# On solving systems of diagonal polynomial equations over finite fields

Gábor Ivanyos * Miklos Santha †

December 11, 2015

**Abstract**

We present a randomized algorithm to solve a system of diagonal polynomial equations over finite fields when the number of variables is greater than some fixed polynomial of the number of equations whose degree depends only on the degree of the polynomial equations. Our algorithm works in time polynomial in the number of equations and the logarithm of the size of the field, whenever the degree of the polynomial equations is constant. As a consequence we design polynomial time quantum algorithms for two algebraic hidden structure problems: for the hidden subgroup problem in certain semidirect product $p$-groups of constant nilpotency class, and for the multi-dimensional univariate hidden polynomial graph problem when the degree of the polynomials is constant.

**Keywords:** algorithm, polynomial equations, finite fields, Chevalley–Warning theorem, quantum computing

## 1 Introduction

Finding small solutions in some well defined sense for a system of integer linear equations is an important, well studied, and computationally hard problem. *Subset Sum*, which asks the solvability of a single equation in the binary domain is one of Karp's original 21 NP-complete problems [16].

The guarantees of many lattice based cryptographic system come from the average case hardness of *Short Integer Solution*, dating back to Ajtai's breakthrough work [1], where we try to find short nonzero vectors in a random integer lattice. Indeed, this problem has a remarkable worst case versus average case hardness property: solving it on the average is at least as hard as solving various lattice problems in the worst case, such as the decision version of the shortest vector problem, and finding short linearly independent vectors.

Turning back to binary solutions, deciding, if there exists a nonzero solution of the system of linear equations

$$\begin{array}{rcl} a_{11}x_1 + \ldots + a_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n & = & 0 \end{array} \tag{1}$$

*Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary (Gabor.Ivanyos@sztaki.mta.hu).

†CNRS, LIAFA, Université Paris Diderot 75205 Paris, France; and Centre for Quantum Technologies, National University of Singapore, Singapore 117543 (miklos.santha@gmail.com).

in the finite field $\mathbb{F}_p$, for some prime number $p$ is easy when $p = 2$. However, by modifying the standard reduction of *Satisfiability* to *Subset Sum* [24] it can be shown that it is an NP-hard problem for $p \geq 3$.

The system (1) is equivalent to the system of equations

$$\begin{array}{rcl} a_{11}x_1^{p-1} + \ldots + a_{1n}x_n^{p-1} & = & 0 \\ \vdots & \vdots & \vdots \\ a_{m1}x_1^{p-1} + \ldots + a_{mn}x_n^{p-1} & = & 0 \end{array} \tag{2}$$

where we look for a nonzero solution in the whole $\mathbb{F}_p^n$.

In this paper we will consider finding a nonzero solution for a system of diagonal polynomial equations similar to (2), but where more generally, the variables are raised to some power $2 \leq d$. We state formally this problem.

**Definition 1** The *System of Diagonal Equation* problem SDE is parametrized by a finite field $\mathbb{F}$ and three positive integers $n, m$ and $d$.

SDE($\mathbb{F}, n, m, d$)

*Input:* A system of polynomial equations over $\mathbb{F}$:

$$\begin{array}{rcl} a_{11}x_1^{d} + \ldots + a_{1n}x_n^{d} & = & 0 \\ \vdots & \vdots & \vdots \\ a_{m1}x_1^{d} + \ldots + a_{mn}x_n^{d} & = & 0 \end{array} \tag{3}$$

*Output:* A nonzero solution $(x_1, \ldots, x_n) \neq 0^n$.

For $j = 1, \ldots, n$, let us denote by $v_j$ the vector $(a_{1j}, \ldots, a_{mj}) \in \mathbb{F}^m$. Then the system of equations (3) is the same as

$$\sum_{j=1}^{n} x_j^d v_j = 0. \tag{4}$$

That is, solving SDE($\mathbb{F}, n, m, d$) is equivalent to the task of representing the zero vector as a nontrivial linear combinations of a subset of $\{v_1, \ldots, v_n\}$ with $d$th power coefficients. We present our algorithm actually as solving this vector problem. The special case $d = |\mathbb{F}| - 1$ is the vector zero sum problem where the goal is to find a non-empty subset of the given vectors with zero sum.

Under which conditions can we be sure that for system (3) there exists a nonzero solution? The elegant result of Chevalley [3] states that a system of homogeneous polynomial equations has a nonzero solution if the number of variables is greater than the sum of the degrees of the polynomials. In our case this means that when $n > dm$, the existence of a nonzero solution is assured. In addition, Warning has proven [26] that under similar condition the number of solutions is in fact a multiple of the characteristic of $\mathbb{F}$.

In general where little is known about the complexity of finding a nonzero solution for systems which satisfy the Chevalley condition. When $|\mathbb{F}| = 2$, Papadimitriou has shown [20] that this problem is in the complexity class Polynomial Parity Argument (PPA), the class of NP search problems where the existence of the solution is guaranteed by the fact that in every finite graph the number of vertices with odd degree is even. This implies that it can not be NP-hard unless NP = co-NP. Nonetheless finding efficiently a nonzero solution in general seems to be a very hard task.

Let us come back to our special system of equations (3). In the case $m = 1$, a nonzero solution can be found in polynomial time for the single equation which satisfies the Chevalley condition due to the remarkable work of van de Woestijne [25] where he proves the following.

**Fact 2** *In deterministic polynomial time in $d$ and $\log|\mathbb{F}|$ we can find a nontrivial solution for $a_1x_1^d + \ldots + a_{d+1}x_{d+1}^d = 0$.*

In the case of more than one equation we don't know how to find a nonzero solution for equation (3) under just the Chevalley condition. However, if we relax the problem, and take much more variable than required for the existence of a nonzero solution, we are able to give a polynomial time solution. Using van de Woestijne's result for the one dimensional case, a simple recursion on $m$ shows that if $n \geq (d+1)^m$ then $\mathrm{SDE}(\mathbb{F}_p, n, m, d)$ can be solved in deterministic polynomial time in $n$ and $\log p$. The time complexity of this algorithm is therefore polynomial for any fixed $m$. The case when $d$ is fixed and $m$ grows appears to be more difficult. To our knowledge, the only existing result in this direction is the case $d = 2$ for which it was shown in [14] that there exists a randomized algorithm that, when $n = \Omega(m^2)$, solves $\mathrm{SDE}(\mathbb{F}_p, n, m, d)$ in polynomial time in $n$ and $\log p$. In the main result of this paper we generalize this result by showing, for every constant $d$, the existence of a randomized algorithm that, for every $n$ larger than some polynomial function of $m$, solves $\mathrm{SDE}(\mathbb{F}_p, n, m, d)$ in polynomial time in $n$ and $\log p$.

**Theorem 3** *Let $d$ be constant. For $n > d^{d^2 \log d}(m+1)^{d \log d}$, the problem $\mathrm{SDE}(\mathbb{F}_p, n, m, d)$ can be solved by a randomized algorithm in polynomial time in $n$ and $\log p$.*

The large number of variables that makes possible a polynomial time solution unfortunately also makes our algorithm most probably irrelevant for cryptographic applications. Nonetheless, it turns out the the algorithm is widely applicable in quantum computing for solving efficiently various algebraic hidden structure problems. We explain now this connection.

Simply speaking, in a hidden structure problem we have to find some hidden object related to some explicitly given algebraic structure $A$. We have access to an oracle input, which is an unknown member $f$ of a family of black-box functions which map $A$ to some finite set $S$. The task is to identify the hidden object solely from the information one can obtain by querying the oracle $f$. This means that the only useful information we can obtain is the structure of the level sets $f^{-1}(s) = \{a \in A : f(a) = s\}$, $s \in S$, that is, we can only determine whether two elements in $A$ are mapped to the same value or not. In these problems we say that the input $f$ *hides* the hidden structure, the output of the problem. We define now the two problems for which we can apply our algorithm for SDE.

**Definition 4** The *hidden subgroup problem* HSP is parametrized by a finite group $G$ and a family $\mathcal{H}$ of subgroups of $G$.

$\mathrm{HSP}(G, \mathcal{H})$

*Oracle input:* A function $f$ from $G$ to some finite set $S$.

*Promise:* For some $H \in \mathcal{H}$, we have $f(x) = f(y) \iff Hx = Hy$.

*Output:* $H$.

The *hidden polynomial graph problem* HPGP is parametrized by a finite field $\mathbb{F}_p$ and three positive integers $n, m$ and $d$.

$\mathrm{HPGP}(\mathbb{F}_p, n, m, d)$.

*Oracle input:* A function $f$ from $\mathbb{F}_p^n \times \mathbb{F}_p^m$ to a finite set $S$.

*Promise:* For some $Q : \mathbb{F}_p^n \to \mathbb{F}_p^m$, where $Q(x) = (Q_1(x), \ldots, Q_m(x))$, and $Q_i(x)$ is an $n$-variate degree $d$ polynomial over $\mathbb{F}_p$ with zero constant term, we have $f(x, y) = f(x', y') \iff y - Q(x) = y' - Q(x')$.
*Output:* $Q$.

While no classical algorithm can solve the HSP with polynomial query complexity even if the group $G$ is abelian, one of the most powerful results of quantum computing is that it can be solved by a polynomial time quantum algorithm for any abelian $G$ (see, e.g., [15]). Shor's factorization and discrete logarithm finding algorithms [23], and Kitaev's algorithm [17] for the abelian stabilizer problem are all special cases of this general solution.

Extending the quantum solution of the abelian HSP to non abelian groups is an active research area since these instances include several algorithmically important problems. For example, efficient solutions for the dihedral and the symmetric group would imply efficient solutions, respectively, for several lattice problems [21] and for graph isomorphism. While the non abelian HSP has been solved efficiently by quantum algorithms in various groups [2, 8, 9, 10, 11, 18, 19], finding a general solutions seems totally elusive.

A different type of extension was proposed by Childs, Schulman and Vazirani [4] who considered the problem where the hidden object is a polynomial. To recover it we have at our disposal an oracle whose level sets coincide with the level sets of the polynomial. Childs et al. [4] showed that the quantum query complexity of this problem is polynomial in the logarithm of the field size when the degree and the number of variables are constant. In [7] the first time efficient quantum algorithm was given for the case of multivariate quadratic polynomials over fields of constant characteristic.

The hidden polynomial graph problem HPGP was defined in [5] by Decker, Draisma and Wocjan. Here the hidden object is again a polynomial, but the oracle is more powerful than in [4] because it can also be queried on the graphs that are defined by the polynomial functions. They obtained a polynomial time quantum algorithm that correctly identifies the hidden polynomial when the degree and the number of variables are considered to be constant. In [7] this result was extended to polynomials of constant degree. The version of the HPGP we define here is more general than the one considered in [5] in the sense that we are dealing not only with a single polynomial but with a vector of several polynomials. The restriction on the constant terms of the polynomials are due to the fact that level sets of two polynomials are the same if they differ only in their constant terms, and therefore the value of the constant term can not be recovered.

It will be convenient for us to consider a slight variant of the hidden polynomial graph problem which we denote by HPGP$'$. The only difference between the two problems is that in the case of HPGP$'$ the input is not given by an oracle function but by the ability to access random *level set states*, which are quantum states of the form

$$\sum_{x \in \mathbb{F}_p^n} |x\rangle |u + Q(x)\rangle,$$

where $u$ is a random element of $\mathbb{F}_p^m$. Given an oracle input $f$ for HPGP, a simple and efficient quantum algorithm can create such a random coset state. Therefore an efficient quantum algorithm for HPGP$'$ immediately provides an efficient quantum algorithm for HPGP.

In [6] it was shown that HPGP$'(\mathbb{F}_p, 1, m, d)$ is solvable in quantum polynomial time when $d$ and $m$ are both constant. Part of the quantum algorithm repeatedly solved instances of SDE$(\mathbb{F}_p, n, m, d)$ under such conditions. We present here a modification of this method which works in polynomial time even if $m$ is not constant.

**Theorem 5** *Let $d$ be constant. If* SDE$(\mathbb{F}_p, n, m, d)$ *is solvable in randomized polynomial time for some $n$, then* HPGP$'(\mathbb{F}_p, 1, m, d)$ *is solvable in quantum polynomial time.*

Using Theorem 3 it is possible to dispense in the result of [6] with the assumption that $m$ is constant.

**Corollary 6** *If $d$ is constant then* $\mathrm{HPGP}'(\mathbb{F}_p, 1, m, d)$ *is solvable in quantum polynomial time.*

Bacon, Childs and van Dam in [2] have considered the HSP in $p$-groups of the form $G = \mathbb{F}_p \ltimes \mathbb{F}_p^m$ when the hidden subgroup belongs to the family $\mathcal{H}$ of subgroups of order $p$ which are not subgroups of the normal subgroup $0 \times \mathbb{F}_p^m$. They have found an efficient quantum algorithm for such groups as long as $m$ is constant. In [7], based on arguments from [2] it was sketched how the $\mathrm{HSP}(\mathbb{F}_p \ltimes \mathbb{F}_p^m, \mathcal{H})$ can be translated into a hidden polynomial graph problem. For the sake of completeness we state here and prove the exact statement about such a reduction.

**Proposition 7** *Let $d$ be the nilpotency class of a group $G$ of the form $\mathbb{F}_p \ltimes \mathbb{F}_p^m$. There is a polynomial time quantum algorithm which reduces* $\mathrm{HSP}(G, \mathcal{H})$ *to* $\mathrm{HPGP}'(\mathbb{F}_p, 1, m, d)$.

Putting together Corollary 6 and Poroposition 7, it is also possible to get rid of the assumption that $m$ is constant in the result of [2].

**Corollary 8** *If the nilpotency class of the group $G$ of the form $\mathbb{F}_p \ltimes \mathbb{F}_p^m$ is constant then* $\mathrm{HSP}(G, \mathcal{H})$ *can be solved in quantum polynomial time.*

The special cases of Theorem 3 for $d = 2, 3$ will be shown in Section 2. The proof of Theorem 3 will be given in Section 3. The proofs of Theorem 5 and Proposition 7 are given in the full and improved version of the paper [13]. We remark that the proof of Theorem 3 extends to arbitrary finite fields (only minor notational changes are needed). Also, the method can be made deterministic using techniques similar to those used by van de Woestijne in [25]. Details of these can also be found in [13].

## 2 Warm-up: the quadratic and cubic cases

### 2.1 The quadratic case

**Proposition 9** *The problem* $\mathrm{SDE}(\mathbb{F}_p, (m+1)^2, m, 2)$ *can be solved in randomized polynomial time.*

**W.** e assume that $p > 2$ and that we have a non-square $\zeta$ in $\mathbb{F}_p$ at hand. Such an element can be efficiently found by a random choice. Assuming GRH, even a deterministic polynomial time method exists for finding a non-square.

Our input is a set $V$ of $(m+1)^2$ vectors in $\mathbb{F}_p^m$, and we want to represent the zero vector as a nontrivial linear combination of some vectors from $V$ where all the coefficients are squares. The construction is based on the following. Pick any $m+1$ vectors $u_1, \ldots, u_{m+1}$ from $\mathbb{F}_p^m$. Since they are linearly dependent, it is easy to represent the zero vector as a proper linear combination $\sum_{i=1}^{m+1} \alpha_i u_i = 0$. Let $J_1 = \{i : \alpha_i^{\frac{p-1}{2}} = 1\}$ and $J_2 = \{i : \alpha_i^{\frac{p-1}{2}} = -1\}$. Using $\zeta$, we can efficiently find in deterministic polynomial time in $\log p$ by the Shanks-Tonelli algorithm [22] field elements $\beta_i$ such that $\alpha_i = \beta_i^2$ for $i \in J_1$ and $\alpha_i = \beta_i^2 \zeta$ for $i \in J_2$. Let $w_1 = \sum_{i \in J_1} \beta_i^2 v_i$ and $w_2 = \sum_{i \in J_2} \beta_i^2 v_i$. Then $w_1 = -\zeta w_2$. Notice that we are done if either of the sets $J_1$ or $J_2$ is empty.

What we have done so far, can be considered as a high-level version of the approach of [14]. The method of [14] then proceeds with recursion to $m-1$. Unfortunately, that approach is appropriate

only in the quadratic case. Here we use a completely different idea which will turn to be extensible to more general degrees.

From the vectors in $V$ we form $m+1$ pairwise disjoint sets of vectors of size $m+1$. By the construction above, we compute $w_1(1)$, $w_2(1), \ldots$, $w_1(m+1)$, $w_2(m+1)$, where

$$w_1(i) = -\zeta w_2(i), \tag{5}$$

for $i = 1, \ldots, m+1$. Moreover, these $2m$ vectors are represented as linear combinations with nonzero square coefficients of $2m$ pairwise disjoint nonempty subsets of the original vectors.

Now $w_1(1), \ldots, w_1(m+1)$ are linearly dependent and again we can find disjoint subsets $J_1$ and $J_2$ and scalars $\gamma_i$ for $i \in J_1 \cup J_2$ such that for $w_{11} = \sum_{i \in J_1} \gamma_i^2 w_1(i)$ and $w_{12} = \sum_{i \in J_2} \gamma_i^2 w_1(i)$ we have $w_{11} = -\zeta w_{12}$. But then for $w_{21} = \sum_{i \in J_2} \gamma_i^2 w_2(i)$ and $w_{22} = \sum_{i \in J_2} \gamma_i^2 w_1(i)$, using equation (5) for all $i$, we similarly have $w_{21} = -\zeta w_{22}$. On the other hand, if we sum up equation (5) for $i \in J_1$, we get $w_{11} = -\zeta w_{21}$. Therefore $w_{11} = \zeta^2 w_{22}$ and $w_{12} = w_{21} = -\zeta w_{22}$.
By Fact 2 we can find field elements $\delta_{11}, \delta_{22}, \delta_{12}$, not all zero, such that
$\zeta^2\delta_{11}^2 - 2\zeta\delta_{12}^2 + \delta_{22}^2 = 0$, and therefore $(\zeta^2\delta_{11}^2 - 2\zeta\delta_{12}^2 + \delta_{22}^2)w_{22} = 0$. But

$$(\zeta^2\delta_{11}^2 - 2\zeta\delta_{12}^2 + \delta_{22}^2)w_{22} = \delta_{11}^2 w_{11} + \delta_{12}^2(w_{12} + w_{21}) + \delta_{22}^2\zeta^2 w_{22}.$$

Then expanding $\delta_{11}^2 w_{11} + \delta_{12}^2(w_{12} + w_{21}) + \delta_{22}^2\zeta^2 w_{22} = 0$ gives a representation of the zero vector as a linear combination with square coefficients (squares of appropriate product of $\beta$s, $\gamma$s and $\delta$s) of a subset of the original vectors. □

## 2.2 The cubic case

**Proposition 10** *Let $n = (9m+1)(3m+1)(m+1)$. Then* SDE($\mathbb{F}_p, n, m, 3$) *can be solved in randomized polynomial time.*

**W.** e assume that $p-1$ is divisible by 3 since otherwise the problem is trivial. By a randomized polynomial time algorithm we can compute two elements $\zeta_2, \zeta_3$ from $\mathbb{F}_p$ such that $\zeta_1 = 1, \zeta_2, \zeta_3$ are a complete set of representatives of the cosets of the subgroup $\{x^3 : x \in \mathbb{F}_p^*\}$ of $\mathbb{F}_p^*$. Let $V$ be our input set of $n$ vectors in $\mathbb{F}_p^m$, now we want to represent the zero vector as a nontrivial linear combination of some vectors from $V$ where all the coefficients are cubes.

As in the quadratic case, for any subset of $m+1$ vectors $u_1, \ldots, u_{m+1}$ from $V$, we can easily find a proper linear combination summing to zero, $\sum_{i=1}^{m+1} \alpha_i u_i = 0$. For $r = 1, 2, 3$, let $J_r$ be the set of indices such that $0 \neq \alpha_i = \beta_i^3 \zeta_r$. We know that at least one of these three sets is non-empty. For each $\alpha_i \neq 0$ we can efficiently identify the coset of $\alpha_i$ and even find $\beta_i$. Let $w_r = \sum_{i \in J_r} \beta_i^3 v_i$. Then $\zeta_1 w_1 + \zeta_2 w_2 + \zeta_3 w_3 = 0$. Without loss of generality we can suppose that $J_1$ is non-empty since if $J_r$ is non-empty for $r \in \{2, 3\}$, we can just multiply $\alpha_i$s simultaneously by $\zeta_1/\zeta_r$.

From any subset of size $(3m+1)(m+1)$ of $V$ we can form $3m+1$ groups of size $m+1$, and within each group we can do the procedure outlined above. This way we obtain, for $k = 1, \ldots, 3m+1$, and $r = 1, 2, 3$, pairwise disjoint subsets $J_r(k)$ of indices and vectors $w_r(k)$ such that

$$\zeta_1 w_1(k) + \zeta_2 w_2(k) + \zeta_3 w_3(k) = 0. \tag{6}$$

For $k = 1, \ldots, 3m+1$, we know that $J_1(k) \neq \emptyset$ and the vectors $w_r(k)$ are combinations of input vectors with indices form $J_r(k)$ having coefficients which are nonzero cubes. Let $W(k) \in F_p^{3m}$ denote the vector obtained by concatenating $w_1(k)$, $w_2(k)$ and $w_3(k)$ (in this order). Then we can

find three pairwise disjoint subsets $M_1, M_2, M_3$ of $\{1, \ldots, 3m+1\}$, and for each $k \in M_s$, a nonzero field element $\gamma_k$ such that

$$\sum_{s=1}^{3} \zeta_s \sum_{k \in M_s} \gamma_k^3 W(k) = 0. \tag{7}$$

We can arrange that $M_2$ is non-empty. For $r, s \in \{1,2,3\}$, set $J_{rs} = \bigcup_{k \in M_s} J_r(k)$ and $w_{rs} = \sum_{k \in M_s} \gamma_k^3 w_r(k)$. Then $w_{rs}$ is a linear combination of input vectors with indices from $J_{rs}$ having coefficients that are nonzero cubes. The equality (7) just states that $\zeta_1 w_{r1} + \zeta_2 w_{r2} + \zeta_3 w_{r3} = 0$, for $r = 1, 2, 3$. Furthermore, summing up the equalities (6) for $k \in M_s$, we get $\zeta_1 w_{1s} + \zeta_2 w_{2s} + \zeta_3 w_{3s} = 0$, for $s = 1, 2, 3$.

Continuing this way, from $(9m+1)(3m+1)(m+1)$ input vectors we can make 27 linear combinations with cubic coefficients $w_{rst}$, for $r, s, t = 1, 2, 3$, having pairwise disjoint supports such that the support of $w_{123}$ is non-empty and they satisfy the 27 equalities $\zeta_1 w_{1st} + \zeta_2 w_{2st} + \zeta_3 w_{3st} = 0$ $(s, t = 1, 2, 3)$; $\zeta_1 w_{r1t} + \zeta_2 w_{r2t} + \zeta_3 w_{r3t} = 0$ $(r, t = 1, 2, 3)$; $\zeta_1 w_{rs1} + \zeta_2 w_{rs2} + \zeta_3 w_{rs3} = 0$ $(r, s = 1, 2, 3)$. From these we use the following 6 equalities: $\zeta_1 w_{123} + \zeta_2 w_{223} + \zeta_3 w_{323} = 0$; $\zeta_1 w_{132} + \zeta_2 w_{232} + \zeta_3 w_{332} = 0$; $\zeta_1 w_{213} + \zeta_2 w_{223} + \zeta_3 w_{233} = 0$; $\zeta_1 w_{312} + \zeta_2 w_{322} + \zeta_3 w_{332} = 0$; $\zeta_1 w_{231} + \zeta_2 w_{232} + \zeta_3 w_{233} = 0$; $\zeta_1 w_{321} + \zeta_2 w_{322} + \zeta_3 w_{323} = 0$. Adding these equalities with appropriate signs so that the terms with coefficients $\zeta_2$ and $\zeta_3$ cancel and dividing by $\zeta_1$, we obtain $w_{123} + w_{231} + w_{312} - w_{132} - w_{213} - w_{321} = 0$. Observing that $-1 = (-1)^3$, this gives a representation of zero as a linear combination of the input vectors with coefficients that are cubes.

□

# 3 The general case

In this section we prove Theorem 3. First we make the simple observation that it is sufficient to solve $\mathrm{SDE}(\mathbb{F}_p, n, m, d)$ in the case when $d$ divides $p-1$. If it is not the case, then let $d' = \gcd(d, p-1)$. Then from a nonzero solution of the system

$$\sum_{j=1}^{n} x_j^{d'} v_j = 0,$$

one can efficiently find a nonzero solution of the original equation. Indeed, the extended Euclidean algorithm efficiently finds a positive integer $t$ such that $td = u(p-1) + d'$ for some integer $u$. Then for any nonzero $x \in \mathbb{F}_p$ we have $(x^t)^d = x^{d'} \mod p$, and therefore $(x_1^t, \ldots, x_n^t)$ is a solution of equation (4). From now on we suppose that $d$ divides $p-1$.

Our algorithm will distinguish two cases, according to the value of $d$. The first case is when $-1$ is not a $d$th power in $\mathbb{F}_p$. Then $d$ is necessarily an even number, and we give a method which reduces to the problem HPGP with polynomials of degree $d/2$. Observe that in that case $-1$ is a $d/2$th power, and the algorithm proceeds with the method of the second case. The second case is when $-1$ is a $d$th power in $\mathbb{F}_p$, then our algorithm directly solves the problem. For both cases we will denote by $C(d, m)$ the number of vectors (variables) used by our algorithm. For $d = 1$, we can take $C(1, m) = m + 1$.

## 3.1 The reduction when $d$ is even

We assume that $p-1$ is divisible by $d$ and that we have a non-square $\zeta$ in $\mathbb{F}_p$ at hand. We also assume that we can efficiently express the zero vector as a nontrivial linear combination with $d$th

power coefficients of any given $t = C(d/2, m)$ vectors $u_1, \ldots, u_t \in \mathbb{F}_p^m$: $\sum_{i=1}^t \alpha_i^d u_i = 0$.

As in the quadratic case, let $J_1 = \{i : \alpha_i^{\frac{p-1}{2}} = 1\}$ and $J_2 = \{i : \alpha_i^{\frac{p-1}{2}} = -1\}$. Using $\zeta$, we can efficiently find $\beta_i$ such that $\alpha_i = \beta_i^2$ for $i \in J_1$ and $\alpha_i = \beta_i^2\zeta$ for $i \in J_2$. Let $w_1 = \sum_{i\in J_1} \beta_i^2 v_i$ and $w_2 = \sum_{i\in J_2} \beta_i^2 v_i$. Then $w_1 = -\zeta^d w_2$. Note that we are done if either of the sets $J_1$ or $J_2$ is empty.

Suppose that we have $C(d/2, m)$ groups, each consisting of $C(d/2, m)$ vectors of length $m$. For each $i$, we can build vectors $w_1(i)$ and $w_2(i)$ in the $i$th group with the properties of $w_1$ and $w_2$ above. Then we can express the zero vector as a linear combination with nonzero $d$th power coefficients from a subset of the vectors $w_1(i)$. Like in the quadratic case, we find four vectors, a scalar multiple of each other, represented as nontrivial linear combinations with $d$th power coefficients of four pairwise disjoint subsets of the original variables.

We can iterate this process. In the $\ell$th iteration we start with $C(d/2, m)$ groups, each consisting of $C(d/2, m)^{\ell-1}$ vectors of length $m$. At the end of the $\ell$th iteration we can find a nonzero vector $w$ and scalars $\lambda_1, \ldots, \lambda_{2^\ell}$ together with representations of the vectors $\lambda_1 w, \ldots, \lambda_{2^\ell} w$ as linear combination with nonzero $d$th power coefficients of $\ell$ pairwise disjoint subsets of the original vectors.

After $\lceil \log_2(d+1) \rceil \le \log d + 1$ iterations, starting from at most $C(d/2, m)^{\log d+1}$ input vectors, we get a vector $w$ and scalars $\lambda_1, \ldots, \lambda_{d+1}$, together with the representations of the vectors $w_1 = \lambda_1 w, \ldots, w_{d+1} = \lambda_{d+1} w$ as above.

By Fact 2 we can find field elements $z_1, \ldots, z_{d+1}$ such that $\sum_{i=1}^{d+1} \lambda_i z_i^d = 0$, which implies that $\sum_{i=1}^{d+1} z_i^d w_i = 0$. The representations of of $w_1, \ldots, w_{d+1}$ give then the desired representation of the zero vector. Observe that we have also shown that in that case $C(d, m) \le C(d/2, m)^{\log d+1}$.

### 3.2 The algorithm when $\sqrt[d]{-1} \in \mathbb{F}_p$

We assume that $p - 1$ is divisible by $d$, we have a $d$th root $\mu$ of $-1$ as well as $\zeta_2, \ldots, \zeta_d$ in $\mathbb{F}_p$ at hand such that $\zeta_1 = 1, \zeta_2, \ldots, \zeta_d$ are a complete set of representatives of the cosets of $\mathbb{F}_p^{*d}$ in $\mathbb{F}_p^*$. To construct such elements $\mu, \zeta_2, \ldots, \zeta_d$ we need $\rho$th non-residues for any prime factor $\rho$ of $2d$. Such non-residues can be found in time polynomial in $\log p$ and $d$ by random choice or a deterministic search assuming GRH [12].

For $\ell = 1, \ldots, d$, put $B_\ell(d, m) = d^{\frac{\ell(\ell-1)}{2}}(m+1)^\ell$. For any $\ell$-tuple $\underline{a} = (a_1, \ldots, a_\ell) \in \{1, \ldots, d\}^\ell$, for $s \in \{1, \ldots, d\}$ and for $1 \le j \le \ell$, set $\underline{a}(j, s) = (a_1, \ldots, a_{j-1}, s, a_{j+1}, \ldots, a_\ell)$.

**Claim.** From $B = B_\ell(d, m)$ input vectors $v_1, \ldots, v_B$, in time polynomial in $B$ and $\log p$, we can can find $d^\ell$ pairwise disjoint subsets $J_{\underline{a}} \subseteq \{1, \ldots, B\}$ and field elements $\beta_1, \ldots, \beta_B$ such that $J_{(1,\ldots,\ell)} \ne \emptyset$, and if we set $w_{\underline{a}} = \sum_{i\in J_{\underline{a}}} \beta_i^d v_i$, then we have

$$\sum_{s=1}^{d} \zeta_s w_{\underline{a}(j,s)} = 0, \quad \text{for every } \underline{a} \in \{1, \ldots, d\}^\ell \text{ and } j = 1, \ldots, \ell.$$

**W.** e prove it by recursion on $\ell$. If $\ell = 1$ then any $B_\ell(d, m) = m + 1$ vectors from $\mathbb{F}_p^m$ are linearly dependent. Therefore there exist $\alpha_1, \ldots, \alpha_{m+1} \in \mathbb{F}_p$, not all zero, such that $\sum_{i=1}^{m+1} \alpha_i v_i = 0$. For $r = 1, \ldots, d$, let $J_r$ be the set of indices $i$ such that there exists $\beta_i \in \mathbb{F}_p^*$ with $\alpha_i = \zeta_r \beta_i^d$. For $i \in J_r$, such a $\beta_i$ can be efficiently found. At least one of the sets $J_r$ is non-empty. If $J_1$ is empty then we multiply the coefficients $\alpha_i$ simultaneously by $\zeta_1/\zeta_r^{-1}$ where $J_r$ is nonempty to arrange that $J_1$ becomes nonempty.

To describe the recursive step, assume that we are given $B_{\ell+1}(d,m) = d^\ell(m+1)B$ vectors. Put $E = d^\ell(m+1)$, and for convenience assume that the input vectors are denoted by $v_{ki}$, for $k = 1,\ldots,E$ and $i = 1,\ldots,B$. By the recursive hypothesis, for every $k \in \{1,\ldots,E\}$, there exist subsets $J_{\underline{a}}(k) \subseteq \{1,\ldots,B\}$ and field elements $\beta_i(k)$ such that $J_{(1,\ldots,\ell)}(k) \neq \emptyset$, and with $w_{\underline{a}}(k) = \sum_{i\in J_{\underline{a}}(k)} \beta_i(k)^d v_{ki}$, we have

$$\sum_{s=1}^{d} \zeta_s w_{\underline{a}(j,s)}(k) = 0, \tag{8}$$

for every $\underline{a} \in \{1,\ldots,d\}^\ell$ and $j = 1,\ldots,\ell$.

For every $k = 1,\ldots,E$, let $W(k)$ be the concatenation of the vectors $w_{\underline{a}}(k)$ in a fixed, say the lexicographic, order of $\{1,\ldots,d\}^\ell$. Then the $W(k)$'s are vectors of length $d^\ell m < E$. Therefore there exist field elements $\alpha_1,\ldots,\alpha_E$, not all zero, such that $\sum_{i=k}^{E} \alpha(k)W(k) = 0$. For a $k$ such that $\alpha(k) \neq 0$, let $\alpha(k) = \zeta_r\gamma(k)^d$ for some $1 \leq r \leq d$ and $\gamma(k) \in \mathbb{F}_p^*$. The index $r$ and $\gamma(k)$ can be computed efficiently. For $r = 1,\ldots,d$, let $M_r$ be the set of $k$'s such that $\alpha(k) = \zeta_r\gamma(k)^d$. We can arrange that $M_{\ell+1}$ is nonzero by simultaneously multiplying the $\alpha(k)$'s by $\zeta_{\ell+1}/\zeta_r$ for some $r$, if necessary. Observe that we have

$$\sum_{s=1}^{d} \zeta_s \sum_{k\in M_s} \gamma(k)^d W(k) = 0. \tag{9}$$

For $i \in \{1,\ldots,B\}$ and $k \in \{1,\ldots,E\}$ set $\beta'_{ki} = \gamma(k)\beta_i(k)$. We fix $\underline{a}' \in \{1,\ldots,d\}^{\ell+1}$, and we set $\underline{a} = (a'_1,\ldots a'_\ell)$ and $r = a'_{\ell+1}$. We define $J'_{\underline{a}'} = \{(k,i) : k \in M_r \text{ and } i \in J_{\underline{a}}(k)\}$ and $w'_{\underline{a}'} = \sum_{(k,i)\in J'_{\underline{a}'}} \beta'^d_{ki} v_{ki}$. Then $w'_{\underline{a}'} = \sum_{k\in M_r} \gamma_k^d w_{\underline{a}}(k)$. This equality, together with the equalities (8) imply that for every $j = 1,\ldots,\ell$, we have

$$\textstyle\sum_{s=1}^{d} \zeta_s w_{\underline{a}'(j,s)} = 0.$$

Equality (9) for $j-\ell+1$ gives $\sum_{s=1}^{d} \zeta_s \sum_{k\in M_s} \gamma(k)^d w_{\underline{a}}(k) = 0$. Expanding $w_{\underline{a}}(k)$ in the inner sum $\sum_{k\in M_s} \gamma(k)^d w_{\underline{a}}(k)$ gives that it equals $w_{\underline{a}'(\ell+1,s)}$. Thus also

$$\textstyle\sum_{s=1}^{d} \zeta_s w_{\underline{a}'(\ell+1,s)} = 0,$$

finishing the proof of the claim. □

We apply the procedure of the claim for $\ell = d$. From any $B = B_d(d,m) = d^{\frac{d(d-1)}{2}}(m+1)^d$ input vectors $v_1,\ldots,v_B$, we compute in time polynomial in $\log p$ and $B$ subsets $J_{\underline{a}}$, with $J_{(12\ldots d)} \neq \emptyset$, as well as nonzero elements $\beta_1,\ldots,\beta_B \in \mathbb{F}_p$ such that with $w_{\underline{a}} = \sum_{i\in J_{\underline{a}}} \beta_i^d v_i$, we have

$$\sum_{s=1}^{d} \zeta_s w_{\underline{a}(j,s)} = 0, \tag{10}$$

for every $j = 1,\ldots,d$ and for every $\underline{a} \in \{1,\ldots,d\}^d$.

Permutative tuples $\underline{a} \in S_d$ are of special interest. By $\mathrm{sgn}(\underline{a})$ we denote the *sign* of such a permutation, which is 1 if $\underline{a}$ is even and $-1$ if $\underline{a}$ is odd. We show that

$$\sum_{\underline{a}\in S_d} \mathrm{sgn}(\underline{a}) w_{\underline{a}} = 0. \tag{11}$$

For $\underline{a} \in S_d$, let $j_{\underline{a}}$ be the position of 1 in $\underline{a}$ and for every $s \in \{1, \ldots, d\}$, we denote by $\underline{a}[s]$ the sequence obtained from $a$ by replacing 1 with $s$. Notice that $\underline{a}[s] = \underline{a}(j_{\underline{a}}, s)$, therefore (10) implies

$$\sum_{\underline{a} \in S_d} \mathrm{sgn}(\underline{a}) \sum_{s=1}^{d} \zeta_s w_{\underline{a}[s]} = 0.$$

We claim that

$$\sum_{\underline{a} \in S_d} \mathrm{sgn}(\underline{a}) \sum_{s=2}^{d} \zeta_s w_{\underline{a}[s]} = 0.$$

To see this, observe that for $s > 1$ the tuple $\underline{a}[s]$ has entries from $\{2, \ldots, d\}$, where $s$ occurs twice, while the others once. Any such sequence $\underline{a}'$ can come from exactly two permutations which differ by a transposition: these are obtained from $\underline{a}'$ by replacing one of the occurrences of $s$ with 1. Then (11) is just the difference of the above two equalities.

For $i \in J_{\underline{a}}$, let $\gamma_i = 0$ if $\underline{a}$ is not a permutation, $\gamma_i = \beta_i$ if $\underline{a}$ is an even permutation and $\gamma_i = \mu\beta_i$ if $\underline{a}$ is an odd permutation. Then (11) gives $\sum_{i=1}^{B} \gamma_i^d v_i = 0$, the required representation of the zero vector. Observe that in that case $C(d, m) \leq d^{\frac{d(d-1)}{2}}(m+1)^d$. The bounds obtained in the two cases imply that $C(d, m) \leq d^{d^2 \log d}(m+1)^{d \log d}$ in general.

**Acknowledgements.** Research was supported in part by the Hungarian Scientific Research Fund (OTKA) Grant NK105645, the Singapore Ministry of Education and the National Research Foundation Tier 3 Grant MOE2012-T3-1-009, by the European Commission IST STREP project Quantum Algorithms (QALGO) 600700, and the French ANR Blanc Program Contract ANR-12-BS02-005.

## References

[1] Ajtai, M.: Generating hard instances of lattice problems. In: 28th annual ACM symposium on Theory of Computing (STOC), pp. 99–108, (1996)

[2] Bacon, D., Childs, A.M., van Dam, W.: From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In: 46th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 469–478, (2005)

[3] Chevalley, C.: Démonstration d'une hypothèse de M. Artin. Abh. Math. Sem. Hamburg 11, pp. 73–75 (1936)

[4] Childs, A.M., Schulman, L., Vazirani, U.: Quantum Algorithms for Hidden Nonlinear Structures. In: 48th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 395–404 (2007)

[5] Decker, T., Draisma, J., Wocjan, P.: Quantum algorithm for identifying hidden polynomial function graphs. Quantum Inf. Comput. 9, pp. 0215–0230 (2009)

[6] Decker, T., Høyer, P., Ivanyos, G., Santha, M.: Polynomial time quantum algorithms for certain bivariate hidden polynomial problems. Quantum Inf. Comput. 14, pp. 790–806 (2014)

[7] Decker, T., Ivanyos, G., Santha, M., Wocjan, P.: Hidden symmetry subgroup problems. SIAM J. Comput. 42, pp. 1987–2007 (2013)

[8] Denney, A., Moore, C. Russell, A.: Finding conjugate stabilizer subgroups in $PSL(2; q)$ and related groups. Quantum Inf. Comput. 10, pp. 282–291 (2010)

[9] Friedl, K., Ivanyos, G., Magniez, F., Santha, M., Sen, P.: Hidden translation and translating coset in quantum computing. SIAM J. Comput. 43, pp. 1–24 (2014)

[10] Grigni, M., Schulman, L., Vazirani M., Vazirani, U.: Quantum mechanical algorithms for the nonabelian Hidden Subgroup Problem. In: 33rd ACM Symposium on Theory of Computing (STOC), pp. 68–74 (2001)

[11] Hallgren, S., Russell, A., Ta-Shma, A.: Normal subgroup reconstruction and quantum computation using group representations. SIAM J. Comput. 32, pp. 916–934 (2003)

[12] Huang, M-D. A:. Riemann hypothesis and finding roots over finite fields. In: 17th annual ACM symposium on Theory of Computing (STOC), pp. 121–130, (1985)

[13] Ivanyos, G., Santha, M.: On solving systems of diagonal polynomial equations over finite fields. arXiv:1503.09016 [cs.CC]

[14] Ivanyos, G., Sanselme, L., Santha, M.: An efficient quantum algorithm for the hidden subgroup problem in nil-2 groups. Algoritmica 62, pp. 480–498 (2012)

[15] Jozsa, R.: Quantum factoring, discrete logarithms, and the hidden subgroup problem. Comput. Sci. Engin. 3, pp. 34–43 (2001).

[16] R. Karp. Reducibility among combinatorial problems. In.: Miller, R. (ed.) Complexity of Computer Computations, pp. 85-103, Springer, 1972.

[17] Kitaev, A. Y.: Quantum measurements and the Abelian Stabilizer Problem. arXiv:quant-ph/9511026v1 (1995)

[18] Kuperberg, G.: A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. SIAM J. Comput. 35, pp. 170–188 (2005)

[19] Moore, C., Rockmore, D., Russell, A., Schulman, L.: The power of basis selection in Fourier sampling: Hidden subgroup problems in affine groups. In: 15th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1113–1122 (2004)

[20] Papadimitriou, C.: On the complexity of the parity argument and other inefficient proofs of existence. J. Comput. System Sci., 48, pp. 498–532 (1994)

[21] Regev., O.: Quantum Computation and Lattice Problems. SIAM J. Comput. 33, pp. 738–760 (2004)

[22] Shanks., D.: Five number-theoretic algorithms. In: 2nd Manitoba Conference on Numerical Mathematics, pp. 51–70 (1972)

[23] Shor, P.: Algorithms for quantum computation: Discrete logarithm and factoring. SIAM J. Comput. 26, pp. 1484–1509 (1997)

[24] Sipser, M.: Introduction to the theory of computation. PWS Publishing Company (1997)

[25] van de Woestijne, C. E.: Deterministic equation solving over finite fields. PhD thesis, Universiteit Leiden (2006)

[26] Warning, E.: Bemerkung zur vorstehenden Arbeit von Herrn Chevalley. Abh. Math. Sem. Hamburg 11, pp. 76-83, 1936.