its 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 3 can be turned into a deterministic algorithm solving IP1S over the base field \mathbb{K} or an extension thereof. That is:

Theorem 4. *Under Assumption 1, 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 verifying equations (4) has dimension 1, then the algorithm can solve quadratic-IP1S over \mathbb{K} .*

If $m \geq 3$, for generic matrices H_1, \dots, 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 here, by generic, we mean that there is no nontrivial vector subspace W of \mathbb{K}^n such that $H_i(W) = H_1(W)$ for all i , $2 \leq i \leq m$, see Schur’s Lemma 17 and Lemma 18. In Section 3.6, 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 2). 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 $(\mathbf{f} = (f_1, \dots, f_m), \mathbf{g} = (g_1, \dots, g_m)) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$, we want to count the number of invertible matrices $A \in GL_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{f}(A \cdot \mathbf{x})$. To do so, we define:

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

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

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 $\mathcal{C}(\mathcal{H})$ of the Hessian matrices \mathcal{H} associated to \mathbf{f} . We prove the following structural results for $\mathcal{C}(\mathcal{H})$:

Proposition 5. *Let α be an algebraic element of degree m over \mathbb{K} . Let $H = \sum_{i=1}^m H_i \alpha^{i-1} \in \mathbb{K}(\alpha)^{n \times n}$ be a matrix and let J be its Jordan normal form. Assuming that blocks associated with eigenvalue η_1 have sizes $s_{1,1} \leq \dots \leq s_{1,d_1}, \dots$, blocks associated with eigenvalue η_r have sizes $s_{r,1} \leq \dots \leq s_{r,d_r}$, then the centralizer of \mathcal{H} is a \mathbb{K} -vector subspace of $\mathbb{K}^{n \times n}$. Its dimension is bounded from above by:*

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

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, \dots, 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, \dots, H_m does not contain a nondegenerate matrix. This situation is well illustrated by the following example $f_1 = x_1x_3, f_2 = x_2x_3$. 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. (2011a); Macario-Rat et al. (2013)) will fail on these instances; making the hardness of the irregular case intriguing and then an interesting open question.

2. Normalization - Canonical form of IP1S

In this section, we prove Theorem 2. In other words, we explain how to reduce, under Assumption 1, any quadratic homogeneous instance $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$ 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 a same diagonal invertible matrix. We emphasize that the reduction presented is randomized and requires to consider an algebraic extension of limited degree.

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 $p^*(x_0, x_1, \dots, x_n) = x_0^2 p(x_1/x_0, \dots, x_n/x_0)$ its *homogenization*.

Proposition 6. *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 $(\mathbf{f}^* = (f_0^* = x_0^2, f_1^*, \dots, f_m^*), \mathbf{g}^* = (g_0^* = x_0^2, g_1^*, \dots, g_m^*))$. This clearly transforms polynomials of degree 2 to homogeneous quadratic polynomials. We can write $f_i(\mathbf{x}) = \mathbf{x}^T H_i \mathbf{x} + L_i \mathbf{x} + c_i$ with $H_i \in \mathbb{K}^{n \times n}$, $L_i \in \mathbb{K}^n$ and $c_i \in \mathbb{K}$, then $f_i(A\mathbf{x} + b) = (A\mathbf{x} + b)^T H_i (A\mathbf{x} + b) + L_i (A\mathbf{x} + b) + c_i$ and its homogenization is $(A\mathbf{x} + bx_0)^T H_i (A\mathbf{x} + bx_0) + L_i (A\mathbf{x} + bx_0) + c_i x_0^2 = f_i^*(A'\mathbf{x}^*)$, with $\mathbf{x}^* = (x_0, x_1, \dots, x_n)^T$. If $(A, b) \in \text{GL}_n(\mathbb{K}) \times \mathbb{K}^n$ is an affine transformation solution on the inhomogeneous instance then $A' = \begin{pmatrix} 1 & \mathbf{0} \\ b & A \end{pmatrix}$ is a solution for the homogenized instance. Conversely, a solution $A' \in \text{GL}_{n+1}(\mathbb{K})$ 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 $\begin{pmatrix} 1 & \mathbf{0} \\ d & C \end{pmatrix}$, and $(C, d) \in \text{GL}_n(\mathbb{K}) \times \mathbb{K}^n$ is a solution for (\mathbf{f}, \mathbf{g}) . \square

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 7. *(Carlini (2005); Kayal (2011)) Let $f \in \mathbb{K}[x_1, \dots, x_n]$ be a polynomial. We shall say that f has s essential variables if $\exists M \in \text{GL}_n(\mathbb{K})$ such that $f(M\mathbf{x})$ depends only of the first s variables x_1, \dots, x_s . The remaining $n - s$ variables x_{s+1}, \dots, x_n will be called *redundant variables*. If $\text{char } \mathbb{K} = 0$ or $\text{char } \mathbb{K} > \deg f$, and f has s essential variables, then we can compute in randomized polynomial time $M \in \text{GL}_n(\mathbb{K})$ such that $f(M\mathbf{x})$ depends only of the first s variables.*

For a set of equations, we extend the notion of essential variables as follows.

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

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

with ℓ_1, \dots, ℓ_s being linear forms in x_1, \dots, x_n of rank s and $\tilde{\mathbf{f}} \in \mathbb{K}[y_1, \dots, y_s]^m$.

For a quadratic form, s is simply the rank of the associated Hessian matrix. The linear forms ℓ_1, \dots, ℓ_s can be easily computed thanks to Proposition 7 when the characteristic of \mathbb{K} is zero or greater than the degrees of f_1, \dots, 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 8. *Let $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$. 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 $(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) \in \mathbb{K}[y_1, \dots, y_s]^m \times \mathbb{K}[y_1, \dots, y_s]^m$ be such that:*

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

with ℓ_1, \dots, ℓ_s (resp. ℓ'_1, \dots, ℓ'_s) linear forms in x_1, \dots, x_n of rank s and $\tilde{\mathbf{f}}, \tilde{\mathbf{g}} \in \mathbb{K}[y_1, \dots, y_s]^m$. It holds that:

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

Proof. Let H_1, \dots, H_m be the Hessian matrices associated of f_1, \dots, f_m (resp. H'_1, \dots, H'_m be the Hessian matrices of g_1, \dots, g_m). Similarly, we define the Hessian matrices $\tilde{H}_1, \dots, \tilde{H}_m$ (resp. $\tilde{H}'_1, \dots, \tilde{H}'_m$) associated to $\tilde{f}_1, \dots, \tilde{f}_m$ (resp. $\tilde{g}_1, \dots, \tilde{g}_m$). Let also M and N be matrices in $\mathbb{K}^{n \times n}$ such that $H_i = M^T \begin{pmatrix} \tilde{H}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} M$ and $H'_i = N^T \begin{pmatrix} \tilde{H}'_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} N$ 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 $M = \begin{pmatrix} M_1 & M_2 \\ \mathbf{0} & \text{Id} \end{pmatrix}$ and $N = \begin{pmatrix} N_1 & N_2 \\ \mathbf{0} & \text{Id} \end{pmatrix}$, with $M_1, N_1 \in \text{GL}_s(\mathbb{K})$. If $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$, $\exists \tilde{A} \in \text{GL}_s(\mathbb{K})$ such that $A^T \tilde{H}_i \tilde{A} = \tilde{H}'_i$, for all $i, 1 \leq i \leq m$. Then, for all $B \in \mathbb{K}^{(n-s) \times s}$ and $C \in \text{GL}_{n-s}(\mathbb{K})$:

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

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

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

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

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 $\text{rank } \tilde{A}$ cannot be smaller than s , hence $\tilde{A} \in \text{GL}_s(\mathbb{K})$. We then get $\tilde{A}^T \tilde{H}_i \tilde{A} = \tilde{H}'_i$ for all $i, 1 \leq i \leq m$, i.e. $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$. \square

According to Proposition 8, 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 $\text{char } \mathbb{K} \neq 2$.

Definition 5. Let $\mathbf{f} = (f_1, \dots, f_m) \in \mathbb{K}[x_1, \dots, x_n]$ be quadratic homogeneous forms with Hessian matrices H_1, \dots, H_m . We shall say that \mathbf{f} is *regular* if its number of essential variables is n and if $\det(\sum_{i=1}^m \lambda_i H_i)$ is not identically zero.

Remark 9. 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 have the following result. We are now in position to reduce quadratic homogeneous instances of IP1S to a first simplified form.

Proposition 10. *Let $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$ be regular quadratic homogeneous polynomials. There is a randomized polynomial-time algorithm which returns "NoSolution" only if $\mathbf{f} \not\sim \mathbf{g}$, or a new instance*

$$(\tilde{\mathbf{f}}, \tilde{\mathbf{g}}) = \left(\left(\sum_{i=1}^n d_i x_i^2, \tilde{f}_2, \dots, \tilde{f}_m \right), \left(\sum_{i=1}^n d_i x_i^2, \tilde{g}_2, \dots, \tilde{g}_m \right) \right) \in \mathbb{L}[x_1, \dots, x_n]^m \times \mathbb{L}[x_1, \dots, x_n]^m,$$

with d_1, \dots, d_n being 1 or nonsquares in \mathbb{L} , such that $\mathbf{f} \sim \mathbf{g} \iff \tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$. If \mathbb{K} is infinite $\mathbb{L} = \mathbb{K}$, otherwise \mathbb{L} is an algebraic extension of \mathbb{K} of degree $2 \left\lceil \frac{\log_{|\mathbb{K}|}(n+1)}{2} \right\rceil + 1$. In the latter case, the output of this algorithm is correct with probability at least $1 - n/|\mathbb{L}|$. If $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$, invertible matrices P, Q and $A' \in \text{GL}_n(\mathbb{L})$ 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, \dots, H_m be the Hessian matrices associated to f_1, \dots, f_m . According to Schwartz-Zippel-DeMillo-Lipton Lemma (DeMillo and Lipton (1978); Zippel (1979)), we can compute in randomized polynomial time $\lambda_1, \dots, \lambda_m \in \mathbb{L}$ such that $\varphi = \sum_{i=1}^m \lambda_i \cdot f_i$ has n essential variables, i.e. $\det(\sum_{i=1}^m \lambda_i H_i) \neq 0$. The probability to pick $(\lambda_1, \dots, \lambda_m) \in \mathbb{L}^m$ on which φ has not n essential variables is bounded from above by $n/|\mathbb{L}|$. We define $\gamma = \sum_{i=1}^m \lambda_i \cdot g_i$. Should one reorder the equations, we can assume w.l.o.g. that $\lambda_1 \neq 0$. We have then:

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

Now, applying Gauß's reduction algorithm to φ , there exists $d_1, \dots, d_n \in \mathbb{K}$, each being 1 or a nonsquare, such that $\varphi = \sum_{i=1}^n d_i \ell_i^2$, where ℓ_1, \dots, ℓ_n are independent linear forms in x_1, \dots, x_n . This gives a $P \in \text{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{f}} = (\tilde{\varphi} = \sum_{i=1}^n d_i x_i^2, \tilde{f}_2, \dots, \tilde{f}_m) = (\varphi(P\mathbf{x}), f_2(P\mathbf{x}), \dots, f_m(P\mathbf{x}))$. 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 $\mathbf{f} \not\sim \mathbf{g}$ and we return "NOSOLUTION". Otherwise, the reduction is given by a matrix $Q \in \text{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{g}} = (\tilde{\gamma} = \sum_{i=1}^n d_i x_i^2, \tilde{g}_2, \dots, \tilde{g}_m) = (\gamma(Q\mathbf{x}), g_2(Q\mathbf{x}), \dots, g_m(Q\mathbf{x}))$ and $\mathbf{g} \sim \tilde{\mathbf{g}}$. Thus, $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}}$ if, and onl, if, $\mathbf{f} \sim \mathbf{g}$. Now, assume that $\exists A' \in \text{GL}_n(\mathbb{L})$ 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.* $\mathbf{f}(PA'Q^{-1}\mathbf{x}) = \mathbf{g}(\mathbf{x})$. \square

Remark 11. Whenever \mathbb{K} is finite, one has to be careful with the extension field \mathbb{L} . Indeed, let us assume that there exist $\lambda_1, \dots, \lambda_m \in \mathbb{K}$, such that $\varphi = \sum_{i=1}^m \lambda_i f_i$ and $\gamma = \sum_{i=1}^m \lambda_i g_i$ have n essential variables but $|\mathbb{K}| \leq n$. Let us assume furthermore that Gauß's reduction algorithm applied to φ and γ over \mathbb{K} yields $\varphi = \sum_{i=0}^{n-1} \ell_i^2 + \ell_n^2$ and $\gamma = \sum_{i=0}^{n-1} \ell_i'^2 + \alpha \ell_n'^2$ with α not a square in \mathbb{K} , see (Lidl and Niederreiter, 1997, Theorem 6.21). Then $\varphi \not\sim \gamma$ and $\mathbf{f} \not\sim \mathbf{g}$.

Since the size of \mathbb{K} is too small, then we cannot find such $\lambda_1, \dots, \lambda_m$ easily and we still have to amplify the size of the field by taking a proper extension \mathbb{L} . However, if \mathbb{L} has even degree over \mathbb{K} , then α is now a square in \mathbb{L} and Gauß's reduction algorithm applied to γ will yield $\gamma = \sum_{i=0}^{n-1} \ell_i'^2 + (\sqrt{\alpha} \ell_n')^2$ and the algorithm will miss to directly return "NOSOLUTION". Fortunately enough, if \mathbb{L} has odd degree over \mathbb{K} , then any nonsquare in \mathbb{K} is still a nonsquare in \mathbb{L} . This is our motivation to only considering \mathbb{L} of odd degree over \mathbb{K} .

2.iv. Invertible Hessian Matrices. We are now in 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 have n essential variables assuming we could find one. From Proposition 10, 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 $H_i - \nu_i D = H_i - \nu_i H_1$, with ν_i not an eigenvalue of $H_i D^{-1}$. As \mathbb{L} has at least $n + 1$ elements, there exists such a ν_i in \mathbb{L} . This gives the following result:

Theorem 12. *Let $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$ be regular quadratic homogeneous polynomials. There is a randomized polynomial-time algorithm which returns "NOSOLUTION" only if $\mathbf{f} \not\sim \mathbf{g}$. Otherwise, the algorithm returns two sets of $n \times n$ invertible symmetric matrices $\{D, \tilde{H}_2, \dots, \tilde{H}_m\}$ and $\{D, \tilde{H}'_2, \dots, \tilde{H}'_m\}$, with D diagonal, defined over an algebraic extension \mathbb{L} of \mathbb{K} of odd degree $O(\log n)$ such that:*

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}(A\mathbf{x}), \text{ for } A \in \text{GL}_n(\mathbb{K}) \iff A'^{-1}D^{-1}\tilde{H}_i A' = D^{-1}\tilde{H}'_i, \forall i, 1 \leq i \leq m, \text{ for } A' \in \mathcal{O}_n(\mathbb{L}, D),$$

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

Proof. Combining Proposition 10 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

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

where all the polynomials are *nondegenerate* homogeneous quadratic polynomials in $\mathbb{L}[x_1, \dots, x_n]$ where \mathbb{L} is \mathbb{K} whenever \mathbb{K} is infinite or an algebraic extension of \mathbb{K} of odd degree at most $2 \left\lceil \frac{\log_{|\mathbb{K}|}(n+1)}{2} \right\rceil + 1$. It follows that $\tilde{\mathbf{f}} \sim \tilde{\mathbf{g}} \iff \exists A' \in \text{GL}_n(\mathbb{L})$ such that $\forall i, 1 \leq i \leq m$, $A'^T \tilde{H}_i A' = \tilde{H}'_i$. In particular $A'^T D A' = D$ and A' is D -orthogonal. Hence, $A'^T \tilde{H}_i A' = D A'^{-1} D^{-1} \tilde{H}_i A' = \tilde{H}'_i, \forall i, 1 \leq i \leq m$. \square

The proof of this result implies Theorem 2.

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 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 $O(nM(n) \log(qn))$ operations in \mathbb{F} , see (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 $2 \leq \omega \leq 3$.

2.vi. Remark. In this normalization process, a field extension can occur in paragraph 2.iii, to have H_1 and H'_1 invertible. Let us notice that if one of the Hessian matrix of \mathcal{H} (hence one of \mathcal{H}') is invertible, then this field extension is not necessary. Indeed, we use this field extension twice. First, to ensure that one can replace H_1 and H'_1 by an invertible matrix. Then, to replace each H_i and $H'_i, 1 \leq i \leq m$ by invertible matrices in paragraph 2.iv. Actually, this last step is useless as Chistov et al. (1997) does not assume that the matrices are invertible.

3. Quadratic IP1S

In this section, we present efficient algorithms for solving regular quadratic-IP1S. According to Proposition 6, we can w.l.o.g. restrict our attention on linear changes of variables and homogeneous quadratic instances. Let \mathbb{L} be an algebraic extension of \mathbb{K} of odd degree $O(\log n)$ and let D be a diagonal invertible matrix with 1 or nonsquare elements on the diagonal. Let $\mathcal{H} = \{D, H_2, \dots, H_m\}$ and $\mathcal{H}' = \{D, H'_2, \dots, H'_m\}$ be two families of invertible symmetric matrices in $\mathbb{L}^{n \times n}$. As explained in Theorem 12, our task reduces – under a randomized process – to finding a D -orthogonal matrix $A' \in \mathcal{O}_n(\mathbb{L}, D)$ such that:

$$A'^{-1} D^{-1} H_i A' = D^{-1} H'_i, \forall i, 1 \leq i \leq m. \quad (5)$$

Case $D = \text{Id}$ 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 $H_i^T Y = Y H_i'^T$. In our case, whenever $D = \text{Id}$, the matrices are symmetric. So, the added conditions – with the transpose – are automatically fulfilled. Chistov et al. (1997) 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 $Z = Y^T Y$ as stated in (Chistov et al., 1997, Section 3). However, in general W and A are not defined over \mathbb{L} but over $\mathbb{M} = \mathbb{L}(\sqrt{\zeta_1}, \dots, \sqrt{\zeta_r})$, where ζ_1, \dots, ζ_r are the eigenvalues of Z . Assuming ζ_1 is the root of an irreducible polynomial P of degree d , then ζ_2, \dots, ζ_d are also roots of the same polynomial. However, there is no reason for them to be in $\mathbb{L}[x]/(P) = \mathbb{L}(\zeta_1)$. But they will be the roots of a polynomial of degree $d-1$, in general, over the field $\mathbb{L}(\zeta_1)$. Then, doing another extension might only add one eigenvalue in the field. Repeating this process yields a field of degree $d!$ over \mathbb{L} . 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 $W = \sqrt{Y^T Y}$ 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. However, this issue does not happen if one applies a small trick to their proof and sets $W = \sqrt{Y Y^T}$ 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.* $W^T = D W D^{-1}$. Let us notice that A and W might be defined only over \mathbb{M} an algebraic extension of \mathbb{L} of some degree.

Proposition 13. *Let $\mathcal{K} = \{K_1, \dots, K_m\}$ and $\mathcal{K}' = \{K'_1, \dots, K'_m\}$ be two subsets of m matrices in $\mathbb{L}^{n \times n}$. Let D be an invertible diagonal matrix. There is a D -orthogonal solution $A \in \mathbb{M}^{n \times n}$ 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 $Y \in \mathbb{M}^{n \times n}$ to the conjugacy problem $K_i Y = Y K_i'$ and $K_i^T D Y D^{-1} = D Y D^{-1} K_i'^T$ 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 $A^T = D A^{-1} D^{-1}$, it is clear that A is

a solution to the second problem. Conversely, let Y be a solution to the second problem, then $Z = D^{-1}Y^T D Y$ 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 $W^T = D W D^{-1}$, 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, $A = Y W^{-1}$ is an D -orthogonal solution of the first problem. As W commutes with K'_i , $A^{-1}K_i A = W Y^{-1}K_i Y W^{-1} = W K'_i W^{-1} = K'_i$ and

$$A^T D A = W^{-T} Y^T D Y W^{-1} = D W^{-1} D^{-1} Y^T D Y W^{-1} = D W^{-T} Z W^{-1} = D. \quad \square$$

As one can remark, the equations that one needs to add in Proposition 13 are in fact automatically verified in our case with $D = H_1 = H'_1$, $K_i = H_1^{-1}H_i$ and $K'_i = H_1^{-1}H'_i$ for all $1 \leq i \leq m$. Indeed, as H_i and H'_i are symmetric, then one has $K_i^T D Y D^{-1} = H_i H_1^{-1} H_1 Y H_1^{-1} = H_i Y H_1^{-1}$ and $D Y D^{-1} K_i^T = H_1 Y H_1^{-1} H'_i H_1^{-1}$. Thus, an equation of the second set reduces to the equation $H_1^{-1}H_i Y = Y H_1^{-1}H'_i$, and thus $K_i Y = Y K'_i$.

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 $Y \in \text{GL}_n(\mathbb{L})$ of equation 5. Instead of computing exactly $A \in \mathcal{O}_n(\mathbb{M}, D)$, 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 conjugate for all $i, 1 \leq i \leq m$. To be more precise, we prove the following proposition.

Proposition 14. *Let $\mathcal{H} = \{H_1 = D, H_2, \dots, H_m\}$ and $\mathcal{H}' = \{H'_1 = D, H'_2, \dots, H'_m\}$ be two sets of invertible matrices in $\mathbb{L}^{n \times n}$. We can compute in polynomial time two matrices S and T defined over an algebraic extension \mathbb{M} 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{M} . However, matrices $S^T H_i S$ and $T^T H'_i T$ can be computed and tested for equality in polynomial time.*

Proof. Let $Y \in \text{GL}_n(\mathbb{L})$ such that $D^{-1}H_i Y = Y D^{-1}H'_i, \forall i, 1 \leq i \leq m$. We set $Z = D^{-1}Y^T D Y$. Let us denote by T , the change of basis matrix such that $J = T^{-1}Z T$ 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 $\mathbb{L}(\zeta_i)$, where ζ_1, \dots, ζ_r are the eigenvalues of Z .

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

As $J = T^{-1}ZT$ is a Jordan normal form, it is block-wise defined over $\mathbb{L}(\zeta)$. 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 $\mathbb{L}(\sqrt{\zeta})$, hence 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 $\mathbb{L}(\zeta)$, see (Cai, 1994, Section 4). Since Y is defined over \mathbb{L} , then YT is columnblock-wise defined over $\mathbb{L}(\zeta)$. Thus $S = AT = YW^{-1}T = YTG^{-1}$ is a columnblock-wise defined over $\mathbb{L}(\sqrt{\zeta})$. We recall that product $AT \cdot T^{-1}$ mangles the eigenvalues and make each coefficients defined over $\mathbb{L}(\sqrt{\zeta_1}, \dots, \sqrt{\zeta_r})$ and thus must be avoided.

Now, to verify that $A^T H A = H'$, for any $H \in \mathcal{H}$ and the corresponding $H' \in \mathcal{H}'$, we compute separately $S^T H S = T^T A^T H A T$ and $T^T H' T$. For the former, $S = AT$ (resp. $S^T = (AT)^T$) is columnblock-wise (resp. rowblock-wise) defined over $\mathbb{L}(\sqrt{\zeta})$ and H is defined over \mathbb{L} . Therefore, the product matrix has its coefficients which are on both the i th block of rows and the j th block of columns defined over $\mathbb{L}(\sqrt{\zeta_i}, \sqrt{\zeta_j})$ 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 $\mathbb{L}(\zeta)$.

Let us assume that the characteristic polynomial of Z , of degree n , can be factored as $P_1^{e_1} \dots P_s^{e_s}$ 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 $p > 2$. \square

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 $\mathcal{H} = \{H_1 = D, \dots, H_m\} \subseteq \mathbb{L}^{n \times n}$ and $\mathcal{H}' = \{H'_1 = D, \dots, H'_m\} \subseteq \mathbb{L}^{n \times n}$.

Output A description of the matrix $A \in \text{GL}_n(\mathbb{M})$ such that $H'_i = A^T H_i A$ for all $1 \leq i \leq m$ or “NOSOLUTION”.

Compute the vector subspace $\mathcal{Y} = \{Y \mid D^{-1} H_i Y = Y D^{-1} H'_i, \forall 1 \leq i \leq m\} \subseteq \mathbb{L}^{n \times n}$.

If \mathcal{Y} is reduced to the null matrix **then return** “NOSOLUTION”

Pick at random $Y \in \mathcal{Y}$.

Compute $Z = D^{-1} Y^T D Y$ and $J = T^{-1} Z T \in \mathbb{M}^{n \times n}$, the Jordan normal form of Z together with T .

Compute G^{-1} the inverse of a square root of J .

Return $Y T G^{-1}$ and T .

Theorem 15. *Algorithm 1 is correct with probability at least $1 - n/|\mathbb{L}|$ and runs in polynomial time.*

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

Remark 16. Let us recall that the conjugacy problem does not depend on the ground field (see Pazzis (2010)), *i.e.* if there exists $Y \in \text{GL}_n(\mathbb{L}')$, such that $H_i Y = Y H'_i$, then there exists $Y' \in \text{GL}_n(\mathbb{L})$ such that $H_i Y' = Y' H'_i$. This allows us to extend \mathbb{L} 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 $1 - n/|\mathbb{L}'|$ for any extension $\mathbb{L}' \supseteq \mathbb{L}$. The probability can be then made overwhelming by considering extension of degree $O(n)$. In this case, the algorithm returns the description of a solution on $\mathbb{L}'(\sqrt{\zeta_1}, \dots, \sqrt{\zeta_r})$. 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 \mathcal{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 4 summarizes this remark together with Theorem 15.

3.4. Application to Multivariate Cryptography

We focus here our attention to instances of quadratic-IP1S used in cryptographic protocols, *e.g.* Patarin (1996a,b); Tang and Xu (2012, 2014); Yang et al. (2011). In this context, we need to generate an instance with a pre-assigned solution. The idea is simply to pick randomly quadratic forms $\mathbf{f} = (f_1, \dots, f_m)$ and an invertible matrix A to create \mathbf{g} . As a consequence, with overwhelming probability, one of the H_i is invertible, and by extension so is the corresponding H'_i . In fact, with overwhelming probability, all the H_i 's and H'_i 's are invertible. Therefore, one need not amplify \mathbb{K} into \mathbb{L} . However, one must return A defined over \mathbb{K} and not \mathbb{L}' .

The polynomial system $A^T H_i A = H'_i, \forall i, 1 \leq i \leq m$ has n^2 unknowns, the coefficients of A , and $\frac{1}{2}mn(n+1)$ equations. Therefore, it seems to be massively overdetermined as long as $m \geq 3$. We can prove that to compute the set of solutions, it is enough to solve the problem only for $\{H_1, H_2, H_3\}$ and $\{H'_1, H'_2, H'_3\}$ and then to substitute the answer in the $m - 3$ other matrix equations to verify the solution. This is a consequence of Schur's Lemma, which is recalled below in the matrix case, see (Lang, 2002, Chap. XVII, Proposition 1.1) and (Newman, 1967, Lemma 2).

Lemma 17 (Schur's Lemma). *Let $\mathcal{K} = \{K_1, \dots, K_m\}$ be a set of invertible matrices of size $n \times n$ over \mathbb{K} . If the only vector subspaces of \mathbb{K}^n stable by \mathcal{K} are $\{0\}$ and \mathbb{K}^n , then the matrices commuting with \mathcal{K} form a 1-dimensional matrix space.*

If \mathcal{K} stabilizes a nontrivial vector subspace W of \mathbb{K}^n , then \mathcal{K} is similar to a set of upper block triangular matrices whose first blocks have size $d = \dim W$. In our situation, we would have to take $\mathcal{K} = \{K_1 = \text{Id}, K_2 = H_1^{-1}H_2, \dots, K_m = H_1^{-1}H_m\}$. In the cryptographic application, \mathbb{K} is a finite field, namely $\mathbb{K} = \mathbb{F}_q$. The following lemma gives the probability of \mathcal{K} stabilizing of nontrivial vector subspace of \mathbb{F}_q^n .

Lemma 18. *For $m \geq 3$, the probability of $\mathcal{K} = \{K_1 = \text{Id}, \dots, K_m\}$ stabilizing a vector subspace of dimension $d, 1 \leq d \leq n - 1$ in \mathbb{F}_q^n is upper bounded by $\frac{n-1}{q^{(n-1)(m-2)}} \in o(1)$.*

Proof. Let W be a nontrivial vector subspace stabilized by \mathcal{K} . For each K_i , W is the sum of vector subspaces of generalized eigenspaces. In particular, we can assume that $W = \text{Span}(e_1, \dots, e_d)$ where $\mathcal{B} = (e_1, \dots, e_n)$ is a generalized eigenbasis of K_2 , as $K_1 = \text{Id}$. The probability \mathbb{P} of having K_3, \dots, K_m as upper triangular matrices with a block of size d written in basis \mathcal{B} is upper bounded by the probability of picking at random $m - 2$ matrices with a $d \times (n - d)$ -block of zeros. Over \mathbb{F}_q , this probability is at most $q^{-d(n-d)(m-2)}$. Taking the sum over d from 1 to $n - 1$ is once again an upper bound of our probability \mathbb{P} . All in all, we have

$$\mathbb{P} \leq \sum_{d=1}^{n-1} q^{-d(n-d)(m-2)} \leq \sum_{d=1}^{n-1} q^{-(n-1)(m-2)} \leq \frac{n-1}{q^{(n-1)(m-2)}},$$

using the fact that $d(n - d) \geq 1 \cdot (n - 1)$ for $1 \leq d \leq n - 1$. □

On the other hand, it is clear that if A is a solution of $A^T H_i A = H'_i$ for all i , then so is $-A$, allowing us to have at least one degree of freedom. In fact, for generic instances, there is exactly one degree of freedom as long as $m \geq 3$ by Lemmas 17 and 18. That is, by generic, we mean that there is no nontrivial vector subspace W of \mathbb{K}^n such that $H_i(W) = H_1(W)$ for all $i, 2 \leq i \leq m$. For $m = 2$ and generic instances, there are n degrees of freedom, see Bouillaguet et al. (2011b). As a consequence, space \mathcal{Y} of matrices Y verifying $H_1^{-1}H_i Y = Y H_1'^{-1}H'_i, \forall i, 2 \leq i \leq m$ has exactly dimension 1 in cryptography applications with $m \geq 3$. Moreover, A is one of the two matrices lying in \mathcal{Y} verifying $A^T H_1 A = H'_1$. To determine A , it is enough to solve one equation $(A^T H_1 A)_{i,j} = (H'_1)_{i,j}$.

Algorithm 2. Simplified Algorithm for Cryptographic Applications.

Input Two sets of invertible symmetric matrices $\mathcal{H} = \{H_1, \dots, H_m\} \subseteq \mathbb{K}^{n \times n}$ and $\mathcal{H}' = \{H'_1, \dots, H'_m\} \subseteq \mathbb{K}^{n \times n}$.

Output A matrix $A \in \text{GL}_n(\mathbb{K})$ such that $H'_i = A^T H_i A$ for all $1 \leq i \leq m$ or “NOSOLUTION”.

Compute the vector subspace $\mathcal{Y} = \{Y \mid H_1^{-1} H_i Y = Y H_1'^{-1} H_i', \forall 2 \leq i \leq 3\} \subseteq \mathbb{K}^{n \times n}$.
If \mathcal{Y} is reduced to a space of singular matrices **then return** “NOSOLUTION”

$\mathcal{Y} = \{\lambda Y_0 \mid \lambda \in \mathbb{K}\}$.

Solve in λ one equation $\lambda^2 (Y_0^T H_1 Y_0)_{i,j} = (H_1')_{i,j}$ for a suitable pair (i, j) .

$A = \lambda Y_0$.

Pick at random $r \in \mathbb{K}^n$.

Verify that $A^T H_i A r = H_i' r$ for all $i, 1 \leq i \leq m$.

Return A

Complexity. Taking the first 3 matrix equations, in the n^2 unknowns, $H_1^{-1} H_i Y = Y H_1'^{-1} H_i'^{-1}$, 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 verify that $A^T H_i A = H_i'$ for all i , with high probability, by picking up at random a vector and check that the products of this vector with both sets of matrices coincides. This can be done in $O(m n^2)$ operations in \mathbb{K} .

In the end, for generic instances over a finite field \mathbb{F}_q with odd q , one can determine if two sets of m quadratic forms are equivalent in $O(n^{2\omega} + m n^2 + (\log q)^3)$ operations in \mathbb{K} .

3.5. 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, \dots, H_m and H_1', \dots, H_m' such that:

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

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

$$\Sigma(H_i') = A^T \Sigma(H_i) A, \quad \forall i, 1 \leq i \leq m, \quad (6)$$

$$\Delta(H_i') = \Delta(A^T H_i A). \quad (7)$$

It suffices for this to expand $\mathbf{f}(A \cdot \mathbf{x})$ and to consider the upper triangular matrices.

Reduction to canonical representations. As in paragraph 2.ii, w.l.o.g. we can assume that x_1, \dots, x_n are the essential variables of both sets \mathbf{f} and \mathbf{g} . 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. Should we embed \mathbb{K} into its extension \mathbb{L} of degree $2 \left\lceil \frac{\log_{|\mathbb{K}|}(n+1)}{2} \right\rceil + 1$, $\lambda_1, \dots, \lambda_m$ 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 ℓ_1, \dots, ℓ_n in \mathbf{x} such that, see (Lidl and Niederreiter, 1997, Theorem 6.30):

Proof. As in the proof of Proposition 10, we can compute in randomized polynomial time $\lambda_1, \dots, \lambda_m \in \mathbb{L}$ such that $\varphi = \sum_{i=1}^m \lambda_i \cdot f_i$ has n essential variables. We define $\gamma = \sum_{i=1}^m \lambda_i \cdot g_i$. Should one reorder the equations, we can assume w.l.o.g. that $\lambda_1 \neq 0$. We have then:

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

Now, applying Gauß's reduction algorithm to φ yields δ . This gives a $P \in \text{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{f}} = (\tilde{\varphi} = \delta, \tilde{f}_2, \dots, \tilde{f}_m) = (\varphi(P\mathbf{x}), f_2(P\mathbf{x}), \dots, f_m(P\mathbf{x}))$. Clearly, $\mathbf{f} \sim \tilde{\mathbf{f}}$, hence, $\mathbf{f} \sim \mathbf{g}$. After that, we can apply once again Gauß's reduction algorithm to γ . If the reduced polynomial is different from δ , then $\mathbf{f} \not\sim \mathbf{g}$ and we return "NOSOLUTION". Otherwise, the reduction is given by a matrix $Q \in \text{GL}_n(\mathbb{L})$ such that $\tilde{\mathbf{g}} = (\tilde{\gamma} = \delta, \tilde{g}_2, \dots, \tilde{g}_m) = (\gamma(Q\mathbf{x}), g_2(Q\mathbf{x}), \dots, g_m(Q\mathbf{x}))$ and $\mathbf{g} \sim \tilde{\mathbf{g}}$. Thus, $\mathbf{f} \sim \tilde{\mathbf{g}}$ if, and only if, $\mathbf{f} \sim \mathbf{g}$.

Finally, equations (6) $A^T \Sigma(H_i) A = \Sigma(H'_i)$ for all i , $1 \leq i \leq m$ of can be rewritten as $A^T \Sigma(D) A = \Sigma(D)$ 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) $\Delta(A^T H_i A) = \Delta(H'_i)$ for all i , $1 \leq i \leq m$ remain. \square

Remark 20. As for Remark 11, \mathbb{L} of odd degree over \mathbb{K} ensures that if $d \in \mathbb{K}$, $\text{Tr}_{\mathbb{L}}(d) = \text{Tr}_{\mathbb{K}}(d)$. While if \mathbb{L} has even degree over \mathbb{K} , for all $d \in \mathbb{K}$, $\text{Tr}_{\mathbb{L}}(d) = 0$ and this could change the canonical form of \tilde{f}_1 or \tilde{g}_1 .

As a consequence, one first solves the linear system given by, for all i , $A \Sigma(D)^{-1} \Sigma(H'_i) = \Sigma(D)^{-1} \Sigma(H_i) A$. Then, the remaining part $A^T \Sigma(D) A = \Sigma(D)$ and $\Delta(A^T H_i A) = \Delta(H'_i)$ can be done in the same way as in Section 3.2, except for the computation of the square root of the Jordan normal form $J = T^{-1} Z T$ where $Z = \Sigma(D)^{-1} Y^T \Sigma(D) Y$. To the contrary of other characteristics, even if Z is invertible, it might not have any square roots. Even worse, should Z have a square root W , W would not need to be a polynomial in Z , unless Z is diagonalizable. We give a proof of these results in Section 6.2. As a consequence, from an algorithmic point a view, one may have to test multiple solutions Y of the conjugacy problem before finding one which would yield a $\Sigma(D)$ -orthogonal solution.

Finally, one must verify that the computed solution is indeed solution of $\Delta(A^T H_i A) = \Delta(H'_i)$ for all i , $1 \leq i \leq m$.

3.6. Benchmarks

We present in this section some timings of our algorithms over instances of IP1S. We created instances $\mathcal{H} = \{H_1, \dots, H_m\}$ and $\mathcal{H}' = \{H'_1, \dots, H'_m\}$ 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

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.

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 strategy proposed in Section 3.4. 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, with good probability, $i = 2, 3$ gives enough equations to retrieve A up to one free parameter if \mathcal{H} and \mathcal{H}' are indeed equivalent. If there are not, this linear system will return the zero matrix only.

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

By Section 3.4, the complexity of our approach is at most in $O(n^{2\omega})$ operations in \mathbb{F}_p . Recall that, in practice, the best matrix multiplication algorithm is due to Strassen (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 3.5. 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 \mathbb{F}_2 or not equivalent at all over $\overline{\mathbb{F}}_2$, 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 \mathcal{H} and \mathcal{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 $A^T \Sigma(H_1) A = \Sigma(H_1')$ and $\Delta(A^T H_i A) = \Delta(H_i')$, for all i , $1 \leq i \leq m$ (see Proposition 19). We used both MAGMA and the C library M4RI, Albrecht and Bard (2012), and compare their timings.

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.

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.

4. Counting the Solutions: #IP1S

In this part, we present a method for counting the number of solutions to quadratic-IP1S. According to Proposition 6, 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 quadratic homogeneous instances $(\mathbf{f}, \mathbf{g}) \in \mathbb{K}[\mathbf{x}]^m \times \mathbb{K}[\mathbf{x}]^m$ whose number of essential variables is n . If this number is $s < n$, then one can expand the solution matrix with any matrices in $\mathbb{K}^{(n-s) \times s}$ and in $\text{GL}_{n-s}(\mathbb{K})$ (see the proof of Proposition 8). Let $\mathcal{H} = \{H_1 = D, \dots, H_m\}$ and $\mathcal{H}' = \{H'_1 = D, \dots, H'_m\}$ 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 verifying:

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

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

Let α be an algebraic element of degree m over \mathbb{K} and let $\mathbb{K}' = \mathbb{K}(\alpha)$. We consider the matrix $H = D^{-1}(H_1 + \dots + \alpha^{m-1}H_m) \in \mathbb{K}^{m \times n}$. 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 $\mathcal{C}(H)$ of H intersected with $\text{GL}_n(\mathbb{K})$. To ease the analysis, we consider the subspace $\mathcal{V} = \mathcal{C}(H) \cap \mathbb{K}^{n \times n}$ of matrices in $\mathbb{K}^{n \times n}$ commuting with H . This provides an upper bound on the number of solutions. The dimension of \mathcal{V} as a \mathbb{K} -vector space is upper bounded by the dimension of $\mathcal{C}(H)$ as a \mathbb{K}' -vector space. Indeed, $\mathcal{V} \otimes \mathbb{K}' \subseteq \mathcal{C}(H)$, hence $\dim_{\mathbb{K}} \mathcal{V} = \dim_{\mathbb{K}'}(\mathcal{V} \otimes \mathbb{K}') \leq \dim_{\mathbb{K}'} \mathcal{C}(H)$. Since we only want the cardinal 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} . This means that J is a block diagonal matrix wherein each block is a Jordan matrix $J_{\eta,s}$ (*i.e.* an upper triangular matrix of size $s \times s$ whose diagonal elements are η and the elements just above the diagonal are 1).

We recall that a matrix X which commutes with a Jordan matrix J of size s is an upper

triangular Toeplitz matrix of size $s \times s$. Indeed, $XJ - JX$ 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 is used in the following theorem to compute the centralizer of a Jordan normal form.

Theorem 21. *Let J be a Jordan normal form. For $1 \leq i \leq r$, let us denote J_i the i th block of J and let us assume it is associated with eigenvalue η_i and it is of size s_i . Let $X = (X_{i,j})_{1 \leq i,j \leq r}$ be a block-matrix, with $X_{i,j} \in \mathbb{L}(\eta_1, \dots, \eta_r)^{s_i \times s_j}$, that commutes with J . If $\eta_i = \eta_j$, then $X_{i,j}$ is an upper triangular Toeplitz matrix whose nonnecessary zero coefficients are the one on the last $\min(s_i, s_j)$ diagonals starting from to the top-right corner. 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 $J_1 = J_{\eta_1, s_1}$ and $X_{2,2}$ with $J_2 = J_{\eta_2, s_2}$. Thus they are upper triangular Toeplitz matrices.

From $X_{2,1}J_2 - J_1X_{2,1}$, one deduces that $(\eta_1 - \eta_2)x_{s_1+s_2,1} = 0$, hence either $\eta_1 = \eta_2$ or $x_{s_1+s_2,1} = 0$. If $\eta_1 \neq \eta_2$, then step by step, one has $X_{1,2} = 0$. Assuming $\eta_1 = \eta_2$, 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 last $\min(s_1, s_2)$ diagonals starting from the top-right corner. The same argument applies to $X_{2,1}$.

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

Since the centralizer of a matrix is a vector subspace, this characterization of the centralizer allows us to determine an upper bound for its dimension.

Proposition 22 (Proposition 5). *Let $H \in \mathbb{K}^{m \times n}$ be a matrix and let J be its normal Jordan form. Assuming the blocks of J are $J_{\eta_1, s_1, 1}, \dots, J_{\eta_1, s_1, d_1}, \dots, J_{\eta_r, s_r, 1}, \dots, J_{\eta_r, s_r, d_r}$, then the centralizer of H is a \mathbb{K}' -vector subspace of $\mathbb{K}^{m \times n}$ 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 \mathcal{W} , contains $\mathcal{C}(H) \otimes \mathbb{L}$. Hence, $\dim_{\mathbb{K}'} \mathcal{C}(H) = \dim_{\mathbb{L}}(\mathcal{C}(H) \otimes \mathbb{L}) \leq \dim_{\mathbb{L}} \mathcal{W}$. Now, let $X = (X_{i,j})_{1 \leq i,j \leq d_1 + \dots + d_r} \in \mathcal{V}$ and let us assume that for all i , the sequences $(s_{i,1}, \dots, s_{i,d_i})$ are increasing. From Theorem 21, 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. \square

As a consequence, 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:

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

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: $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{K}[x_1, \dots, x_n]^n$, and $\mathbf{POW}_{n,d} = (x_1^d, \dots, x_n^d) \in \mathbb{K}[x_1, \dots, x_n]^m$ for some $d > 0$.

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

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

In Kayal (2011), Kayal proposed a randomized polynomial-time algorithm for solving the problem below when B is the identity matrix and $m = 1$. 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 23. *Let $\mathbf{g} = (g_1, \dots, g_n)$ be polynomials of degree d over $\mathbb{K}[x_1, \dots, x_n]$. We assume that there exists a polynomial-size arithmetic circuit to evaluate the determinant of the Jacobian matrix of \mathbf{g} of size L . If the cardinal 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. 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 $g_i(a_1, \dots, a_{j-1}, a_j + x_j, a_{j+1}, \dots, a_n)$ 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. The determinant of the matrix can be recovered by linear algebra in $O(n^\omega)$ operations. Using this polynomial-size arithmetic circuit of size L to evaluate the determinant of the Jacobian matrix, one can use Kaltofen’s algorithm to factor it in polynomial time. \square

As in Kayal (2011); 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:

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

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

Lemma 24. *Let $(\mathbf{f} = (f_1, \dots, f_m), \mathbf{g} = (g_1, \dots, g_m)) \in \mathbb{K}[x_1, \dots, x_n]^m \times \mathbb{K}[x_1, \dots, x_n]^m$. If there exists $(A, B) \in \text{GL}_n(\mathbb{K}) \times \text{GL}_m(\mathbb{K})$ such that $\mathbf{g} = B \cdot \mathbf{f}(A \cdot \mathbf{x})$, then:*

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

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

As long as $\text{char } \mathbb{K}$ 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 $(\mathbf{J}_{\mathbf{f}}(\mathbf{x}))_{i,i} = d \cdot x_i^{d-1}$, $\forall 1 \leq i \leq n$. Thus:

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

This gives:

Lemma 25. *Let $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{K}[x_1, \dots, x_n]^n$. Let $d > 0$ be an integer, and define $\mathbf{POW}_{n,d} = (x_1^d, \dots, x_n^d) \in \mathbb{K}[x_1, \dots, x_n]^m$. If there exists $(A, B) \in \text{GL}_n(\mathbb{K}) \times \text{GL}_n(\mathbb{K})$ such that $\mathbf{g}(\mathbf{x}) = B \cdot \mathbf{POW}_{n,d}(A \cdot \mathbf{x})$, then:*

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

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

Proof. According to Lemma 24, $\det(\mathbf{J}_{\mathbf{g}}(\mathbf{x})) = \det(A) \cdot \det(B) \cdot d^n \cdot \prod_{i=1}^n \ell_i(\mathbf{x})^{d-1}$. \square

From Lemmas 25, we can derive a randomized polynomial-time algorithm for solving IP on the instance $(\mathbf{f} = \mathbf{POW}_{n,d}, \mathbf{g}) \in \mathbb{K}[x_1, \dots, x_n]^n \times \mathbb{K}[x_1, \dots, x_n]^n$ in characteristic 0. It suffices to use Kaltofen (1989)'s algorithm for factoring $\det(\mathbf{J}_{\mathbf{g}}(\mathbf{x}))$ 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 1 whenever $\text{char } \mathbb{K} \nmid d$.

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

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

with $A^{(p^r)} = \left(a_{i,j}^{p^r} \right)_{\substack{1 \leq j \leq n \\ 1 \leq i \leq n}}$ and $\mathbf{x}^{p^r} = \left(x_1^{p^r}, \dots, x_n^{p^r} \right)$. 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 check 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, instead of considering classical derivatives, which can vanish if $e \geq p$, we consider Hasse-Schmidt derivatives of $\tilde{\mathbf{f}} = \mathbf{POW}_{n,e}$ and $\tilde{\mathbf{g}}$, introduced in Schmidt and Hasse (1936). We define $\frac{\mathcal{D}\tilde{g}_i}{\mathcal{D}x_j}$ as the coefficient in \mathbf{x} of monomial y_j when expanding $\tilde{g}_i(\mathbf{x} + \mathbf{y})$. These derivatives suit better for finite fields and can also be computed in polynomial time using the same argument as in the proof of Theorem 23. The matrix we consider is then

$$J'_{\tilde{\mathbf{g}}}(\mathbf{x}) = \left\{ \mathcal{D}_j \tilde{g}_i = \frac{\mathcal{D}g_i}{\mathcal{D}x_j} \right\}_{\substack{1 \leq j \leq n \\ 1 \leq i \leq m}}.$$

Matrix $J'_{\tilde{\mathbf{f}}}(\mathbf{x})$ is invertible and diagonal with diagonal elements $J'_{\tilde{\mathbf{f}}}(\mathbf{x})_{i,i} = ex_i^{e-1}$. As the chain rule is still valid, a quick computation shows that

$$J'_{\tilde{\mathbf{g}}}(\mathbf{x})_{i,j} = e \sum_{k=1}^m b_{i,k} \left(\sum_{\ell=1}^n a_{k,\ell}^{p^r} x_\ell \right)^{e-1} a_{k,j}^{p^r}.$$

That is, $J'_{\tilde{\mathbf{g}}}(\mathbf{x}) = B \cdot J'_{\tilde{\mathbf{f}}}(A^{(p^r)} \cdot \mathbf{x}) \cdot A^{(p^r)}$. Hence, $\det J'_{\tilde{\mathbf{g}}}(\mathbf{x}) = \det B \cdot \det J'_{\tilde{\mathbf{f}}}(A^{(p^r)} \cdot \mathbf{x}) \cdot \det A^{(p^r)} = (\det A)^{p^r} \cdot \det B \cdot e^n \cdot \prod_{i=1}^n \tilde{\ell}_i(\mathbf{x})^{e-1}$, where the $\tilde{\ell}_i$'s are linear forms whose coefficients are the i th 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 straight-line program for evaluating the polynomial we want to factor has size L , then the cardinal 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 cardinal 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 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 1 for any characteristic, as in the introduction.

Theorem 1. *Let $\mathbf{g} = (g_1, \dots, g_n) \in \mathbb{K}[x_1, \dots, x_n]^n$, and $\mathbf{f} = \mathbf{POW}_{n,d} = (x_1^d, \dots, x_n^d) \in \mathbb{K}[x_1, \dots, x_n]^m$ for some $d > 0$. Assuming \mathbb{K} is big enough, then there is a randomized polynomial-time algorithm which recovers – if any – $(A, B) \in \text{GL}_n(\mathbb{K}) \times \text{GL}_n(\mathbb{K})$ such that:*

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

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 an invertible 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. Let us 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 26. *Let $Z \in \mathbb{K}^{n \times n}$ be an invertible matrix whose eigenvalues are ζ_1, \dots, ζ_r . There exists W a square root of Z whose eigenvalues $\omega_1, \dots, \omega_r$ verify $\omega_i^2 = \zeta_i$ for all $1 \leq i \leq r$. Furthermore, W is a polynomial in Z with coefficients in $\mathbb{K}(\omega_1, \dots, \omega_r)$.*

Proof. We shall prove this proposition incrementally. First, we shall assume that Z only has simple eigenvalues, then that it is diagonalizable and finally that it is any invertible matrix.

Whenever $Z \in \mathbb{K}^{n \times n}$ only has simple eigenvalues ζ_1, \dots, ζ_n , then it is similar to the diagonal matrix D whose entries are the ζ_i 's. Let C be a diagonal matrix whose coefficients are the ω_i 's. Matrix C is a polynomial $P(D)$ if, and only if, there exists p_0, \dots, p_{n-1} such that

$$\begin{pmatrix} \omega_1 & & 0 \\ & \ddots & \\ 0 & & \omega_n \end{pmatrix} = \begin{pmatrix} p_0 + p_1\zeta_1 + \dots + p_{n-1}\zeta_1^{n-1} & & 0 \\ & \ddots & \\ 0 & & p_0 + p_1\zeta_n + \dots + p_{n-1}\zeta_n^{n-1} \end{pmatrix}.$$

Since ζ_1, \dots, ζ_n are pairwise distinct, then this Vandermonde system is invertible over $\mathbb{K}(\omega_1, \dots, \omega_n)$ and thus has a solution. Let us note that no choice of the determination of the square root is needed.

Whenever Z has eigenvalues with multiplicity but is diagonalizable, then one must be careful with the choices of the square roots. For instance, although $W = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ is a square root of $Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, it is not a polynomial in Z over \mathbb{K} . One must choose by advance a determination of the square roots and must stick to it for each eigenvalue. This means, that if $\zeta_i = \zeta_j$, then $\omega_i = \omega_j$. In this case, considering only the eigenvalues that are different, the computation of W comes down naturally to the simple eigenvalues case.

Let us assume that Z is similar to a single Jordan block J for eigenvalue $\zeta \neq 0$ and let ω be a square root of ζ . Noting that $J - \zeta\text{Id}$ is a upper triangular nilpotent matrix and

that $\binom{1/2}{k}$ is well-defined in \mathbb{K} for $\text{char } \mathbb{K} \neq 2$ and $k \geq 0$, then a square root G of J can be computed as follows:

$$G = \sqrt{J} = \omega \sqrt{\text{Id} + \frac{1}{\omega^2} (J - \zeta \text{Id})} = \omega \left(\text{Id} + \frac{1}{2} \frac{1}{\omega^2} (J - \zeta \text{Id}) - \frac{3}{8} \frac{1}{\omega^4} (J - \zeta \text{Id})^2 + \dots \right),$$

where the Taylor expansion is in fact a polynomial in J since $J - \zeta \text{Id}$ is nilpotent. As such, G is the upper triangular matrix such that for all $1 \leq i \leq n, i \leq j \leq n$, the element at row i and column j is $\binom{1/2}{j-i} \omega^{1-2(j-i)}$. Let us note that if J is a Jordan block of size at least 2 for eigenvalue $\zeta = 0$, then J has no square roots.

Let Z be any invertible matrix whose Jordan normal form J is made of blocks J_1, \dots, J_r for eigenvalues ζ_1, \dots, ζ_r . Let G be a block diagonal matrix with blocks G_1, \dots, G_r being square roots of J_1, \dots, J_r with eigenvalues $\omega_1, \dots, \omega_r$ such that $\zeta_i = \zeta_j$ implies $\omega_i = \omega_j$. Obviously G is a square root of J and it remains to prove that G is a polynomial $P(J)$. Assuming J_1 has size $d_1 + 1, \dots, J_r$ has size $d_r + 1$, then finding P comes down to interpolate P knowing that $P(\zeta_1) = \omega_1, \dots, P^{(d_1)}(\zeta_1) = \binom{1/2}{d_1} \omega^{1-2d_1}, \dots, P(\zeta_r) = \omega_r, \dots, P^{(d_r)}(\zeta_r) = \binom{1/2}{d_r} \omega^{1-2d_r}$ and such a P can always be found. For instance, if ζ_1 is an eigenvalue with *algebraic* multiplicity 2 but *geometric* multiplicity 1 and if ζ_2 has multiplicity 1, then one will have to interpolate a polynomial $P(z) = p_0 + p_1 z + p_2 z^2$ such that

$$\begin{aligned} \begin{pmatrix} \omega_1 & \frac{1}{2\omega_1} & 0 \\ 0 & \omega_1 & 0 \\ 0 & 0 & \omega_2 \end{pmatrix} &= \begin{pmatrix} p_0 + p_1 \zeta_1 + p_2 \zeta_1^2 & p_1 + 2p_2 \zeta_1 & 0 \\ 0 & p_0 + p_1 \zeta_1 + p_2 \zeta_1^2 & 0 \\ 0 & 0 & p_0 + p_1 \zeta_2 + p_2 \zeta_2^2 \end{pmatrix} \\ &= \begin{pmatrix} P(\zeta_1) & P'(\zeta_1) & 0 \\ 0 & P(\zeta_1) & 0 \\ 0 & 0 & P(\zeta_2) \end{pmatrix}, \end{aligned}$$

and this system has a unique solution since $\zeta_1 \neq \zeta_2$. Note that if $\omega_1 = \omega_2$, then there is also a unique P of degree 1 verifying the equation above.

Finally, once one has P such that $P(J) = G$, then it is clear that $P(Z) = W$ with $W^2 = Z$. \square

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 $\text{char } \mathbb{K} = 2$. Unfortunately, unlike other characteristics, an invertible matrix has not necessarily a square root over \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 27. *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 ζ_1, \dots, ζ_r and blocks of sizes 1 with eigenvalues v_1, \dots, v_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 $v_j = \zeta_i$.

Before, proving this result, we give some example of matrices with or without square root. Following matrices J and J' have square roots K and K' , while J'' and J''' do not have any:

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

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 $1/y$.

Proof. Let J be a Jordan block of size d associated to eigenvalue ζ . Then $J^2 - \zeta^2 \text{Id} = \begin{pmatrix} 0 & \text{Id}_{d-2} \\ & 0 \end{pmatrix}$ 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 $(J - \zeta \text{Id})^d = 0$ and $(J - \zeta \text{Id})^e \neq 0$ for all $e < d$, then $(J^2 - \zeta^2 \text{Id})^{\lceil d/2 \rceil} = 0$ and $(J^2 - \zeta^2 \text{Id})^e \neq 0$ 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, let us prove that if $W^2 = Z$ and Z is not diagonalizable, then W is not a polynomial in Z . Let J be the Jordan normal form of Z with blocks J_1, \dots, J_r . For any polynomial P , $P(J)$ is also block diagonal with blocks $P(J_1), \dots, P(J_r)$. 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. \square

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 $\text{char } \mathbb{K} > 2$. 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 $D = T^{-1}ZT$ has an easily computable square root. This matrix D 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 \mathbb{K} , then D is block diagonal, otherwise it is a block upper triangular matrix. Its diagonal blocks are companion matrices $\mathcal{C}(P_1), \dots, \mathcal{C}(P_r)$ of irreducible factors P_1, \dots, 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 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 C of D , one can retrieve W and M in $O(n^\omega)$ operations in the field of coefficients of C . Furthermore, computing a square root of D 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, C is a polynomial in D . In the following, we only show how to determine the square root of a companion matrix $\mathcal{C}(P)$, for an irreducible P .

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

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

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 $\mathcal{C}(Q)^2$ is

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

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

As P is irreducible over $\mathbb{K} = \mathbb{F}_q$, up to reindexing the roots of P , the conjugates $\alpha_1, \dots, \alpha_{d-1}$ of α_0 are just its iterated q th 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 q th powers. In that case, the previous equations can be rewritten

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

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} :

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

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 $y^{q^i} = yy^{q^i-1} = yx^{\frac{q^i-1}{2}}$, we can compute the following values in $O(d \log q)$ field operations in \mathbb{L} :

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

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 $O(dM(d) \log q)$ operations in $\mathbb{K} = \mathbb{F}_q$. As a consequence, the computation of W can be done in no more than $O(n^\omega + nM(n) \log q)$ 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.

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: Khanna (2013), pp. 1337–1355.
- Bini, D., Pan, V. Y., 1994. Polynomial and matrix computations. Vol. 1. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, fundamental algorithms.

- 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).
URL <http://dx.doi.org/10.1006/jsco.1996.0125>
- Bouillaguet, C., Faugère, J.-C., Fouque, P.-A., Perret, L., 2011a. 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., Macario-Rat, G., 2011b. Practical Key-Recovery for All Possible Parameters of SFLASH. In: Lee, D. H., Wang, X. (Eds.), *ASIACRYPT*. Vol. 7073 of *Lecture Notes in Computer Science*. Springer, pp. 667–685.
- 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.
- 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: *Khanna (2013)*, pp. 1074–1087.
- 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.

- 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.
- Gathen, J. v. z., Gerhard, J., 1999. *Modern computer algebra*. Cambridge University Press, New York.
- 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.
- Khanna, S. (Ed.), 2013. *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*. SIAM.

- 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., 1996a. 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., 1996b. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: EUROCRYPT. pp. 33–48, extended version available on <http://www.minrank.org/hfe.pdf>.
- 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.
- Pazzis, C. d. S., 2010. Invariance of simultaneous similarity and equivalence of matrices under extension of the ground field. *Linear Algebra and its Applications* 433 (3), 618 – 624.

- 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., June 2006. Morphisms of rings and applications to complexity. Ph.D. thesis, INDIAN INSTITUTE OF TECHNOLOGY KANPUR.
- Schmidt, F., Hasse, H., 1936. Noch eine Begründung der Theorie der höheren Differentialquotienten in einem algebraischen Funktionenkörper einer Unbestimmten. (Nach einer brieflichen Mitteilung von F.K. Schmidt in Jena). *J. Reine Angew. Math* 177, 215–237.
- 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.
- 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 (0), 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 Selected Papers.
- Valiant, L. G., 1979. The complexity of computing the permanent. *Theor. Comput. Sci.* 8, 189–201.
- 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.