Annali di Matematica Pura ed Applicata (1923 -) (2023) 202:967–991
https://doi.org/10.1007/s10231-022-01267-x

On the equivalence of two post-quantum cryptographic families

Alessio Meneghetti[1] · **Alex Pellegrini**[2] · **Massimiliano Sala**[1]

Received: 10 February 2022 / Accepted: 18 August 2022 / Published online: 16 September 2022

Abstract
The maximum likelihood decoding problem (MLD) is known to be NP-hard and its complexity is strictly related to the security of some post-quantum cryptosystems, that is, the so-called code-based primitives. Analogously, the multivariate quadratic system problem (MQ) is NP-hard and its complexity is necessary for the security of the so-called multivariate-based primitives. In this paper we present a closed formula for a polynomial-time reduction from any instance of MLD to an instance of MQ, and viceversa. We also show a polynomial-time isomorphism between MQ and MLD, thus demonstrating the direct link between the two post-quantum cryptographic families.

Keywords Maximum likelihood decoding · Quadratic multivariate systems · Polynomial-time reductions · Code-based cryptography · Multivariate-based cryptography

Mathematics Subject Classification 94B35 · 11T06 · 68Q17

1 Introduction

Computationally difficult algebraic problems, e.g. the discrete logarithm problem (DLP) and the integer factorisation problem (IP), have historically been successfully exploited to construct secure cryptographic protocols, such as RSA, Diffie-Hellmann and ECDSA. An interested reader can find details on these schemes and the underlying algebraic problems

Alessio Meneghetti, Alex Pellegrini and Massimiliano Sala have contributed equally to this work.

✉ Alessio Meneghetti
alessio.meneghetti@unitn.it

Alex Pellegrini
a.pellegrini@tue.nl

Massimiliano Sala
massimiliano.sala@unitn.it

[1] Department of Mathematics, University of Trento, Trento, Italy

[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

in [1] or [2]. The advent of quantum algorithms, for example Shor's algorithm [3], has, however, incited several established primitives vulnerable to quantum attackers, thus encouraging researchers to design and analyse new families of cryptosystems based on NP-hard problems, since no quantum algorithm is known to be able to efficiently solve them. Post-Quantum Cryptography [4] refers to the the class of cryptographic primitives built upon computational problems not readily solvable by quantum computers. Among these, of particular interest we list lattice-based, code-based and multivariate-based cryptosystems, namely, systems whose security is intertwined with the computational complexity of solving problems over lattices, e.g. the shortest vector problem (SVP) or the closest vector problem (CVP) [5], problems based on coding theory, e.g. the Maximum Likelihood Decoding Problem (MLD) [6], and problems over polynomial ideals, e.g. the problem of deciding whether a quadratic Boolean polynomial system admits a solution, usually referred to as Multivariate Quadratic Problem (MQ) [7].

This massive research effort is still ongoing, as can be noticed by the wide participation to the NIST standardisation process for post-quantum primitives[1]. The round-3 finalists for key-encapsulation mechanisms (KEM) are Classic McEliece [8], CRYSTALS-KYBER [9], NTRU [10] and SABER [11], while the round-3 finalists for digital signature schemes are CRYSTALS-DILITHIUM [12], FALCON [13] and Rainbow [14]. Notably, the only round-3 finalists which are not lattice-based are Classic McEliece and Rainbow, which are, respectively, code-based and multivariate-based.

As briefly stated above, code-based schemes are designed by exploiting computational and decision problems obtained by questions arising from Coding Theory, such as for example the Maximum-Likelihood Decoding problem (MLD) for linear codes. Using the words of Guruswami and Vardy, "MLD is one of the central (perhaps, the central) algorithmic problems in coding theory" [15]. This problem, given here in Definition 1, has been proven to be NP-complete in 1978 by Berlekamp, McEliece and van Tilborg in [6], where the authors provide a reduction from the Three-Dimensional Matching problem for graphs [16]. Then, in 1990, MLD was proven to be in P/Poly by Bruck and Naor [17] and by Lobstein [18]. Other interesting results on the complexity of MLD were then given by several authors, and an interested reader can find more information in [6, 19] for the general case and in [15, 20–22] for specific classes of codes.

In the context of post-quantum code-based cryptography, the most famous cryptosystem is that proposed by McEliece in 1978 [23]. This scheme has been studied for over forty years, proving its resilience and security, and it was then used, together with the Niederreiter scheme [24], as a building block for the NIST round-3 finalist Classic McEliece. We remark that the security of each of these primitives is related to both MLD and the problem of distinguishing between apparently-random codes and permuted versions of algebraic codes. In this work, we focus on the first of these two problems. A presentation of code-based cryptography and the underlying problems can be found in the chapter *Code-based cryptography* by Overbeck and Sendrier in [4].

Before stating MLD, we provide some basics on binary linear codes. An $[n, k]$ linear code C is a k dimensional subspace of $\left(\mathbb{F}_q\right)^n$. The parameters n and k are called the *length* and *dimension* of the code. Since we are interested in binary codes, throughout this paper we only consider the case $q = 2$, hence with $\mathbb{F}$ we mean $\mathbb{F}_2$ and we often omit the term binary whenever we use the term code. A *codeword* is any vector in the

[1] https://csrc.nist.gov/projects/post-quantum-cryptography, accessed on 2022-01-18.

code. A *generator matrix* G of C is a $k \times n$ matrix whose rows span C. Similarly, a *parity-check matrix* of C is a generator matrix of the dual of C. Due to this definition, a vector c is a codeword if and only if $H \cdot c^\top = 0$. We recall that if G is systematic, i.e. $G = \left[I_k \mid R\right]$, then $H = \left[-R^T \mid I_{n-k}\right]$. The (Hamming) *weight* of a vector v is the number $\mathrm{w}(v)$ of its nonzero components.

Definition 1 (MLD) Let $H = [h_{i,j}]_{i=1,\dots m, j=1,\dots,n}$ be an $m \times n$ binary matrix, let $s \in \mathbb{F}^m$ and let $t \leq n$ be a positive integer. Decide whether there is a vector $v \in \mathbb{F}^n$ of weight at most t, such that $Hv^\top = s^\top$.

We denote with I_{MLD} a generic instance of MLD, determined by the triple $I_{\mathrm{MLD}} = (H, s, t)$. The values (n, m) determine the memory space required to store I_{MLD}, indeed, we need a total of $nm + m + \lfloor \log_2 n \rfloor + 1$ bits to write a given instance (H, s, t). We call (n, m) the complexity parameters of I_{MLD}, and $|I_{\mathrm{MLD}}| = nm + m + \lfloor \log_2 n \rfloor + 1$ the size of I_{MLD}. We can assume that $m \leq n$, since this case has the same hardness as the general case.

The second problem we consider is to decide whether a multivariate-quadratic Boolean system admits a solution. This problem, known as the "multivariate quadratic equation system problem" (MQ) is linked to the security of multivariate-based cryptosystems (e.g. Oil and Vinegar [25], Rainbow [14], GeMSS [26]).

Let I be an ideal of a polynomial ring $\mathcal{R}$ over a field $\mathbb{K}$ and let $\mathbb{E}$ be an extension field of $\mathbb{K}$, we denote by

$$\mathcal{V}_{\mathbb{E}}(I) = \{A \in \mathbb{E}^{\mathsf{n}} \mid f(A) = 0 \ \forall f \in I\}$$

the set of all the zeroes of I in $\mathbb{E}^{\mathsf{n}}$. $\mathcal{V}_{\mathbb{E}}(I)$ is called the *variety* of I over $\mathbb{E}$. An MQ-system of equations over $\mathbb{F}$ is a set of m polynomial equations of degree at most 2 in $\mathbb{F}[x_1, \dots, x_{\mathsf{n}}]$ of the form:

$$S = \begin{cases} f_1(x_1, \dots, x_{\mathsf{n}}) = 0 \\ f_2(x_1, \dots, x_{\mathsf{n}}) = 0 \\ \vdots \\ f_{\mathsf{m}}(x_1, \dots, x_{\mathsf{n}}) = 0 \end{cases} \tag{1}$$

where for every $h \in \{1, \dots, \mathsf{m}\}$

$$f_h(x_1, \dots, x_{\mathsf{n}}) = \sum_{1 \leq i < j \leq \mathsf{n}} \gamma_{ij}^{(h)} x_i x_j + \sum_{1 \leq i \leq \mathsf{n}} \lambda_i^{(h)} x_i + \delta^{(h)} \tag{2}$$

with $\gamma_{ij}^{(h)}, \lambda_i^{(h)}, \delta^{(h)} \in \mathbb{F}$. The *(decision) multivariate quadratic equation system problem* (MQ) can now be stated as

Definition 2 (MQ) Consider a polynomial system $S = \{f_1, \dots, f_{\mathsf{m}}\}$ as in (1) of degree at most 2 over $\mathbb{F}$ and let I be the ideal generated by S.

Decide whether $\mathcal{V}_{\mathbb{F}}(I)$ is non-empty.

We denote with I_{MQ} a generic instance of MQ, determined by the polynomial system S. Similarly to the case of MLD, the values (n, m) determine the memory space required to

store an instance I_{MQ}. In this case, we need a total of $\mathsf{m}\left(\binom{\mathsf{n}}{2}+\mathsf{n}+1\right)$ bits to write S. We call (n,m) the complexity parameters of I_{MQ}, and $|I_{\mathrm{MQ}}| = \mathsf{m}\left(\binom{\mathsf{n}}{2}+\mathsf{n}+1\right)$ its size.

MQ has been proven to be NP-hard over any field [7], and many cryptosystems rely their security on such problem [25, 27, 28]. Several mathematical approaches have been employed to tackle this problem, such as the Newton and the tensor-based algorithms [29, 30], Gröbner bases, resultants and eigenvalues/eigenvectors of companion matrices [31], semidefinite relaxations [32–34], numerical homotopy [35, 36], low-rank matrix recovery [37], and symbolic computation [38].

The most established method to perform cryptanalysis of public-key systems is to focus on the algebraic problems underlying them. In this work we look at it from a slightly different perspective, establishing a link between MLD and MQ, and thus providing new directions in the analysis of both code-based and multivariate-based primitives. More precisely, the aim of this paper is to show explicit reductions between the two previous problems. Since both are NP-complete problems, one might be reduced to the other, but it is not obvious how to do it explicitly without losing their algebraic nature.

MQ and MLD are problems of a purely algebraic nature, naturally stated and studied in the context of vector spaces and polynomial rings over $\mathbb{F}_2$, the smallest possible field. Interestingly, their complexity and the complexity of the numerous related problems (including search problems) is at the heart of research for the mathematical community working in coding theory and cryptography. Yet, known results about their complexity are obtained via techniques of a rather different nature, such as graph theory. As far as we know, this is the first paper that investigates their direct explicit complexity links, using only languages and tools familiar to standard research in coding theory and cryptography alike. To be more precise, we will establish in Sect. 3 an explicit reduction from MLD to MQ, while in Sect. 4 we will present a reduction from MQ to MLD. The remainder of this paper contains Sect. 2, where we provide preliminaries and our notation, Sect. 6, where we leave some open problems, and notably Sect. 5. In this latter section we draw some significant conclusions, among which we report our proof of the existence of a polynomial-time isomorphism between NP and MQ, and thus between code-based and multivariate-based primitives.

2 Preliminary results and definitions

Before introducing the reductions between MLD and MQ we need some preliminary notation. Throughout this paper we consider vectors to be row vectors, unless otherwise specified. Moreover, we denote with $\bar{\cdot}$ each element of a set which is not a variable (regardless it being an element of fields or vector spaces), whereas without the notation $\bar{\cdot}$ we mean variables. As an example, if $f(x)$ is a polynomial in the variable x then $f(\bar{x})$ is to be considered as the evaluation of f at the point $\bar{x}$. In Sect. 4, we will also use the notation $\hat{\cdot}$ and $\widetilde{\cdot}$ instead of $\bar{\cdot}$ to distinguish between elements belonging to distinct spaces. As an example, in Sect. 4 we will define two distinct sets $\widehat{\Sigma}$ and $\widetilde{\Sigma}$, which are somewhat linked. In this case, to distinguish between their elements, we will use $\hat{v} \in \widehat{\Sigma}$ and $\widetilde{v} \in \widetilde{\Sigma}$.

Let l be a positive integer. We define the map $\mathtt{int} : \mathbb{F}^l \to \mathbb{Z}$ as

$$\texttt{int}(a) = \texttt{int}\big(\big(a_1, \ldots, a_l\big)\big) := \sum_{j=1}^{l} a_j \cdot 2^{j-1} \ ,$$

where the sum on the right-hand side is over the integers. In this way, $\texttt{int}(\bar{a})$ is the integer value corresponding to the input vector of bits $\bar{a} = (\bar{a}_1, \ldots, \bar{a}_l) \in \mathbb{F}^l$. When we regard a vector in $\mathbb{F}^l$ as the binary representation of an integer, we list its bits from the least-significant to the most-significant, e.g. if $l = 4$ then $3 = (1, 1, 0, 0)$. From now on we will often use the binary representation of the parameter t of an MLD instance (H, s, t) as a vector of $\ell = \lfloor \log_2 n \rfloor + 1$ bits $(t_1, \ldots, t_\ell)$, and therefore we will use both $\mathrm{w}(v) \leq t$ and $\mathrm{w}(v) \leq \texttt{int}(t_1, \ldots, t_\ell)$.

Let $\mathbf{1}$ be the vector $(1, \ldots, 1)$. We define the map $\texttt{increase} : \mathbb{F}^l \to \mathbb{F}^l$ such that, for all $\bar{a} \neq \mathbf{1}$, we have

$$\texttt{int}(\texttt{increase}(\bar{a})) = \texttt{int}(\bar{a}) + 1 \ ,$$

namely, if $\bar{a} = \big(\bar{a}_1, \ldots, \bar{a}_z, \ldots, \bar{a}_l\big)$, where $\bar{a}_z = 0$ is the left-most 0 bit of $\bar{a}$, then

$$\texttt{increase}(\bar{a}) = \begin{cases} \big(0, \ldots, 0, 1, \bar{a}_{z+1}, \ldots, \bar{a}_l\big) & \text{if } \bar{a} \neq \mathbf{1} \\ \bar{a} & \text{if } \bar{a} = \mathbf{1} \ . \end{cases}$$

Finally, we introduce the projection and truncation maps π and τ. Let $i \leq l$ be a non-negative integer, and let $\pi_i : \mathbb{F}^l \to \mathbb{F}^l$ the projection defined as

$$\pi_i\big(\bar{v}_1, \ldots, \bar{v}_l\big) = \big(\bar{v}_1, \ldots, \bar{v}_i, 0, \ldots, 0\big)$$

for $1 \leq i \leq l$ and

$$\pi_0\big(\bar{v}_1, \ldots, \bar{v}_l\big) = (0, \ldots, 0) \ .$$

Similarly, let $i \leq l$ be a positive integer. We define the truncation $\tau_i : \mathbb{F}^l \to \mathbb{F}^i$ as

$$\tau_i\big(\bar{v}_1, \ldots, \bar{v}_l\big) = \big(\bar{v}_1, \ldots, \bar{v}_i\big) \ .$$

Consider now a polynomial equation $f = 0$, where $f \in \mathbb{F}[x_1, \ldots, x_l]$ with $\deg(f) = d > 2$. The goal of the algorithm described in the following fact is to reduce this equation to a set of equations of degree at most 2. We use an idea similar to that described by Kipnis and Shamir [39] in their *relinearization* technique.

Fact 1 Let $x_{i_1} x_{i_2} \cdots x_{i_d}$ be a monomial with degree d. We introduce a set of new $d - 2$ variables as follows

$$\begin{cases} y_1 = x_{i_1} x_{i_2} \\ y_2 = y_1 x_{i_3} \\ \vdots \\ y_{d-2} = y_{d-3} x_{i_{d-1}} \end{cases}$$

and thus rewrite $x_{i_1} x_{i_2} \cdots x_{i_d}$ as $y_{d-2} x_{i_d}$. With this procedure, a monomial of degree d is substituted by a set of $d - 1$ quadratic equations by introducing $d - 2$ variables.

By applying the same argument to each monomial of f, we obtain a system of quadratic equations, as required.

For convenience, given a polynomial equation $f = 0$ we denote with $\texttt{quadr}(f)$ the quadratic polynomial system obtained by applying the procedure in Fact 1 to f. Similarly, given a polynomial system S, we denote with $\texttt{quadr}(S)$ the quadratic system obtained by applying the procedure to each of the polynomials in S and joining all the systems of equations, formally

$$\texttt{quadr}(S) = \bigcup_{f \in S} \texttt{quadr}(f). \tag{3}$$

Example 1 Let $S = \{f_1, f_2\}$ be the polynomial system

$$S = \begin{cases} f_1 = x_1x_2x_4 + x_1x_3x_4 + x_2x_3 + x_1 = 0 \\ f_2 = x_1x_2x_3x_4 + 1 = 0 \end{cases}$$

To compute $\texttt{quadr}(S)$ we start with computing $\texttt{quadr}(f_1)$. There are two monomials of degree larger than 2 in f_1, namely $x_1x_2x_4$ and $x_1x_3x_4$, which have degree $d_1 = 3$. When dealing with a monomial with degree $d_1 = 3$, as stated in Fact 1, we introduce $d_1 - 2 = 1$ variable and then we obtain a set of $d_1 - 1 = 2$ quadratic equations (including the rewriting of the starting equation $f_1 = 0$ in terms of the new variables).

We start from $x_1x_2x_4$, we call the new variable y_1, and we create the quadratic equation $y_1 = x_1x_2$. Due to this new equation, we can substitute x_1x_2 with y_1 into $f_1 = 0$, obtaining

$$f_1 = 0 \quad \Leftrightarrow \quad \begin{cases} y_1 = x_1x_2 \\ y_1x_4 + x_1x_3x_4 + x_2x_3 + x_1 = 0 \end{cases}$$

Similarly, we deal with the monomial $x_1x_3x_4$ by introducing $y_2 = x_1x_3$. The system becomes

$$\texttt{quadr}(f_1) = \begin{cases} y_1 = x_1x_2 \\ y_2 = x_1x_3 \\ y_1x_4 + y_2x_4 + x_2x_3 + x_1 = 0 \end{cases}$$

We proceed now with the second equation. In f_2 only $x_1x_2x_3x_4$ has degree larger than 2, and in this case we need to introduce two variables z_1, z_2 and transform $f_2 = 0$ into a system of 3 quadratic equations:

$$\texttt{quadr}(f_1) = \begin{cases} z_1 = x_1x_2 \\ z_2 = z_1x_3z_2x_4 + 1 = 0 \end{cases}$$

Thus $\texttt{quadr}(S) = \texttt{quadr}(f_1) \cup \texttt{quadr}(f_2)$ is a quadratic system equivalent to S.

Definition 3 A system of equations is said to be in *standard form* if

- it contains equations of the form $xy + z = 0$ which do not share any variable;
- it contains linear equations with up to three monomials, that is, of the form $x + \bar{\delta} = 0$, $x + y + \bar{\delta} = 0$ or $x + y + z + \bar{\delta} = 0$, with $\bar{\delta} \in \mathbb{F}$;
- each variable that appears in a linear equation appears also in exactly one quadratic equation;
- it does not contain any other kind of equation.

In Lemma 1, we will show that any system of quadratic equations can be brought to standard form by adding (a bounded number of) new variables and equations. A preliminary result for the case of linear equations is the following:

Fact 2 Let us consider a linear equation with l variables

$$x_1 + x_2 + x_3 + \ldots + x_l = 0\,.$$

If we define y_i to be the sum of the the first i variables $x_1 + \ldots + x_i$, then $x_1 + x_2 + x_3 + \ldots + x_l = 0$ is equivalent to the linear system

$$\begin{cases} x_1 + x_2 + y_2 = 0 \\ y_2 + x_3 + y_3 = 0 \\ y_3 + x_4 + y_4 = 0 \\ \vdots \\ y_{l-3} + x_{l-2} + y_{l-2} = 0 \\ y_{l-2} + x_{l-1} + x_l = 0 \end{cases}$$

which has $l - 2$ equations, each one involving exactly three variables, and a total of $2l - 3$ variables.

2.1 Our computational model

To establish reductions' performances between MLD and MQ we need a computational model that defines the computational cost of operations performed in the reduction. In our case, the set of operations we need in order to perform a reduction are the sum and multiplication in the finite field $\mathbb{F}$. The two operations can be identified with OR and AND logical operators, respectively, to which we assign cost 1. We also need to carefully consider the memory requirements of our method, so we assign cost 1 to every coefficient required to express a single polynomial. Notice that the number of bits required to completely define a single quadratic polynomial is approximately the square of the number of variables.

In Sect. 3 we will present a reduction $\alpha : \text{MLD} \to \text{MQ}$, whose memory analysis will be based on the number of equations and variables required to describe the related polynomials (i.e. the complexity parameters of the resulting MQ instance). In Sect. 4 we present a reduction $\beta : \text{MQ} \to \text{MLD}$ whose memory analysis is based on the size of the generated parity-check matrix in the reduction (i.e. the complexity parameters of the resulting MLD instance).

The following lemma, together with Fact 2, implies that every instance of MQ can be reduced to an instance in which the polynomial system is given in standard form.

Lemma 1 *Consider a polynomial system* $S = \{f_1, \ldots, f_{\mathsf{m}}\}$ *with* $f_i \in \mathbb{F}[x_1, \ldots, x_{\mathsf{n}}]$ *and* $\deg(f_i) = 2$ *for each* $i = 1, \ldots, \mathsf{m}$. *S can be taken to standard form in* $\mathcal{O}(\mathsf{mn}^2)$ *operations.*

More precisely, the number of quadratic equations is at most $\mathsf{m}\left(\frac{\mathsf{n}(\mathsf{n}-1)}{2}\right)$ *and the number of linear equations is at most* $\mathsf{m}\left(\frac{3\mathsf{n}^2-\mathsf{n}}{2} - 2\right)$.

Proof Assume first $S = \{f\}$. In the worst case

$$f = \sum_{i<j\in\{1,\ldots,\mathsf{n}\}} x_i x_j + \sum_{i\in\{1,\ldots,\mathsf{n}\}} x_i + \bar{\delta}\ ,$$

with $\bar{\delta} \in \mathbb{F}$. Clearly, f has $\binom{\mathsf{n}}{2} = \frac{\mathsf{n}(\mathsf{n}-1)}{2}$ quadratic monomials $x_i x_j$. As a first step, we introduce $\binom{\mathsf{n}}{2}$ new variables, along with a set of $\binom{\mathsf{n}}{2}$ equations of the form $x_{ij} + x_i x_j = 0$. These newly introduced variables are then substituted into f, obtaining in this way a linear polynomial f' with $\frac{\mathsf{n}(\mathsf{n}+1)}{2}$ variables.

Notice that the quadratic equations we just introduced share some variables x_i: for each i, x_i appears indeed in $\mathsf{n}-1$ such equations. However, our aim is to have variables in degree-2 monomials appearing in exactly one monomial. This is achieved in a second step by introducing new variables and new linear equations: if x_i appears in both $x_{1,i} + x_1 x_i = 0$ and $x_{2,i} + x_2 x_i = 0$ then we define a new variable x'_i and write

$$\begin{cases} x_{1,i} + x_1 x_i &= 0 \\ x_{2,i} + x_2 x'_i &= 0 \\ x_i + x'_i = & 0 \quad . \end{cases}$$

By doing this for all shared variables, we add a set of $\mathsf{n}(\mathsf{n}-1)$ linear equations of the form $x'_i + x_i = 0$, each one introducing a new variable.

With the first step we have produced a linear polynomial f' with $\frac{\mathsf{n}(\mathsf{n}+1)}{2}$ variables. However, for the system to be in standard form, each linear equation has to involve at most three variables. As in Fact 2, this can be done in a third step by substituting f' with $\frac{\mathsf{n}(\mathsf{n}+1)}{2} - 2$ linear equations involving each one three variables, for a total of $2\left(\frac{\mathsf{n}(\mathsf{n}+1)}{2}\right) - 3$ variables.

We end up with a set of $\frac{\mathsf{n}(\mathsf{n}-1)}{2}$ quadratic equations and a set of $\frac{3\mathsf{n}^2-\mathsf{n}}{2} - 2$ linear equations. The total number of variables is $\frac{5\mathsf{n}^2-\mathsf{n}}{2} - 3$.

Now let S contain m equations. We perform the same transformation as above for each of them; however, the sets of quadratic equations of the polynomials might share variables. To solve this problem, we rename the variables of each quadratic polynomial: if the h-th polynomial contains the variable x_k then we substitute x_k with a new variable $X_{h,k}$. We also need to track this substitution, therefore add a new set of linear equations $X_{h,k} + x_k = 0$ for $h = 1, \ldots, \mathsf{m}$ and $k = 1, \ldots, \mathsf{n}$. In this way, the quadratic equations do not share variables, and we can substitute each one of them with a system in standard form, as we did in the first part of this proof. As a consequence, the total number of quadratic equations is bounded by $\mathsf{m}\left(\frac{\mathsf{n}(\mathsf{n}-1)}{2}\right)$ and the number of linear equations is at most $\mathsf{m}\left(\frac{3\mathsf{n}^2-\mathsf{n}}{2} - 2\right)$. Similarly, the number of variables is bounded by $\mathsf{m}\left(\frac{5\mathsf{n}^2-\mathsf{n}}{2} - 3\right)$.

Putting everything together, we obtain $\mathcal{O}(\mathsf{m}\mathsf{n}^2)$ new variables and equations. □

3 MLD to MQ reduction

In this section, we provide an explicit reduction α, which maps an instance I_{MLD} of the MLD problem to an instance I_{MQ} of the MQ problem. More precisely, for any pair of complexity parameters (n, m) we are going to define a reduction $\alpha_{n,m}$, which deals with binary codes of length n and dimension at least $n - m$.

An MLD instance $I_{\mathrm{MLD}} = (\bar{H}, \bar{s}, \bar{t})$ can be thought of as the union of two requirements:

1. *parity-check constraint*; the solution $\bar{v}$ has to satisfy $\bar{H}\bar{v}^\top = \bar{s}^\top$;
2. *weight constraint*; the solution $\bar{v}$ has to satisfy $\mathrm{w}(\bar{v}) \leq \bar{t}$. To obtain our reduction, we split this constraint in two parts: $\mathrm{w}(\bar{v}) = w$ and $w \leq \bar{t}$.

We propose three encodings, each one parametrised by the complexity parameters m and n of MLD, which together correspond to a reduction from MLD to MQ. Here the term *encoding* has nothing to do with the mapping of a message to a codeword, instead it is the the rewriting of a constraint in terms of quadratic equations. The set of quadratic Boolean polynomials $\mathtt{pcce}_{n,m}$ is the encoding of the parity-check constraint $\bar{H}v^\top = \bar{s}^\top$. A complete description is provided in Sect. 3.1. The polynomial system $\mathtt{hwce}_{n,m}$, detailed in Sect. 3.2, corresponds to the Hamming weight computation of $\bar{v}$. The third encoding $\mathtt{wce}_{n,m}$, in Sect. 3.3, is a polynomial system corresponding to the weight constraint $\mathrm{w}(v) \leq \bar{t}$.

We define the map $\alpha_{n,m} = \mathtt{pcce}_{n,m} \cup \mathtt{hwce}_{n,m} \cup \mathtt{wce}_{n,m}$, where we mean that, given a specific instance $I_{\mathrm{MLD}} = (\bar{H}, \bar{s}, \bar{t})$, the actual reduction is given by $I_{\mathrm{MQ}} = \alpha_{n,m}(I_{\mathrm{MLD}}) = \mathtt{pcce}_{n,m}(\bar{H}, \bar{s}, \bar{t})$ $\cup \mathtt{hwce}_{n,m}(\bar{H}, \bar{s}, \bar{t}) \cup \mathtt{wce}_{n,m}(\bar{H}, \bar{s}, \bar{t})$.

Observe that $\alpha_{n,m}$ is a system of polynomial equations depending only on n and m, and whose evaluation on a specific MLD instance gives us an MQ instance.

3.1 Parity-check constraint encoding

We claim that the parity-check matrix constraint $Hv^\top = s$ is equivalent to a set of m linear equations corresponding to polynomials

$$f_i \in \mathbb{F}[h_{i,1}, \ldots, h_{i,n}, v_1, \ldots, v_n, s_i]$$

of the form $f_i = \sum_{j=1}^{n} h_{i,j} v_j + s_i$. Indeed, $\bar{H}\bar{v}^\top = \bar{s}$ if and only if $f_i(\bar{H}, \bar{v}, \bar{s}) = 0$ for every $1 \leq i \leq m$. Thus, we can define $\mathtt{pcce}_{n,m}$ as

$$\{f_i = 0\}_{i=1,\ldots,m},$$

and so

$$\mathtt{pcce}_{n,m}(\bar{H}, \bar{s}, \bar{t}) = \{f_i(\bar{H}, v, \bar{s}) = 0\}_{i=1,\ldots,m}.$$

Observe that $f_i(\bar{H}, v, \bar{s})$ belongs to $\mathbb{F}[v_1, \ldots, v_n]$.

We explicitly state the following trivial result for completeness.

Lemma 2 *Let I_{MLD} be an instance with complexity parameters n and m. Then, $\mathtt{pcce}_{n,m}(I_{\mathrm{MLD}})$ contains m linear equations in n variables.*

3.2 Weight-computation encoding

Let $\bar{v} \in \mathbb{F}^n$ and $\ell = \lfloor \log_2(n) \rfloor + 1$, so that the weight of a length-n vector can be written as a length-ℓ vector. For $i = 0, \ldots, n$ and $j = 1, \ldots, \ell$ we want to define some functions $a^{(i)} : \mathbb{F}^n \to \mathbb{F}^\ell$ and their component functions $a^{(i)}(\bar{v}) = (a_1^{(i)}(\bar{v}), \ldots, a_\ell^{(i)}(\bar{v})) \in \mathbb{F}^\ell$. We set $a^{(0)}(\bar{v}) = (0, \ldots, 0)$ for any $\bar{v}$ (for convenience), and we define $a^{(i)}$ recursively for $i = 1, \ldots, n$ by computing its coefficients $a_j^{(i)}$ as the polynomials in $\mathbb{F}[v_1, \ldots, v_n] = \mathbb{F}[v]$

$$a_j^{(i)}(v) = a_j^{(i-1)}(v) + \left(\prod_{h=1}^{j-1} a_h^{(i-1)}(v) \right) v_i \, .$$

The following lemma and theorem prove that $a^{(i)}(\bar{v})$ is the implementation of a counter that stores the Hamming weight of $\bar{v}$ until the *i-th* coordinate.

Lemma 3 *Let* $0 \leq k \leq n-1$. *If* $\bar{v}_{k+1} = 1$ *then* $a^{(k+1)}(\bar{v}) = \texttt{increase}(a^{(k)}(\bar{v}))$. *If* $\bar{v}_{k+1} = 0$ *then* $a^{(k+1)}(\bar{v}) = a^{(k)}(\bar{v})$.

Proof Let $a_z^{(k)}(\bar{v})$ be the leftmost 0 in $a^{(k)}(\bar{v})$, $1 \leq z \leq \ell$, and let $\Lambda \in \mathbb{N}$. Observe that

$$\prod_{h=1}^{\Lambda} a_h^{(k)} = \begin{cases} 1 & \text{if } \Lambda < z \\ 0 & \text{otherwise.} \end{cases}$$

We start with the case $\bar{v}_{k+1} = 1$ and we compute $a_j^{(k+1)}(\bar{v})$ according to the value of j.

- if $1 \leq j \leq z-1$ then $a_j^{(k+1)}(\bar{v}) = a_j^{(k)}(\bar{v}) + \left(\prod_{h=1}^{j-1} a_h^{(k)}(\bar{v}) \right) \bar{v}_{k+1} = 1 + (1) \cdot 1 = 0$;
- if $j = z$ we have $a_z^{(k+1)}(\bar{v}) = a_z^{(k)}(\bar{v}) + \left(\prod_{h=1}^{z-1} a_h^{(k)}(\bar{v}) \right) \cdot 1 = 0 + (1) \cdot 1 = 1$;
- if $j \geq z+1$ then $a_j^{(k+1)}(\bar{v}) = a_j^{(k)}(\bar{v}) + \left(\prod_{h=1}^{j-1} a_h^{(k)}(\bar{v}) \right) \cdot 1 = a_j^{(k)}(\bar{v}) + (0) \cdot 1 = a_j^{(k)}(\bar{v})$.

The procedure we described flips every 1-bit (until the $(z-1)$-th bit) and the first 0-bit, while leaving all the other bits unchanged, which is the behaviour of the function `increase()`.

The second possible case is $\bar{v}_{k+1} = 0$. Observe that, regardless of j, the value $a_j^{(k+1)}(\bar{v})$ is given by

$$a_j^{(k+1)}(\bar{v}) = a_j^{(k)}(\bar{v}) + \left(\prod_{h=1}^{j-1} a_h^{(k)}(\bar{v}) \right) \bar{v}_{k+1} \, ,$$

which, since $\bar{v}_{k+1} = 0$, is simply

$$a_j^{(k+1)}(\bar{v}) = a_j^{(k)}(\bar{v}) + \left(\prod_{h=1}^{j-1} a_h^{(k)}(\bar{v}) \right) \cdot 0 = a_j^{(k)}(\bar{v}) \, .$$

□

Theorem 4 *Let $\bar{v} \in \mathbb{F}^n$ and let $1 \leq i \leq n$. Then*

$$\mathrm{w}(\pi_i(\bar{v})) = \mathtt{int}\left(a_1^{(i)}(\bar{v}), \ldots, a_\ell^{(i)}(\bar{v})\right).$$

Proof We proceed by induction on i. Let $i = 1$, then $a_1^{(1)}(\bar{v}) = a_1^{(0)}(\bar{v}) + \bar{v}_1 = \bar{v}_1$, while $a_j^{(1)}(\bar{v}) = 0 + 0 \cdot 0 = 0$ for every $j = 2, \ldots, \ell$. Therefore $\mathtt{int}(a^{(1)}(\bar{v})) = \mathtt{int}(\bar{v}_1, 0, \ldots, 0) = \mathrm{w}(\pi_1(\bar{v}))$.

Assume that $\mathtt{int}(a^{(i)}(\bar{v})) = \mathrm{w}(\pi_i(\bar{v}))$ holds for a certain value of $1 \leq i \leq n-1$, then we are going to prove $\mathtt{int}(a^{(i+1)}(\bar{v})) = \mathrm{w}(\pi_{i+1}(\bar{v}))$. We have two cases: either $\bar{v}_{i+1} = 0$ or $\bar{v}_{i+1} = 1$.

1. If $\bar{v}_{i+1} = 0$ then by Lemma 3 $a^{(i+1)}(\bar{v}) = a^{(i)}(\bar{v})$, in accordance with $\pi_{i+1}(\bar{v}) = \pi_i(\bar{v})$.
2. If $\bar{v}_{i+1} = 1$ then, by Lemma 3, we have $a^{(i+1)}(\bar{v}) = \mathtt{increase}(a^{(i)}(\bar{v}))$ and therefore $\mathtt{int}(a^{(i+1)}(\bar{v})) = \mathtt{int}(\mathtt{increase}(a^{(i)}(\bar{v}))) = \mathtt{int}(a^{(i)}(\bar{v})) + 1 = \mathrm{w}(\pi_i(\bar{v})) + 1 = \mathrm{w}(\pi_{i+1}(\bar{v}))$, where the last equality comes from obvious properties of weights.

□

The following corollary states that the Hamming weight of a vector $\bar{v} \in \mathbb{F}^n$ is exactly the integer represented by the vector $\left(a_1^{(n)}(\bar{v}), \ldots, a_\ell^{(n)}(\bar{v})\right)$.

Corollary 5 *Let $\bar{v} \in \mathbb{F}^n$ then*

$$\mathrm{w}(\bar{v}) = \mathtt{int}\left(a_1^{(n)}(\bar{v}), \ldots, a_\ell^{(n)}(\bar{v})\right) \tag{4}$$

Proof It follows from Theorem 4 by noticing that $\pi_n(\bar{v}) = (\bar{v}_1, \ldots, \bar{v}_n) = \bar{v}$. □

Observe that, for $1 \leq j \leq \ell$ and for any $1 \leq i \leq n$, Formula (4) is an equation of degree j of the form $x + y + M = 0$ where M is a degree-j monomial. By using the procedure described in Remark 1 each equation can be reduced to a system of less than ℓ quadratic equations by adding less than ℓ new variables. Indeed, $\mathtt{quadr}(x + y + M = 0)$ is a quadratic system with $j - 1 < \ell$ equations in $j - 2 < \ell$ variables.

By performing this procedure to each of $n \cdot \ell$ equations we obtain a system of quadratic equations. We remark that many variables were shared between equations, so there are several possible optimisations to be applied instead of applying $\mathtt{quadr}()$ to each Eq. (4). However, since the degree of each Eq. (4) is bounded by ℓ, even without optimising the procedure, we end up with a system with a manageable number of quadratic equations, as stated by the following lemma.

Lemma 6 *The number of quadratic equations in*

$$\mathtt{quadr}\left(\left\{a_j^{(i)} = a_j^{(i-1)} + \left(\prod_{h=1}^{j-1} a_h^{(i-1)}\right) v_i\right\}_{i=1,\ldots,n,\ j=1,\ldots,\ell}\right)$$

is $\mathcal{O}(n\ell^2)$. The total number of variables is $\mathcal{O}(n\ell^2)$.

Proof Each of the $n\ell$ equations is transformed into a set of less than ℓ quadratic equations by adding less than ℓ variables. □

We are finally ready to construct the set of equations corresponding to the (Hamming) weight-computation encoding, which contains the polynomials corresponding to the weight-computation procedure depicted in this section. We define

$$\texttt{hwce}_{n,m} = \texttt{quadr}\left(\left\{ a_j^{(i)} = a_j^{(i-1)} + \left(\prod_{h=1}^{j-1} a_h^{(i-1)}\right) v_i \right\}_{i=1,\ldots,n,\, j=1,\ldots,\ell}\right),$$

where the variables $a_j^{(i)}$ obviously play the role of the previously defined functions. From Theorem 4 and Corollary 5, it follows that by evaluating $\texttt{hwce}_{n,m}$ at $\bar{v}$ we obtain the vector $(a_1^{(n)}(\bar{v}), \ldots, a_\ell^{(n)}(\bar{v}))$ containing the binary expansion of $\mathrm{w}(\bar{v})$.

3.3 Weight constraint encoding

The idea for the construction of this encoding is the definition of a quadratic Boolean polynomial capable of comparing two Boolean vectors according to the values of these two vectors when seen as integers. More precisely, in this section we construct a polynomial whose evaluation at a pair of Boolean vectors $\bar{u}$ and $\bar{v}$ is 0 if and only if $\texttt{int}(\bar{u}) \leq \texttt{int}(\bar{v})$. For the sake of completeness, we define the claimed polynomial for a more general situation, and in the end we specify the parameters useful in the context of our reduction from MLD to MQ.

Consider two binary vectors $\bar{u}, \bar{v} \in \mathbb{F}^l$, where the most significant bits are $\bar{u}_l$ and $\bar{v}_l$, respectively. To compare the integers associated to $\bar{u}$ and $\bar{v}$ we can follow the following procedure, which outputs 0 when $\texttt{int}(\bar{u}) \leq \texttt{int}(\bar{v})$ and 1 otherwise, starting from $j = l$:

- if $\bar{u}_j = \bar{v}_j$, then we move to the next bits $\bar{u}_{j-1}$ and $\bar{v}_{j-1}$;
- if $\bar{u}_j \neq \bar{v}_j$, we output $\bar{u}_j$, since $\texttt{int}(\bar{u}) \leq \texttt{int}(\bar{v})$ if $\bar{u}_j = 0$;
- if we reach $j = 1$ and $\bar{u}_1 = \bar{v}_1$, then we output 0.

We make use of this procedure to define a polynomial $F \in \mathbb{F}[u_1, \ldots, u_l, v_1, \ldots, v_l]$ such that

$$F(\bar{u}, \bar{v}) = \begin{cases} 0 & \text{if } \texttt{int}(\bar{u}) \leq \texttt{int}(\bar{v}) \\ 1 & \text{if } \texttt{int}(\bar{u}) > \texttt{int}(\bar{v}) \,. \end{cases} \tag{5}$$

Define $g_h(u, v) = (u_h + v_h) \in \mathbb{F}[u_h, v_h]$ for every $h = 1, \ldots, l$ and notice that $g_h(\bar{u}, \bar{v}) = 0$ if and only if $\bar{u}_h = \bar{v}_h$. Moreover, for $j = 1, \ldots, l$, define the polynomials

$$f_j = g_j \prod_{h=j+1}^{l} (g_h + 1) \,, \tag{6}$$

where $f_j \in \mathbb{F}[u_1, \ldots, u_l, v_1, \ldots, v_l]$. Clearly in the special case $j = l$ we have $f_l = g_l$. Observe that the degree of f_j is $l - j + 1$.

Given two vectors $\bar{u}, \bar{v} \in \mathbb{F}^l$, the purpose of the set of polynomials $\{f_j\}_{j=1}^l$ in (6) is to locate the most significant bit in which $\bar{u}$ and $\bar{v}$ differ. We prove this in the following lemmas.

Lemma 7 *Let $\bar{u}, \bar{v} \in \mathbb{F}^l$, then $f_j(\bar{u}, \bar{v}) = 1$ for at most one value of j.*

Proof Assume by contradiction that $f_j(\bar{u}, \bar{v}) = f_i(\bar{u}, \bar{v}) = 1$ for $i \neq j$. We can assume, without loss of generality, that $i < j$. By construction of f_j we must have that $g_j(\bar{u}, \bar{v}) = 1$. But $(g_j + 1) \mid f_i$ since $i < j$, $(g_j + 1)(\bar{u}, \bar{v}) = 0$, and therefore $f_i(\bar{u}, \bar{v}) = 0$, which is a contradiction. □

Lemma 8 *Let $\bar{u}, \bar{v} \in \mathbb{F}^l$, then $f_j(\bar{u}, \bar{v}) = 0 \quad \forall \quad j = 1, \ldots, l$ if and only if $\bar{u} = \bar{v}$.*

Proof We show the first implication. Since $f_l(\bar{u}, \bar{v}) = g_l(\bar{u}, \bar{v}) = 0$, then $\bar{u}_l = \bar{v}_l$.

We claim that, for any $1 \leq k \leq l - 1$, $\bar{u}_k$ is equal to $\bar{v}_k$, provided that $\bar{u}_h = \bar{v}_h$ for any $k + 1 \leq h \leq l$. Notice that $(g_h + 1) \mid f_k$ for every value $h = k + 1, \ldots, l$. Since $\bar{u}_h = \bar{v}_h$ then $(g_h + 1)(\bar{u}, \bar{v}) = 1$, for $h = k + 1, \ldots, l$. This implies $0 = f_k(\bar{u}, \bar{v}) = g_k(\bar{u}, \bar{v}) \cdot 1 = \bar{u}_k + \bar{v}_k$ and therefore $\bar{u}_k = \bar{v}_k$.

The first implication follows by an iterated application of our claim from $k = l - 1$ until $k = 1$.

As regards the second implication, $\bar{u} = \bar{v}$ implies $\bar{u}_j = \bar{v}_j$ for every $j = 1, \ldots, l$, and so $g_j(\bar{u}, \bar{v}) = 0$. Observe also that, by construction in (6), $g_j \mid f_j$ for every $j = 1, \ldots, l$, which forces $f_j(\bar{u}, \bar{v}) = 0$. □

Lemma 9 *Let $\bar{u}, \bar{v} \in \mathbb{F}^l$ with $\bar{u} \neq \bar{v}$, then there exists a unique j such that $f_j(\bar{u}, \bar{v}) = 1$. Moreover $\bar{u}_j \neq \bar{v}_j$ while $\bar{u}_k = \bar{v}_k$ for every $k = j + 1, \ldots, l$.*

Proof If $\bar{u} \neq \bar{v}$ then there exists j such that $\bar{u}_j \neq \bar{v}_j$. Let j be such that $\bar{u}_j \neq \bar{v}_j$ and $\bar{u}_k = \bar{v}_k$ for every $k > j$. This implies $(g_k + 1)(\bar{u}, \bar{v}) = 1$ for each $k > j$, as well as $g_j(\bar{u}, \bar{v}) = 1$, thus $f_j(\bar{u}, \bar{v}) = 1$. The uniqueness of j follows from Lemma 7.

For the second part of the statement, if there exists $k > j$ such that $\bar{u}_k \neq \bar{v}_k$, then $g_k(\bar{u}_k, \bar{v}_k) = 1$. However, $(g_k + 1) \mid f_j$ and $(g_k + 1)(\bar{u}, \bar{v}) = 0$ imply $f_j(\bar{u}, \bar{v}) = 0$, which is a contradiction. □

We are ready to define our function F as in Eq. (5).

Proposition 10 *Let* $F = \sum_{j=1}^{l} f_j \cdot (v_j + 1) \in \mathbb{F}[u_1, \ldots, u_l, v_1, \ldots, v_l]$ *and let* $\bar{u}, \bar{v} \in \mathbb{F}^l$. *Then* $F(\bar{u}, \bar{v}) = 0$ *if and only if* $\mathtt{int}(\bar{u}) \leq \mathtt{int}(\bar{v})$.

Proof We have two cases: either $f_j(\bar{u}, \bar{v}) = 0$ for every value $j = 1, \ldots, l$, or there exists a (by Lemma 9) *unique* k such that $f_k(\bar{u}, \bar{v}) = 1$.

The first case, due to Lemma 8, is equivalent to $\bar{u} = \bar{v}$ and thus it corresponds to the equality $\mathtt{int}(\bar{u}) = \mathtt{int}(\bar{v})$. By definition of F, the first case also implies $F(\bar{u}, \bar{v}) = 0$.

We have proved that $\mathtt{int}(\bar{u}) = \mathtt{int}(\bar{v})$ implies $F(\bar{u}, \bar{v}) = 0$, while $F(\bar{u}, \bar{v}) = 0$ implies either that $\mathtt{int}(\bar{u}) = \mathtt{int}(\bar{v})$ or that we are not in the first case.

In the second case, let $f_k(\bar{u}, \bar{v}) = 1$ for a certain value k. Lemma 8 implies that $\mathtt{int}(\bar{u}) \neq \mathtt{int}(\bar{v})$, and by Lemma 9 we have $\bar{u}_j = \bar{v}_j$ for $k + 1 \leq j \leq l$ and $\bar{u}_k \neq \bar{v}_k$.

If $\bar{u}_k = 0$ and $\bar{v}_k = 1$, then $\mathtt{int}(\bar{u}) < \mathtt{int}(\bar{v})$ and

$$F(\bar{u}, \bar{v}) = \sum_{j \neq k} f_j(\bar{u}, \bar{v}) \cdot (\bar{v}_j + 1) + f_k(\bar{u}, \bar{v}) \cdot (\bar{v}_k + 1) = \sum_{j \neq k} 0 \cdot (\bar{v}_j + 1) + 1 \cdot (\bar{v}_k + 1) = \bar{v}_k + 1 = 0 \,.$$

If instead $\bar{u}_k = 1$ and $\bar{v}_k = 0$, then $\mathtt{int}(\bar{u}) > \mathtt{int}(\bar{v})$ and

$$F(\bar{u}, \bar{v}) = \sum_{j \neq k} f_j(\bar{u}, \bar{v}) \cdot (\bar{v}_j + 1) + f_k(\bar{u}, \bar{v}) \cdot (\bar{v}_k + 1) = \sum_{j \neq k} 0 \cdot (\bar{v}_j + 1) + 1 \cdot (\bar{v}_k + 1) = \bar{v}_k + 1 = 1 \,.$$

Either way, if $\mathtt{int}(\bar{u}) \neq \mathtt{int}(\bar{v})$ then $F(\bar{u}, \bar{v}) = 0$ if and only if $\mathtt{int}(\bar{u}) < \mathtt{int}(\bar{v})$. □

Observe that the degree of F is equal to $\deg(f_1) + 1 = l + 1$. Observe also that the degree of f evaluated at $\bar{v}_1, \ldots, \bar{v}_l$, which is a polynomial in $\mathbb{F}[u_1, \ldots, u_l]$, is at most l.

Let $I_{\mathrm{MLD}} = (\bar{H}, \bar{s}, \bar{t})$, let $\bar{v}$ be a vector for which $\bar{H}\bar{v}^\top = \bar{s}^\top$ and let $\bar{t} = (\bar{t}_1, \ldots, \bar{t}_\ell)$. We recall that in Sect. 3.2 we defined a set of ℓ Boolean functions $a_1^{(n)}, \ldots, a_\ell^{(n)}$ representing the weight of a length-n vector v, i.e. by Corollary 5 $\mathtt{int}(a_1^{(n)}(\bar{v}), \ldots, a_\ell^{(n)}(\bar{v})) = \mathrm{w}(\bar{v})$.

Then, by Corollary 5 and Proposition 10,

$$F\left(a_1^{(n)}(\bar{v}), \ldots, a_\ell^{(n)}(\bar{v}), \bar{t}_1, \ldots, \bar{t}_\ell\right) = 0$$

if and only if

$$\mathrm{w}(\bar{v}) \leq \mathtt{int}(\bar{t}) \,.$$

Let us consider the following polynomial

$$F_\ell \in \mathbb{F}\left[a_1^{(n)}, \ldots, a_\ell^{(n)}, t_1, \ldots, t_\ell\right] ,$$

which is obtained by rewriting F in the variables $a_1^{(n)}, \ldots, a_\ell^{(n)}$ and in (the new variables of) $t_1, \ldots, t_\ell$. Clearly, $F_\ell(a_1^{(n)}, \ldots, a_\ell^{(n)}, t_1, \ldots, t_\ell) = 0$ encodes the constraint $\mathrm{w}(\bar{v}) \leq \mathtt{int}(\bar{t})$. Obviously, the degree of $F_\ell(a_1^{(n)}, \ldots, a_\ell^{(n)}, \bar{t}_1, \ldots, \bar{t}_\ell) \in \mathbb{F}[a_1^{(n)}, \ldots, a_\ell^{(n)}]$ is at most ℓ. By applying the map $\mathtt{quadr}()$ to F_ℓ (see Remark 1) we obtain the system of quadratic equations

$$\texttt{wce}_{n,m} = \texttt{quadr}\big(\{F_\ell\}\big)\ .$$

Lemma 11 $\texttt{wce}_{n,m}(I_{\mathrm{MLD}})$ *is a system of* $\mathcal{O}(n\ell)$ *quadratic equations in* $\mathcal{O}(n\ell)$ *variables.*

Proof Since the degree of F_ℓ is at most ℓ and the size of its support, i.e. the number of monomials, is at most 2^ℓ, by applying quadr() we obtain a system with at most $2^\ell \ell$ equations in $2^\ell \ell$ variables. However, $2^\ell \leq 2n$, and so the system contains $\mathcal{O}(n\ell)$ equations in $\mathcal{O}(n\ell)$ variables. □

3.4 MLD to MQ

By combining the results of Sects. 3.1, 3.2 and 3.3, we construct the reduction $\alpha_{n,m}$, a function mapping MLD instances to MQ instances.

Theorem 12 *Let* $I_{\mathrm{MLD}} = (\bar{H}, \bar{s}, \bar{t})$ *and let* $\alpha_{n,m} : \mathrm{MLD} \to \mathrm{MQ}$ *be defined by*

$$\alpha_{n,m}(\bar{H}, \bar{s}, \bar{t}) = \texttt{pcce}_{n,m}(\bar{H}, \bar{s}, \bar{t}) \cup \texttt{hwce}_{n,m}(\bar{H}, \bar{s}, \bar{t}) \cup \texttt{wce}_{n,m}(\bar{H}, \bar{s}, \bar{t})\,;$$

If $\bar{u}$ *is a witness of* $\alpha_{n,m}(I_{\mathrm{MLD}})$ *then it is also a witness of* I_{MLD}.

Proof By ordering the variables according to their first appearance in this paper, the first n variables in $\alpha_{n,m}(I_{\mathrm{MLD}})$ are $v_1, \ldots, v_n$ and the first n bits of a witness u for $\alpha_{n,m}(I_{\mathrm{MLD}})$ are the values $\bar{v}_1, \ldots, \bar{v}_n$. Let us call $\bar{v} = (\bar{v}_1, \ldots, \bar{v}_n)$.

The set $\texttt{pcce}_{n,m}(\bar{H}, \bar{s}, \bar{t})$ contains only equations in the variables $v_1, \ldots, v_n$. Therefore, since $\bar{u}$ is a witness for $\alpha_{n,m}(I_{\mathrm{MLD}})$, we have $f(\bar{u}) = 0$ for $f \in \texttt{pcce}_{n,m}(\bar{H}, \bar{s}, \bar{t})$, meaning that $\bar{H}\bar{v}^\top = \bar{s}$.

From Corollary 5, we can use the polynomials in $\texttt{hwce}_{n,m}(\bar{H}, \bar{s}, \bar{t})$ to compute a binary expansion of the weight of $\mathrm{w}(\bar{v})$. More precisely, we have that

$$\texttt{int}\Big(a_1^{(n)}(\bar{v}), \ldots, a_l^{(n)}(\bar{v})\Big) = \mathrm{w}(\bar{v})\ .$$

Finally, since $F_\ell\Big(a_1^{(n)}(\bar{v}), \ldots, a_l^{(n)}(\bar{v}), \bar{t}\Big) = 0$, by Proposition 10 we have $\mathrm{w}(\bar{v}) \leq \texttt{int}(\bar{t})$.

□

Theorem 13 *Let* (n, m) *be the complexity parameters of* I_{MLD} *and let* $I_{\mathrm{MQ}} = \alpha_{n,m}(I_{\mathrm{MLD}})$. *Then, the complexity parameters* $(\mathtt{n}, \mathtt{m})$ *of* I_{MQ} *are in* $\mathcal{O}(n \log_2^2 n)$.

Proof We analyse the 3 sub-problems separately.

- According to Lemma 2, $\texttt{pcce}_{n,m}(I_{\mathrm{MLD}})$ is a linear system with m equations and n variables.
- As described by Lemma 6, $\texttt{hwce}_{n,m}(I_{\mathrm{MLD}})$ is a quadratic system of $\mathcal{O}(n\ell^2)$ equations in $\mathcal{O}(n\ell^2)$ variables.
- In the weight-constraint step described in Sect. 3.3, we introduce a quadratic system $\texttt{wce}_{n,m}(I_{\mathrm{MLD}})$ containing $\mathcal{O}(n\ell)$ equations and variables, as stated in Lemma 11.

By putting everything together we can compute the complexity parameters (n, m) of $\alpha_{n,m}(I_{\mathrm{MLD}})$. □

4 MQ to MLD reduction

In this section we construct a reduction $\beta : \mathrm{MQ} \to \mathrm{MLD}$. We consider a general system of equations and we take it to standard form S, as defined in Definition 3, by using the procedure hinted in the Proof of Lemma 1. The result of the process will be an instance $\beta(I_{\mathrm{MQ}}) = (H, s, t) \in \mathrm{MLD}$, with $H \in \mathbb{F}^{m \times n}$ for some $m, n, t \in \mathbb{Z}^+$, $s \in \mathbb{F}^m$. The key idea is to regard the two different types of equations in S separately. First we create one part of the MLD instance according to the quadratic equations in S, and then we complete the work by integrating the linear ones. We also build a transformation that takes as input a solution of $\beta(I_{\mathrm{MQ}})$ and outputs a solution of I_{MQ}, proving that β is actually a reduction between the two problems.

4.1 Quadratic equations

Consider a system S containing solely the equation $xy + z = 0$ and let $I = \langle xy + z \rangle \subset \mathbb{F}[x, y, z]$ be the principal ideal generated by S. The associated variety $\mathcal{V}_{\mathbb{F}}(I) \subset \mathbb{F}^3$ is

$$\mathcal{V}_{\mathbb{F}}(I) = \{(0,0,0), (1,0,0), (0,1,0), (1,1,1)\}. \tag{7}$$

Lemma 14 *Let $\widehat{C}$ be the linear code generated by the generator matrix*

$$\widehat{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{8}$$

let

$$\widehat{\epsilon} = (0,0,0,0,0,0,0,1,1,1)\,,$$

let $\widehat{\Sigma}$ be the coset $\widehat{\epsilon} + \widehat{C} \subseteq \mathbb{F}^{10}$ and let $v \in \widehat{\Sigma}$.

Then, v has weight at most 3 if and only if $\tau_3(v) \in \mathcal{V}_{\mathbb{F}}(I)$ as in (7).

Proof It follows by direct inspection of the 8 vectors in $\widehat{\Sigma}$. □

The truncation map present in the statement of the previous Lemma, $\tau_3 : \mathbb{F}^{10} \to \mathbb{F}^3$ defined as $\tau_3(\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4, \bar{v}_5, \bar{v}_6, \bar{v}_7, \bar{v}_8, \bar{v}_9, \bar{v}_{10}) = (\bar{v}_1, \bar{v}_2, \bar{v}_3)$, can be represented in matrix form as

$$M_{\tau_3} = \left[\frac{\mathbb{I}_3}{\mathbf{0}}\right],$$

where $\mathbb{I}_3$ is the identity matrix of dimension 3 and $\mathbf{0}$ is the 3×7 zero matrix, namely

$$\tau_3(\bar{v}) = \bar{v}M_{\tau_3} \ .$$

Proposition 15 *Let $\widehat{H}$ be a parity-check matrix for the code $\widehat{C}$ generated by $\widehat{G}$ as defined in Lemma 14, let $\widehat{s} = \widehat{H} \cdot \widehat{\epsilon}^\top$ and let $\widehat{t} = 3$. Let $\widehat{W} \subset \widehat{\Sigma}$ be the set of witnesses of the MLD instance $I_{\text{MLD}} = (\widehat{H}, \widehat{s}, \widehat{t})$ and let $\mathcal{V}_{\mathbb{F}}(I)$ be as in (7). Then $\mathcal{V}_{\mathbb{F}}(I) = \tau_3(\widehat{W})$.*

Proof It follows by Lemma 14 and the well-known bijection between cosets and syndromes (once the parity-check matrix has been chosen). □

Remark 1 The witnesses of the MLD instance $(\widehat{H}, \widehat{s}, \widehat{t})$ we constructed, i.e. solutions of $\widehat{H}v^\top = \widehat{s}^\top$ with weight at most $\widehat{t} = 3$, have Hamming weight exactly 3, which is the weight of any coset leader (e.g. $\widehat{\epsilon}$). Notice that the remaining solutions of $\widehat{H}v^\top = \widehat{s}^\top$ have weight at least 5. This gap in the weight is crucial for the generalisation we are going to give next.

We now extend the construction in Proposition 15 to a standard-form system S that contains more than one quadratic equation. Assume that S contains q quadratic equations f_i's of the form $x_i y_i + z_i = 0$, for $i = 1, \ldots, q$. Recall that by Definition 3, such equations do not share any variable with each other. Consider the ideal $J = \langle f_1, \ldots f_q \rangle \subset \mathbb{F}[\{x_i, y_i, z_i\}_{i=1,\ldots,q}]$. The variety $\mathcal{V}_{\mathbb{F}}(J)$ can be seen as

$$\mathcal{V}_{\mathbb{F}}(J) = \underbrace{\mathcal{V}_{\mathbb{F}}(I) \times \mathcal{V}_{\mathbb{F}}(I) \times \cdots \times \mathcal{V}_{\mathbb{F}}(I)}_{\text{q}} \subset \mathbb{F}^{3q} \ , \tag{9}$$

where $\mathcal{V}_{\mathbb{F}}(I)$ is as in (7). To address the case of standard-form systems consisting of only quadratic equations, we construct a new parity-check matrix as the diagonal block matrix $\widetilde{H}$ of size $7q \times 10q$

$$\widetilde{H} = \begin{bmatrix} \widehat{H} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \widehat{H} \end{bmatrix} ,$$

where $\widehat{H}$ is a parity-check matrix for the code generated by $\widehat{G}$ in Eq. (8). Obviously, the null space of $\widetilde{H}$ is the direct product of q copies of $\widehat{C}$, i.e.

$$\underbrace{\widehat{C} \times \widehat{C} \times \cdots \times \widehat{C}}_{\text{q}} \subset \mathbb{F}^{10q} \ .$$

Define $\widetilde{\Sigma} = \overbrace{\widehat{\Sigma} \times \widehat{\Sigma} \times \cdots \times \widehat{\Sigma}}^{\text{q}}$, $\widetilde{t} = 3q$ and $\widetilde{\epsilon} = (\overbrace{\widehat{\epsilon} \| \widehat{\epsilon} \| \cdots \| \widehat{\epsilon}}^{\text{q}})$ where $\|$ denotes vector concatenation. With this setting, for any $\widetilde{v} \in \widetilde{\Sigma}$, we can write $\widetilde{v} = \left(\widehat{v}^{(1)} \| \widehat{v}^{(2)} \| \cdots \| \widehat{v}^{(q)}\right)$ with $\widehat{v}^{(i)} \in \widehat{\Sigma}$ for any i. In the following lemma $\widehat{W}$ is the set of vectors in $\widehat{\Sigma}$ with weight at most $\widehat{t}$, as defined in Proposition 15.

Lemma 16 *Let $\widetilde{W} = \{\widehat{v} \in \widetilde{\Sigma} \mid \mathrm{w}(\widehat{v}) \leq \widetilde{t}\}$. Then $\widetilde{W} = \underbrace{\widehat{W} \times \widehat{W} \times \cdots \times \widehat{W}}_{\text{q}}$.*

Proof Notice that, by Remark 1, $\mathrm{w}(\widehat{v}) = \widehat{t} = 3$ for every $\widehat{v} \in \widehat{W}$. So $\mathrm{w}(\widetilde{v}) = \widetilde{t} = 3q$ for every $\widetilde{v} \in \widehat{W} \times \widehat{W} \times \cdots \times \widehat{W}$. Therefore $\widetilde{W} \supseteq \widehat{W} \times \widehat{W} \times \cdots \times \widehat{W}$.

To prove the other inclusion consider $\widetilde{v} \in \widetilde{W}$.

Then $\widetilde{v}$ can be written as a concatenation of q vectors in $\widehat{\Sigma}$ and, by Remark 1, each of such vectors has Hamming weight at least 3, and $\mathrm{w}(\widetilde{v}) = \sum_{i=1}^{q} \mathrm{w}(\widehat{v}^{(i)})$. If there is a $\widehat{v}^{(i)}$ with weight more than 3, then the weight of $\widetilde{v}$ is strictly larger that $3q$. Therefore, all $\widehat{v}^{(i)}$ must have weight exactly 3. This proves $\widetilde{W} \subseteq \widehat{W} \times \widehat{W} \times \cdots \times \widehat{W}$. □

Consider the truncation map $\tau : \mathbb{F}^{10q} \to \mathbb{F}^{3q}$ whose matrix representation is

$$M_\tau = \begin{bmatrix} M_{\tau_3} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_{\tau_3} \end{bmatrix}. \tag{10}$$

The following lemma proves a useful property of τ that comes at hand to prove the subsequent theorem.

Lemma 17 *Let $\widetilde{v} \in \widetilde{\Sigma}$ then $\tau(\widetilde{v}) = \left(\tau_3(\widehat{v}^{(1)}), \ldots, \tau_3(\widehat{v}^{(q)})\right)$ where $\widehat{v}^{(i)} \in \widehat{\Sigma}$ for every $i = 1, \ldots, q$*
.

Proof Considering the matrix representation of τ, we obtain

$$\begin{aligned} \tau(\widetilde{v}) &= \widetilde{v} M_\tau \\ &= \left(\widehat{v}^{(1)} \| \cdots \| \widehat{v}^{(q)}\right) \begin{bmatrix} M_{\tau_3} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_{\tau_3} \end{bmatrix} \\ &= \left(\widehat{v}^{(1)} M_{\tau_3}, \ldots, \widehat{v}^{(q)} M_{\tau_3}\right) \\ &= \left(\tau_3(\widehat{v}^{(1)}), \ldots, \tau_3(\widehat{v}^{(q)})\right). \end{aligned}$$

□

Proposition 18 *Set $\widetilde{s} = \widetilde{H}\widetilde{e}^\top$, then $\widetilde{W}$ solves the MLD instance $\left(\widetilde{H}, \widetilde{s}, \widetilde{t}\right)$. Moreover $\mathcal{V}_{\mathbb{F}}(J) = \tau(\widetilde{W})$.*

Proof We need to prove that given $\widetilde{v} \in \widetilde{W}$ it holds $\widetilde{H}\widetilde{v}^\top = \widetilde{s}$ and $\mathrm{w}(\widetilde{v}) \leq \widetilde{t}$. The null space of $\widetilde{H}$ is $\widetilde{H}^\perp = \widehat{C} \times \widehat{C} \times \cdots \times \widehat{C}$. Observe that

$$\begin{aligned} \widetilde{\Sigma} &= \widehat{\Sigma} \times \widehat{\Sigma} \times \cdots \times \widehat{\Sigma} \\ &= \left(\widehat{C} + \widehat{e}\right) \times \left(\widehat{C} + \widehat{e}\right) \times \cdots \times \left(\widehat{C} + \widehat{e}\right) \\ &= \left(\widehat{C} \times \widehat{C} \times \cdots \times \widehat{C}\right) + \widetilde{e} \\ &= \widetilde{H}^\perp + \widetilde{e} = \{v + \widetilde{e} \mid v \in \widehat{C} \times \widehat{C} \times \cdots \times \widehat{C}\} \end{aligned}$$

Therefore, for every $\widetilde{v} \in \widetilde{\Sigma}$, we have $\widetilde{H}^{\perp}\widetilde{v}^{\top} = \widetilde{H}^{\perp}\widetilde{v}^{\top} + \widetilde{H}^{\perp}\widetilde{e}^{\top} = \widetilde{s}$. Considering $\widetilde{v} \in \widetilde{W} \subset \widetilde{\Sigma}$ we also obtain $\mathrm{w}(\widetilde{v}) \leq \widetilde{t}$.

For the second claim we have, by Lemmas 16 and 17, that

$$\begin{aligned}\tau\left(\widetilde{W}\right) &= \tau\left(\widehat{W} \times \widehat{W} \times \cdots \times \widehat{W}\right) \\ &= \tau_3(\widehat{W}) \times \ldots \times \tau_3(\widehat{W}) \\ &= \mathcal{V}_{\mathbb{F}}(I) \times \mathcal{V}_{\mathbb{F}}(I) \times \cdots \times \mathcal{V}_{\mathbb{F}}(I) \\ &= \mathcal{V}_{\mathbb{F}}(J)\ ,\end{aligned}$$

where the last two equalities hold due to Proposition 15 and Eq. (9). □

4.2 Linear equations

Let S be a standard-form system containing q quadratic equations. Due to Definition 3, S is a system in exactly $3q$ variables. We can thus write $S \subset \mathbb{F}[x_1, \ldots, x_{3q}]$.

Remark 2 Consider a linear polynomial f in $\mathbb{F}[x_1, \ldots, x_{3q}]$ for some value of $q \in \mathbb{Z}^+$. We can write $f = \sum_{i=1}^{3q} a_i x_i + \delta$ and define the vector of its coefficients $a_f = (a_1, \ldots, a_{3q}) \in \mathbb{F}^{3q}$. Notice that the vector a_f contains only the coefficients of $x_1, \ldots, x_{3q}$ and not the term δ. With this notation we observe that $\bar{w} \in \mathbb{F}^{3q}$ belongs to $\mathcal{V}_{\mathbb{F}}(\langle f \rangle)$ if and only if the product $\bar{w} \cdot a_f^{\top} = \delta$.

The reduction introduced in Sect. 4.1 deals with standard-form systems that include only quadratic equations. This reduction is formalised in Proposition 18 as a map taking as input a system of q equations in $3q$ variables and outputting an MLD instance corresponding to a $3q \times 10q$ parity-check matrix. To deal with linear equations we need a map ν sending a linear polynomial in $\mathbb{F}[x_1, \ldots, x_{3q}]$ to a vector in $\mathbb{F}^{10q}$. We define ν as

$$\nu(f) = a_f M_{\tau}^{\top}\ ,$$

with M_{τ} as in (10).

Example 2 Assume $q = 2$, then we are working in $\mathbb{F}[x_1, \ldots, x_6]$. Let $f = x_1 + x_3 + x_5$ and $a_f = (1, 0, 1, 0, 1, 0)$. Since $q = 2$ then $M_{\tau}^{\top}$ is the matrix

$$M_{\tau}^{\top} = \left[\begin{array}{cc|cc} \mathbb{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{I}_3 & \mathbf{0} \end{array}\right]\ .$$

We obtain

$$\nu(f) = a_f M_{\tau}^{\top} = (1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$$

The following lemma will be used to prove the correctness of our reduction.

Lemma 19 *Let $f = \sum_{i=1}^{3q} a_i x_i + \delta \in \mathbb{F}[x_1, \ldots, x_{3q}]$ be a linear polynomial. Let $\widetilde{v} \in \mathbb{F}^{10q}$. $\widetilde{v} \cdot \nu(f)^{\top} = \delta$ if and only if $\tau(\widetilde{v}) \in \mathcal{V}_{\mathbb{F}}(\langle f \rangle)$.*

Proof By Remark 2, observing that

$$\widetilde{v} \cdot \nu(f)^{\top} = \widetilde{v} \cdot \left(a_f M_\tau^{\top}\right)^{\top} = \widetilde{v} M_\tau a_f^{\top} = \left(\widetilde{v} M_\tau\right) \cdot a_f^{\top} = \tau(\widetilde{v}) \cdot a_f^{\top} = f(\tau(\widetilde{v})) \, .$$

□

We construct now the MLD instance (H, s, t) for a general MQ system. The basic idea is to see the parity-check matrix $\widetilde{H}$ we built so far as a matrix of coefficients for an equation system. Adding rows to the matrix means adding new equations to the system and thus reducing the solution space. Consider a standard-form system $S \in \mathbb{F}[x_1, \ldots, x_{3q}]$ containing q quadratic equations and λ linear equations, as in Definition 3. Let $K = \langle S \rangle$ and $\mathcal{V}_{\mathbb{F}}(K) \subset \mathbb{F}^{10q}$ be its variety. Denote by $S_Q \subset S$ the subset of quadratic equations in S, and by $\langle S_Q \rangle$ the ideal generated by it.

Let also $\left(\widetilde{H}, \widetilde{s}, \widetilde{t}\right)$ be the MLD instance corresponding to S_Q. Let $\{f_1, \ldots, f_\lambda\} \subset S$ be the set of linear equations of S. We build a parity-check matrix H as follows

$$H = \begin{bmatrix} \widetilde{H} \\ \nu(f_1) \\ \vdots \\ \nu(f_\lambda) \end{bmatrix} \in \mathbb{F}^{(7q+\lambda) \times 10q} \tag{11}$$

Consider the following syndrome vector

$$s = \widetilde{s} \| \delta_1 \| \cdots \| \delta_\lambda \, ,$$

where $f_i(0) = \delta_i$ for any i, and set $t = \widetilde{t} = 3q$.

Consider furthermore the set $W = \{\widetilde{v} \in \widetilde{W} \mid \widetilde{v} \cdot \nu(f_i)^{\top} = \delta_i \ \forall i = 1, \ldots, \lambda\} \subset \widetilde{W}$.

Theorem 20 *W is the set of witnesses for the MLD instance (H, s, t). Moreover $\mathcal{V}_{\mathbb{F}}(K) = \tau(W)$.*

Proof We need to prove that given $\widetilde{v} \in W$ it holds $H\widetilde{v}^{\top} = s^{\top}$ and $\mathrm{w}(\widetilde{v}) \leq t$. By definition of W, we obtain that $\widetilde{v} \in W$ implies $\widetilde{v} \in \widetilde{W}$, which means $\mathrm{w}(\widetilde{v}) \leq \widetilde{t}$ and also $\widetilde{H}\widetilde{v}^{\top} = \widetilde{s}^{\top}$. Moreover we have $\widetilde{v} \cdot \nu(f_i)^{\top} = \delta_i$ for every $i = 1, \ldots, \lambda$ implying

$$\begin{bmatrix} \widetilde{H} \\ \nu(f_1) \\ \vdots \\ \nu(f_\lambda) \end{bmatrix} \widetilde{v}^{\top} = \begin{pmatrix} \widetilde{s} \\ \delta_1 \\ \vdots \\ \delta_\lambda \end{pmatrix} = s^{\top} \, .$$

Due to Lemma 19 we have $\tau(\widetilde{v}) \in \mathcal{V}_{\mathbb{F}}(\langle f_i \rangle)$ for $i = 1, \ldots, \lambda$. Therefore $\tau(\widetilde{v}) \in \bigcap_{i=1}^{\lambda} \mathcal{V}_{\mathbb{F}}(\langle f_i \rangle) = \mathcal{V}_{\mathbb{F}}(\langle f_1, \ldots, f_\lambda \rangle)$. Since by Proposition 18 we have $\tau(\widetilde{v}) \in \mathcal{V}_{\mathbb{F}}(S_q)$, then $\tau(\widetilde{v}) \in \mathcal{V}_{\mathbb{F}}(\langle S_q \rangle) \cap \mathcal{V}_{\mathbb{F}}(\langle f_1, \ldots, f_\lambda \rangle) = \mathcal{V}_{\mathbb{F}}(\langle S \rangle) = \mathcal{V}_{\mathbb{F}}(K)$.

On the other hand let $z \in \mathcal{V}_{\mathbb{F}}(\langle K \rangle) = \mathcal{V}_{\mathbb{F}}(\langle S_q \rangle) \cap \mathcal{V}_{\mathbb{F}}(\langle f_1, \ldots, f_\lambda \rangle) = \tau(\widetilde{W}) \cap \mathcal{V}_{\mathbb{F}}(\langle f_1, \ldots, f_\lambda \rangle) \subseteq \tau(\widetilde{W})$, where the last equality comes from Proposition 18. Therefore there exists

$z' \in \tau(\widetilde{W})$ such that $z = \tau(z')$. Since $a_{f_i} \cdot z^\top = a_{f_i} \cdot \tau(z')^\top = \delta_i$ for every $i = 1, \ldots, \lambda$, this implies $\nu(a_{f_i}) \cdot z'^\top = \delta_i$ and therefore $z \in \tau(W)$. □

We can now define the map $\beta : \mathrm{MQ} \to \mathrm{MLD}$ as follows

$$\beta(S) = (H, s, t)$$

where H, s and t are as in Theorem 20. We now prove that a witness of such instance can be transformed into a witness of S by applying the truncation τ.

Theorem 21 *Let $I_{\mathrm{MQ}} = S$ where S is a standard form system of quadratic equations. If $\widetilde{v} \in \mathbb{F}^{10q}$ is a solution of $\beta(I_{\mathrm{MQ}})$ then $\tau(\widetilde{v})$ is a solution of I_{MQ}.*

Proof Let $K = \langle S \rangle$ and apply Theorem 20. If $\widetilde{v}$ solves $\beta(I_{\mathrm{MQ}}) = (H, s, t)$ then $\tau(\widetilde{v}) \in \mathcal{V}_{\mathbb{F}}(K)$, i.e. $\tau(\widetilde{v})$ is a solution of S. □

Theorem 22 *Given a MQ system $S \in \mathbb{F}[x_1, \ldots, x_{\mathsf{n}}]$ consisting of m equations, the reduction β runs in polynomial space bounded by $\mathcal{O}(\mathsf{n}^4\mathsf{m}^2)$.*

Proof Recall that by Lemma 1 we can transform S into a standard form system S' in $\mathcal{O}(\mathsf{n}^2\mathsf{m})$ operations. This process produces $S' = S'_Q \cup S'_L$ where S'_Q and S'_L are sets of quadratic and linear equations, namely. Let $q = |S'_Q| \leq \mathsf{m}\left(\frac{\mathsf{n}(\mathsf{n}-1)}{2}\right) \leq \mathsf{m}\mathsf{n}^2$ and $\lambda = |S'_L| \leq \mathsf{m}\left(\frac{3\mathsf{n}^2-\mathsf{n}}{2} - 2\right) \leq \frac{3}{2}\mathsf{m}\mathsf{n}^2$. We estimate only the construction of the matrix H since s and t have a significant smaller size.

The matrix H generated via β has dimension $(7q + \lambda) \times 10q$ as in (11), therefore it takes space at most

$$\left(7(\mathsf{m}\mathsf{n}^2) + \frac{3}{2}\mathsf{m}\mathsf{n}^2\right) \cdot 10\mathsf{m}\mathsf{n}^2 = 85\mathsf{m}^2\mathsf{n}^4 \in \mathcal{O}(\mathsf{m}^2\mathsf{n}^4).$$

□

5 Conclusions

In this work we introduced two polynomial-time reductions: α, from MLD to MQ, and β, from MQ to MLD. Therefore, the composition of α and β is a polynomial-time auto-reduction in MLD, while $\alpha \circ \beta$ is a polynomial-time auto-reduction in MQ. Hence each MLD instance can be solved if we are able to solve each MLD instance defined by $\beta \circ \alpha$ and even more so if we are able to solve those defined by β. Similarly, each MQ instance can be solved if we are able to solve each MQ instance defined by $\alpha \circ \beta$, or even only defined by α. So if we can decide in polynomial time the existence of solutions for all systems in the image of α, then we can solve MQ in polynomial time. Notice that the same property holds

for systems in standard form. So we can identify two families of systems which play a special role in the MQ problem: one is classical and the other comes from our reduction.

In the case of MLD there exist families of codes that plays a similar role in the MLD context, for example the one obtained via the reductions in [6] and the one obtained with our results. We can formalise this in the following theorem.

Theorem 23 *Let $\mathcal{C}$ be the family of codes defined by parity-check matrices as in* (11).

If we can solve all MLD instances for $\mathcal{C}$ in polynomial time, then we can solve in polynomial time all instances of MLD, and so, P=NP.

Regarding the relation between MLD and MQ, we remark that two NP-complete problems might be not isomorphic (see [40] for a formal definition of polynomial-time isomorphism and for the Berman-Hartmanis conjecture on isomorphic NP problems). We rephrase here [40, Th.1] since it is needed in our subsequent discussion.

Theorem 24 ([40, Th.1]) *If there are two length-increasing invertible p-reductions, one of A to B and the other of B to A, then A and B are isomorphic.*

In our case A and B are MLD and MQ, while the two reductions are α and β, which are polynomial-time length-increasing reductions. However, only α can be inverted, since β requires the reduction in MQ instances into standard-form systems (a many-to-one reduction), hence the hypotheses of Theorem 24 are not completely satisfied. To obtain a one-to-one reduction from MQ to MLD, we can modify the definition of standard-form systems by adding new equations containing the information about the original MQ instance. For example, we can consider an additional set of $\mathsf{m}\left(\binom{\mathsf{n}}{2}+\mathsf{n}+1\right)$ equations and new variables of the form

$$\begin{cases} \bar{\gamma}_{ij}^{(h)} + \gamma_{ij}^{(h)} = 0 & 1 \le h \le \mathsf{m}, 1 \le i < j \le \mathsf{n} \\ \bar{\lambda}_{i}^{(h)} + \lambda_{i}^{(h)} = 0 & 1 \le h \le \mathsf{m}, 1 \le i \le \mathsf{n} \\ \bar{\delta}^{(h)} + \delta^{(h)} = 0 & 1 \le h \le \mathsf{m} \end{cases}$$

namely, these equations specify the monomials' coefficients of the polynomials in the original MQ instance (see (2)).

Once we modify the definition of standard-form instance (and β accordingly), we obtain a one-to-one reduction from MQ to MLD. In this way both α and β satisfy the hypotheses of Theorem 24, thus proving the existence of an isomorphism between MLD and MQ.

Theorem 25 *MLD and MQ are isomorphic.*

This isomorphism shows our claimed equivalence and it implies the importance of studying the security of code-based and multivariate-based schemes by meaning of both methods from Coding Theory and Computational Algebra.

6 Open problems

We highlight here a few directions for future works.

- An investigation of the image of α. Polynomial systems obtained via α have a special form, and by analysing them new hints on the difficulty of MLD could be determined. Similarly for polynomials coming from $\alpha \circ \beta$. In particular, MLD instances obtained from code-based cryptosystems are of particular interests. We recall that, in terms of MLD, most cryptographic code-based schemes are modelled as triples (H, s, t) with H defined as a permuted version of the parity-check matrix of an algebraic code (for instance, in Classic McEliece [8], H hides the parity-check matrix of a binary irreducible Goppa code). Loosely speaking, only who can solve the instance can decrypt a ciphertext, and, to the current knowledge, this is feasible only to those who know the hidden Goppa code. However, some vulnerabilities may be revealed by looking at the associated MQ instance.
- Analogously, for codes coming from the image of β and $\beta \circ \alpha$. In particular, similar to above, it would be interesting to focus on MQ instances corresponding to multivariate-based cryptosystems. For instance, in the Digital Signature Scheme Rainbow [14] the public key is a masked version of a quadratic Boolean polynomial system for which there exists a fast solving algorithm based on Gaussian elimination. It appears that only those who know the original form of the polynomial system are capable of signing messages. However, it may be the case that hidden vulnerabilities are disclosed by applying β to these instances.
- Even though apparently similar to the directions hinted above, the third future work we propose is even more linked to the security of code-based and multivariate-based cryptosystems. As already introduced, both post-quantum cryptographic families rely on the security of two kinds of problems, the first is the NP-hard problem of solving a generic instance (i.e. MLD for code-based ciphers and MQ for multivariate-based ciphers), the second is the ability of distinguishing between a generic instance and the masked easy-to-solve underlying algebraic instance (e.g. the Goppa code hidden inside a Classic McEliece public key or the multi-level Oil &Vinegar system hidden inside a Rainbow public key). Instead of blindly applying our reductions to cryptographic public keys, it would be important to model the precise problem of extrapolating the private keys from the public keys, and thus study the complexity of attacking code-based and multivariate-based schemes by understanding their key-generation algorithms.

Acknowledgements The publication was created with the co-financing of the European Union - FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062 / 2021. The core of this work is contained in the second author's MSC thesis, supervised by the other authors. Part of this work was presented in 2019 by the first author at the Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica of the University of L'Aquila, Italy. The first author is a member of the INdAM Research group GNSAGA.

The authors would like to thank Stefano Baratella and Roberto Sebastiani for their helpful comments.

Author Contributions The share of the author's contributions in this paper is equal.

Funding Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and materials On behalf of all authors, the corresponding author states that this manuscript has no associated data.

Declarations

Competing interests On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (2018)
2. Stinson, D., Paterson, M.: Cryptography Theory and Practice (Fourth Edi). CRC Press, Taylor & Francis Group, Boca Raton (2019)
3. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994). Ieee
4. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post-Quantum Cryptography. Springer, Berlin (2009)
5. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a Cryptographic Perspective. Springer, Berlin (2002)
6. Berlekamp, E., McEliece, R., Van Tilborg, H.: On the inherent intractability of certain coding problems (corresp.). IEEE Trans. Inf. Theory **24**(3), 384–386 (1978)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. freeman San Francisco (1979)
8. Bernstein, D.J., Chou, T., Lange, T., von Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., et al.: Classic McEliece: conservative code-based cryptography. Submission to the NIST's post-quantum cryptography standardization process (2017)
9. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber algorithm specifications and supporting documentation. Submission to the NIST's post-quantum cryptography standardization process (2017)
10. Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P., Whyte, W., Zhang, Z.: NTRU algorithm specications and supporting documentation. Submission to the NIST's post-quantum cryptography standardization process (2019)
11. Basso, A., Bermudo Mera, J.M., D'Anvers, J.-P., Karmakar, A., Roy, S.S., Van Beirendonck, M., Vercauteren, F.: SABER: Mod-LWR based KEM (Round 3 submission). Submission to the NIST's post-quantum cryptography standardization process (2020)
12. Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium - Algorithm Specifications and Supporting Documentation (Version 3.1) (2021). https://pq-crystals.org/dilithium/resources.shtml
13. Fouque, P.-A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON: Fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST's post-quantum cryptography standardization process (2020)
14. Ding, J., Chen, M.-S., Petzoldt, A., Schmidt, D., Yang, B.-Y., Kannwischer, M., Patarin, J.: Rainbow - algorithm specification and documentation. Submission to the NIST's post-quantum cryptography standardization process (2015)
15. Guruswami, V., Vardy, A.: Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. IEEE Trans. Inf. Theory **51**(7), 2249–2256 (2005)
16. Karp, R.M.: Reducibility Among Combinatorial Problems. Springer, Berlin (1972)
17. Bruck, J., Naor, M.: The hardness of decoding linear codes with preprocessing. IEEE Trans. Inf. Theory **36**(2), 381–385 (1990)

18. Lobstein, A.: The hardness of solving subset sum with preprocessing. IEEE Trans. Inf. Theory **36**(4), 943–946 (1990)
19. Vardy, A.: The intractability of computing the minimum distance of a code. IEEE Trans. Inf. Theory **43**(6), 1757–1766 (1997)
20. Cheng, Q.: Hard problems of algebraic geometry codes. IEEE Trans. Inf. Theory **54**(1), 402–406 (2008)
21. Gandikota, V., Ghazi, B., Grigorescu, E.: On the NP-hardness of bounded distance decoding of Reed-Solomon codes. In: 2015 IEEE International Symposium on Information Theory (ISIT), pp. 2904–2908 (2015). IEEE
22. Peterson, W.: Encoding and error-correction procedures for the Bose-Chaudhuri codes. IRE Trans. Inf. Theory **6**(4), 459–470 (1960)
23. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Coding Thv **4244**, 114–116 (1978)
24. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Prob. Contr. Inform. Theory **15**(2), 157–166 (1986)
25. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar signature schemes. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 206–222 (1999). Springer
26. Casanova, A., Faugere, J.-C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: GeMSS: a great multivariate short signature. Submission to the NIST's post-quantum cryptography standardization process (2017)
27. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Workshop on the Theory and Application of of Cryptographic Techniques, pp. 419–453 (1988). Springer
28. Patarin, J.: Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 33–48 (1996). Springer
29. Dennis Jr, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM (1996)
30. Schnabel, R.B., Frank, P.D.: Tensor methods for nonlinear equations. SIAM J. Numer. Anal. **21**(5), 815–843 (1984)
31. Cox, D., Little, J., O'Shea, D.: Using algebraic geometry, ser. Graduate Texts in Mathematics. Springer (2005)
32. Bucero, M.A., Mourrain, B.: Border basis relaxation for polynomial optimization. J. Symbolic Comput. **74**, 378–399 (2016)
33. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)
34. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Programm. **96**(2), 293–320 (2003)
35. Li, T.-Y.: Numerical solution of multivariate polynomial systems by homotopy continuation methods. Acta Numerica **6**, 399–436 (1997)
36. Verschelde, J.: Polynomial homotopies for dense, sparse and determinantal systems. arXiv preprint math/9907060 (1999)
37. Davenport, M.A., Romberg, J.: An overview of low-rank matrix recovery from incomplete observations. IEEE J. Selected Topics Signal Process. **10**(4), 608–622 (2016)
38. Grigoriev, D., Pasechnik, D.V.: Polynomial-time computing over quadratic maps i: sampling in real algebraic sets. Comput. Complex. **14**(1), 20–52 (2005)
39. Shamir, A., Kipnis, A.: Cryptanalysis of the HFE public key cryptosystem. In: Advances in Cryptology, Proceedings of Crypto, vol. 99 (1999)
40. Berman, L., Hartmanis, J.: On isomorphisms and density of NP and other complete sets. SIAM J. Comput. **6**(2), 305–322 (1977)