Pseudo-Oil Subspaces and the Geometry of Underdetermined MQ Problems

Massimo Ostuzzi

Ruhr University Bochum, Bochum, Germany

Abstract. The concrete security of multivariate post-quantum signature schemes is coming under increasing scrutiny as the NIST standardisation process for additional signatures approaches its final stages. Among the leading candidates, the security of MAYO and QR-UOV relies on the hardness of the underdetermined multivariate quadratic (MQ) problem. This work revisits Hashimoto's algorithm for solving underdetermined systems of MQ equations, reinterpreting it as a computation of a pseudo-oil subspace. In light of this geometric point of view, we design a new algorithm that, by computing richer pseudo-oil structures, distributes algebraic work across more than two Gröbner Basis steps, subdividing the initial MQ problem into multiple subproblems that can be solved separately, while linearising multiple equations. Optimising a set of discrete parameters, we select the best trade-off between algebraic solving and combinatorial search. Concretely, our approach lowers the cost of the direct attack against Security Level I parameter sets of MAYO and QR-UOV by 7 and 8 bits, respectively.

1 Introduction

With the advent of quantum computers, the National Institute of Standards and Technology (NIST) launched a standardization competition for quantum-resistant cryptographic systems. This initiative resulted in numerous submissions, four of which were selected as finalists, three based on lattice problems [SAB+20,LDK+20,PFH+20,HBD+20]. Subsequently, NIST initiated an additional call for digital signature schemes, explicitly requiring cryptographic security assumptions independent of lattices to diversify the portfolio of post-quantum standards.

MAYO [Beu22,BCC+24] and QR-UOV [FIKT21,FIH+24] are two prominent hash-and-sign digital signatures that have attracted attention for their performances both in terms of efficiency and compactness. Both schemes are based on UOV trapdoor multivariate quadratic (MQ) maps, introduced by Patarin at the Dagstuhl Workshop in 1997 and revised together with Kipnis and Goubin in 1999 [KPG99]. Forging a signature in either scheme reduces to inverting such an MQ map. The trapdoor information is a special subspace, called *oil space*, that allows to reduce the inversion problem of such MQ map to a linear system of equations. Without the knowledge of the trapdoor information, it is conjectured that inverting this MQ map is equivalent to the presumed computational

hardness of solving systems of MQ equations over finite fields. As these schemes come closer to deployment in real-world scenarios, assessing their concrete security against both classical and quantum adversaries becomes of compelling importance.

An instance of the *MQ problem* comprises m quadratic polynomials in n variables over a field $\mathbb{F}_q$. Accordingly one distinguishes the *underdetermined* $n > m$, *square* $n = m$ and *overdetermined* $n < m$ regimes. The square MQ problem is known to be NP-hard [GJ79,BG94], but it appears not to provide enough flexibility for UOV-like signatures. In the context of MAYO and QR-UOV, a successful forgery reduces to solving an underdetermined MQ problem, which motivates the cryptanalytic study of algorithms tailored to this specific regime.

Heavily Underdetermined Solvers. The first attempt to solve underdetermined MQ problems was designed by Kipnis, Patarin and Goubin in their seminal paper [KPG99] about Unbalanced Oil and Vinegar (UOV), as a tangential contribution. Their polynomial time algorithm can solve *heavily underdetermined* MQ systems by reducing the problem to a linear system of equations, provided $n \geq m(m+1)$ and the field has even characteristic. In a nutshell, their approach consists of computing a change of variables that reduces the MQ problem to a linear system of equations, without the use of the oil subspace. This strategy has been the blueprint for all subsequent algorithms for underdetermined MQ problems. Courtois, Goubin, Meier and Tacier [CGMT02] extended the algorithm to odd characteristics, despite achieving a slightly worse bound for the minimum number of variables. This polynomial time technique was later refined by Miura, Hashimoto and Takagi [MHT13], relaxing the requirement on the number of variables to $n \geq m(m+1)/2$ while retaining the applicability to odd characteristic.

Moderately Underdetermined Regime. Thomae and Wolf [TW12] were the first to propose an algorithm tailored to cryptanalytic applications, targeting *moderately underdetermined* MQ instances. Their analysis expresses the running time as the solution of a constrained optimisation problem, where the constraints depend on the relation between the number of variables n and the number of equations m. Subsequent works by Cheng, Hashimoto, Miura and Takagi [MHT13,CHMT14] refined this approach: by relaxing some of the optimisation constraints, they were able to choose parameters more aggressively, yielding improved practical performance. More recently, Furue, Nakamura and Takagi [FNT21] introduced a hybrid strategy that combines these algebraic techniques with a *guessing* step. This adds an extra optimisation parameter k, the number of guessed variables, and requires enumerating assignments for k coordinates. The guessing-enhanced method was proven to outperform the corresponding non-guessing variants in certain parameter regimes.

Hashimoto's Algorithm. The state-of-the-art classical approach for moderately underdetermined MQ instances is Hashimoto's algorithm [Has21]. Follow-

ing the blueprint strategy of [KPG99], Hashimoto's method is divided into two phases. The first phase computes a linear change of coordinates that partitions the original instance into two smaller subproblems that can be solved sequentially, linearising one quadratic equation on the way, while the second phase takes care of solving these smaller MQ problems. The running time is again the result of an optimisation problem, governed by a small set of discrete parameters: the sizes of the two smaller subproblems, and the number of guessed variables, as in the approach by [FNT21]. While this design yields strong practical performance, its rigid partitioning and the restriction to a single linearisation limit the available trade-offs between algebraic solving and combinatorial search.

Just Guess' Extended Linearisation. Later contributions explored these trade-offs more aggressively. In particular, Just Guess by May, Ostuzzi and Ressler [MOR25] has shown that extended linearisation combined with heavy guessing can significantly alter the balance between algebraic computation and enumeration, to the extreme point of even turning the problem into a purely combinatorial one, which does not require the use of Gröbner Basis (GB) algorithms [CKPS00,BFP09,BFP12,MS17]. While this does not reduce the bit security of MQ-based schemes *classically*, it is particularly favourable in the *quantum* setting, where Grover's search [Gro98] can substitute classical enumeration.

1.1 Our contribution

The extended linearisation technique of Just Guess motivates a systematic investigation of whether relaxing Hashimoto's structural constraints can improve *classical* attack complexities. In this work, we answer in the affirmative. Our improvements stem from a geometric reinterpretation of the algorithms in this line of work, which we believe also addresses a broader issue: the family of underdetermined MQ solvers, despite its cryptanalytic relevance, remains geometrically opaque. All existing algorithms are presented as sequences of algebraic manipulations with little unifying intuition, making it difficult to reason about their design or identify room for improvement. We provide this intuition by introducing *pseudo-oil subspaces*, the natural geometric objects that these algorithms implicitly compute. Making them explicit serves two purposes at once: it clarifies the structure of existing algorithms, and naturally suggests how to improve them. Our pseudo-oil subspaces can be seen as a weaker trapdoor information. They are computable more efficiently than real oil subspaces and, differently from real oil subspaces, they exist for generic MQ maps. While the knowledge of pseudo-oil subspaces is not sufficient to solve the MQ problem in polynomial time, it still allows to reduce its solving complexity. In particular, improving Hashimoto's algorithm reduces to computing richer pseudo-oil structures, which is precisely what our algorithm achieves.

Concretely, we bring the following technical contributions:

1. We introduce the concept of pseudo-oil subspaces, providing a unifying geometric language for the entire family of underdetermined MQ solvers.

2. We improve the state of the art along two independent axes: distributing the algebraic workload across multiple balanced MQ subproblems, and linearising multiple equations instead of one, each controlled by a free optimisation parameter.
3. We prove an average-case complexity formula valid under a mild heuristic assumption for both Hashimoto's and our algorithm.
4. We reduce the forgery complexity for Security Level I of MAYO and QR-UOV.

The complexity of our algorithm is obtained by solving an optimisation problem over discrete parameters. In particular, we introduce a free parameter ℓ for the number of linearised equations, and a set of p parameters controlling the sizes of the smaller MQ subproblems produced by the knowledge of pseudo-oil subspaces. Optimising these parameters results in concrete improvements for forgery complexity for Security Level I parameter sets, both for MAYO and QR-UOV, see Table 1. Our algorithm is also more memory efficient than Hashimoto's algorithm. Indeed, adopting different memory access models that penalise heavier memory usage results in more significant improvements, up to a factor 2^{18} in the square-root model.

Table 1: Impact of our algorithm on Security Level I parameter sets of MAYO and QR-UOV.

Scheme	(n, m, q)	p	ℓ	Ours	Hashimoto
MAYO	$(860, 78, 16)$	8	14	**146**	153
QR-UOV	$(840, 100, 7)$	5	13	**172**	180

Implementation. We implemented a proof of concept of our algorithm in SageMath. The code can be found in the git repository `https://github.com/maxostuzzi/MQ_chemistry`. In the repository, one can find our optimisation script that computes the optimal choice of parameters for the attack. Additionally, we have a script that generates small toy examples and allows to visualise the solving strategy.

Organisation of the paper. We provide the preliminary definitions and lemmas in Section 2. In Section 3, we revisit Hashimoto's algorithm using our geometric framework, while Section 4 presents our new algorithm. Section 5 is devoted to the complexity analysis our algorithm. Finally, Section 6 reports the optimisation outcomes and the resulting bit security estimates for MAYO and QR-UOV under different memory access models.

Acknowledgments. Massimo Ostuzzi is part of the Quantum-Safe Internet (QSI) ITN, which received funding from the European Union's Horizon-Europe programme as Marie Sklodowska-Curie Action (PROJECT 101072637 - HORIZON - MSCA-2021-DN-01).

2 Preliminaries

Linear Algebra. Let V and W be vector spaces over $\mathbb{F}_q$. We use the notation $V \simeq W$ to indicate that V and W are isomorphic as vector spaces. We denote by 0 the zero vector, and the ambient vector space will be clear from the context. Let $\mathbf{v}_1, \ldots, \mathbf{v}_k \in V$. We denote by $\langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$ the linear subspace of V spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_k$. If $W \subseteq V$, the quotient space V/W is defined as the set of cosets

$$V/W := \{\mathbf{v} + W : \mathbf{v} \in V\}.$$

Throughout the paper, we will often identify a coset $\mathbf{v} + W \in V/W$ with one of its representatives $\mathbf{v} \in V$, meaning that one can take any lift of $\mathbf{v} + W$ to V. We denote by $V \oplus W$ the direct sum of V and W.

MQ Maps. Throughout the paper, we restrict our attention to *homogeneous* quadratic polynomials, i.e. polynomials in which every monomial has total degree two. Indeed, our primary aim is cryptanalysis of multivariate signature schemes, in which only homogeneous polynomials are used. Our result can be extended to non-homogeneous polynomials with minor modifications.

Let us fix positive integers $n, m \in \mathbb{Z}$ and a finite field $\mathbb{F}_q$. Let $V \subseteq \mathbb{F}_q^n$ and $W \subseteq \mathbb{F}_q^m$ be linear subspaces. A *multivariate quadratic map* (MQ map) is a function

$$\begin{aligned} \mathcal{P} : V &\to W \\ \mathbf{x} &\mapsto (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}))^t, \end{aligned}$$

where each p_i is a homogeneous quadratic polynomial in the variables $\mathbf{x} = (x_1, \ldots, x_n)$. The *polar map* $\mathcal{P}' : V \times V \to W$ associated to $\mathcal{P}$ is a bilinear map defined as

$$\mathcal{P}'(\mathbf{x}, \mathbf{y}) = \mathcal{P}(\mathbf{x} + \mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y}).$$

An *isotropic vector* for an MQ map $\mathcal{P}$ is a nonzero vector $\mathbf{v} \in V$ satisfying $\mathcal{P}(\mathbf{v}) = 0$. An *isotropic*[1] *subspace* of $\mathcal{P}$ is a linear subspace $O \subset V$ such that $\mathcal{P}(\mathbf{o}) = 0$ for all $\mathbf{o} \in O$, i.e. all its vectors are isotropic. Two vectors $\mathbf{v}_1, \mathbf{v}_2 \in V$ are *orthogonal with respect to* $\mathcal{P}$ if $\mathcal{P}'(\mathbf{v}_1, \mathbf{v}_2) = 0 \in W$.

Lemma 1. *Two isotropic vectors $\mathbf{v}_1, \mathbf{v}_2 \in V$ for $\mathcal{P}$ span an isotropic subspace for $\mathcal{P}$ if and only if they are orthogonal with respect to $\mathcal{P}$.*

[1] In the literature, one often finds isotropic subspaces defined as subspaces containing *some* isotropic vectors, and *totally* isotropic for subspaces in which every vector is isotropic. We do not make this distinction.

Proof. The span of $\mathbf{v}_1$ and $\mathbf{v}_2$ is an isotropic subspace if and only if for all $\alpha, \beta \in \mathbb{F}_q$ we have $\mathcal{P}(\alpha\mathbf{v}_1 + \beta\mathbf{v}_2) = 0$. By the definition of polar map and isotropic vectors, this is true if and only if

$$\mathcal{P}(\alpha\mathbf{v}_1 + \beta\mathbf{v}_2) = \alpha\beta\mathcal{P}'(\mathbf{v}_1, \mathbf{v}_2) + \alpha^2 \underbrace{\mathcal{P}(\mathbf{v}_1)}_{0} + \beta^2 \underbrace{\mathcal{P}(\mathbf{v}_2)}_{0} = 0$$

for all $\alpha, \beta \in \mathbb{F}_q$, i.e. if and only if $\mathcal{P}'(\mathbf{v}_1, \mathbf{v}_2) = 0$. □

Generalising Lemma 1, we obtain the following result.

Lemma 2. *A vector subspace $O \subseteq V$ spanned by $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_\ell$ is isotropic for $\mathcal{P}$ if and only if all its basis vectors are isotropic for $\mathcal{P}$ and pairwise orthogonal with respect to $\mathcal{P}$.*

Remark 1. The notion of isotropic vector and subspace is not new in the context of cryptography. In fact, the oil subspace of UOV MQ maps is exactly a large secret isotropic subspace.

Projections and Shifts. For any positive integer n, we denote by $[n]$ the set $\{1, \ldots, n\}$.

We now introduce some non-standard notation that will be useful throughout the whole paper to keep our formulae compact. Let $\mathcal{P} : V \to \mathbb{F}_q^m$ be an MQ map as above, $\mathcal{J} \subset [m]$ a set of indices, $\mathbf{v} \in V$ and $\mathbf{w} \in \mathbb{F}_q^m$ any vectors. Let us write $\pi_\mathcal{J}$ for the projection map from $\mathbb{F}_q^m$ onto the span of the canonical basis vectors of $\mathbb{F}_q^m$ with indices in $\mathcal{J}$. We denote by $\mathbf{w}_\mathcal{J} = \pi_\mathcal{J}(\mathbf{w})$ the projection of $\mathbf{w}$. We denote by $\mathcal{P}_{\mathbf{v},\mathcal{J}}$ the *shifted and projected* MQ map defined as

$$\begin{aligned} \mathcal{P}_{\mathbf{v},\mathcal{J}} : V &\to \pi_\mathcal{J}(\mathbb{F}_q^m) \\ \mathbf{x} &\mapsto \pi_\mathcal{J}(\mathcal{P}(\mathbf{x} + \mathbf{v})). \end{aligned} \tag{1}$$

If $\mathcal{J} = [m]$ (resp. $\mathbf{v} = 0 \in \mathbb{F}_q^n$), we drop the superfluous notation and use $\mathcal{P}_\mathbf{v}$ (resp. $\mathcal{P}_\mathcal{J}$) for the shifted (resp. projected) MQ map.

MQ Problems. Let $\mathcal{P} : V \to W$ be an MQ map, where $\dim V = n$ and $\dim W = m$. An instance of the *MQ*(n, m) *problem* consists of the tuple $(\mathcal{P}, V, \mathbf{t})$, with $\mathbf{t} \in W$ called *target*. The vector $\mathbf{s} \in V$ is a solution of the instance if $\mathcal{P}(\mathbf{s}) = \mathbf{t}$. The MQ$(n, m)$ problem is called *overdetermined*, *square* or *underdetermined* when $n < m$, $n = m$ or $n > m$, respectively. We write GB(n, m) for the bit complexity of solving an instance of the MQ(n, m) problem in expectation using algebraic methods, i.e. Gröbner Basis algorithms. The underlying field $\mathbb{F}_q$ of the MQ problem will be clear from context or specified otherwise.

3 A Geometric Interpretation of Hashimoto's Algorithm

In this section, we revisit Hashimoto's algorithm [Has21]. The algorithm takes as input an instance $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$ of the MQ(n, m) problem, with $n > m$, where

$\mathbf{t} \in \mathbb{F}_q^m$ is the target. Additionally, the algorithm takes as input two parameters b and k, which are positive integers such that $b + k \leq m - 1$. The choice of these parameters determines the complexity of Hashimoto's algorithm and will be discussed in Section 3.3.

Hashimoto's algorithm is divided in two stages, which we call *Centrifugation*[2] *phase* and *Extraction phase*. In the Centrifugation phase, the goal is to compute an m-dimensional linear subspace $O \subset \mathbb{F}_q^n$ that admits a direct sum decomposition $O = O_0 \oplus \tilde{O}$ with $\dim O_0 = m - (b + k)$ and $\dim \tilde{O} = b + k$ satisfying the following properties:

1. O_0 is isotropic for $\mathcal{P}_{\mathcal{L}}$, i.e. $\mathcal{P}_{\mathcal{L}}(O_0) = 0$,
2. $\tilde{O}$ is orthogonal to O_0 with respect to $\mathcal{P}'_{\mathcal{B}}$, i.e. $\mathcal{P}_{\mathcal{B}}(\tilde{O}, O_0) = 0$,

where $\mathcal{B} = [b] \subset [b + 1] = \mathcal{L}$. We call the linear subspace O *pseudo-oil subspace*, O_0 and $\tilde{O}$ *pseudo-oil summands*.

In the Extraction phase, these particular properties of the pseudo-oil subspace allow to construct (or *extract*) a preimage of $\mathbf{t}$ progressively by solving *two* smaller MQ subproblems, using Gröbner Basis algorithms. Moreover, thanks to the orthogonality property of O_0, it is possible to efficiently eliminate one variable and one equation from the second MQ subproblem, thereby reducing its solving complexity.

We start by describing the Centrifugation phase. We then illustrate the Extraction phase in two different (but equivalent) formulations: the *traditional* and the *geometric* formulation. We believe that the traditional formulation is easier to understand, especially for those already familiar with Hashimoto's algorithm, while the geometric one might be of interest for further improving the algorithm in the future. Finally, we give the time complexity analysis of the algorithm.

3.1 Centrifugation

The Centrifugation consists of identifying inside $\mathbb{F}_q^n$ the pseudo-oil subspace O and separating it into the pseudo-oil decomposition $O_0 \oplus \tilde{O}$. We compute such decomposition by constructing a basis for each pseudo-oil summand, starting from O_0. We give the pseudocode for the Centrifugation in Algorithm 1.

Computing O_0. To construct a basis of the pseudo-oil summand O_0, we start by computing the first pseudo-oil vector $\mathbf{o}_1 \in \mathbb{F}_q^n$ as any isotropic vector for the projected MQ map $\mathcal{P}_{\mathcal{L}}$. In other words, we solve for a *non-zero* solution of the projected MQ problem $(\mathcal{P}_{\mathcal{L}}, \mathbb{F}_q^n, 0)$, which is an $\mathrm{MQ}(n, b + 1)$ problem.

[2] Centrifugation is a separation process in which a mixture is spun at high speed in a Centrifuge so that components with different densities separate under the effect of centrifugal force.

Algorithm 1: CENTRIFUGATION

Input: MQ map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$.

Output: Pseudo-oil subspace O and summands $O_0, \tilde{O}$ satisfying the defining conditions.

1 Compute a basis for a pseudo-oil summand O_0 by solving $m - (b + k)$ MQ problems as in Equation (3).

2 Compute a basis of a pseudo-oil summand $\tilde{O}$ by solving $b + k$ linear systems as in Equation (4).

3 $O \leftarrow O_0 \oplus \tilde{O}$.

4 **return** $O, O_0, \tilde{O}$.

Then, we compute the second pseudo-oil vector by finding a non-zero solution in $\mathbb{F}_q^n/\langle \mathbf{o}_1 \rangle$ to the system of equation

$$\begin{cases} \mathcal{P}_{\mathcal{L}}(\mathbf{x}) = 0 \\ \mathcal{P}'_{\mathcal{L}}(\mathbf{x}, \mathbf{o}_1) = 0 \end{cases} \tag{2}$$

Note that both $\mathcal{P}_{\mathcal{L}}$ and $\mathcal{P}'_{\mathcal{L}}$ are well-defined on $\mathbb{F}_q^n/\langle \mathbf{o}_1 \rangle$, since $\mathbf{o}_1$ is isotropic for $\mathcal{P}_{\mathcal{L}}$.

The system in Equation (2) consists of two types of equations. The first $b+1$ are quadratic equations ensuring that the solution $\mathbf{x}$ is an isotropic vector for the projected MQ map $\mathcal{P}_{\mathcal{L}}$. The next $b + 1$ are linear equations ensuring that the vector subspace spanned by the solution $\mathbf{x}$ and $\mathbf{o}_1$ is an isotropic subspace, see Lemma 1. Moreover, note that using the linear equations we can eliminate $b+1$ variables from the first $b+1$ quadratic equations. Therefore, computing o_2 amounts to solving an MQ$(n - (b + 2), b + 1)$ problem, because the system of equations in 3 consists of $b + 1$ quadratic equations in

$$\dim(\mathbb{F}_q^n/\langle \mathbf{o}_1 \rangle) - (b + 1) = n - (b + 2)$$

variables.

Proceeding analogously, we find the pseudo-oil vector $\mathbf{o}_i$ for $i \leq m - (b + k)$ by solving for a non-zero solution in $\mathbb{F}_q^n/\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1} \rangle$ of the MQ problem

$$\begin{cases} \mathcal{P}_{\mathcal{L}}(\mathbf{x}) = 0 \\ \mathcal{P}'_{\mathcal{L}}(\mathbf{x}, \mathbf{o}_j) = 0 \qquad 1 \leq j < i \end{cases} \tag{3}$$

Note again that both $\mathcal{P}_{\mathcal{L}}$ and $\mathcal{P}'_{\mathcal{L}}$ are well-defined on the quotient since the pseudo-oil vectors $\mathbf{o}_j$ are isotropic and orthogonal to each other with respect to $\mathcal{P}_{\mathcal{L}}$. The number of variables in this system of equations is

$$\dim(\mathbb{F}_q^n/\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1} \rangle) - (i - 1)(b + 1) = n - (i - 1)(b + 2),$$

implying that computing $\mathbf{o}_i$ amounts to solving an MQ$(n - (i - 1)(b + 2), b + 1)$ problem.

Computing $\tilde{O}$. The pseudo-oil summand $\tilde{O}$ is a $(b+k)$-dimensional subspace orthogonal to O_0 with respect to $\mathcal{P}_{\mathcal{B}}$. While computing the oil subspace O_0 itself required solving an MQ problem, the computation of the pseudo-oil summand $\tilde{O}$ can be done efficiently, as it only requires solving linear systems of equations.

The first pseudo-oil vector $\mathbf{o}_{m-(b+k)+1} \in \mathbb{F}_q^n$ of $\tilde{O}$ can be computed by solving the linear system

$$\mathcal{P}'_{\mathcal{B}}(\mathbf{x}, O_0) = 0 \tag{4}$$

for a non-zero solution in $\mathbb{F}_q^n/O_0$. This linear system has

$$b \dim O_0 = b(m-(b+k))$$

equations in $\dim \mathbb{F}_q^n/O_0 = n - \dim O_0 = n - (m-(b+k))$ variables.

Once the first $i-1$ pseudo-oil vectors of $\tilde{O}$ have been computed, one computes the pseudo-oil vector $\mathbf{o}_{m-(b+k)+i}$ by solving the linear system in Equation (4) for a non-zero solution in $\mathbb{F}_q^n/(O_0 \oplus \langle \mathbf{o}_{m-(b+k)+1}, \ldots, \mathbf{o}_{m-(b+k)+i-1} \rangle)$, which has dimension $n-(m-(b+k))-(i-1)$. Proceeding in this way, we efficiently compute a basis for a pseudo-oil summand $\tilde{O}$. Note that all these linear systems are well-defined on their corresponding quotient spaces since O_0 is an isotropic subspace and each pseudo-oil vector of $\tilde{O}$ is by definition orthogonal to O_0 with respect to $\mathcal{P}_{\mathcal{B}}$.

3.2 Extraction: Traditional Formulation

The traditional formulation of the Extraction phase makes use of a change of variables S on the domain of $\mathcal{P}$. The map $S \in \mathrm{GL}_n(\mathbb{F}_q)$ is defined as any invertible linear map satisfying

$$\mathbf{e}_i \overset{S}{\mapsto} \mathbf{o}_i \quad \text{for } i = 1, \ldots, m,$$

where we denote by $\mathbf{e}_i$ the canonical basis vectors. In words, we define $S \in \mathrm{GL}_n(\mathbb{F}_q)$ as any invertible linear map sending the first m canonical basis vectors to the ordered basis of the pseudo-oil subspace O that we have computed during the Centrifugation phase.

In the traditional formulation of the Extraction phase, we consider the *transformed MQ map* $\tilde{\mathcal{P}} = \mathcal{P} \circ S$. Moreover, in this section, we abuse the notation denoting by $\tilde{\mathcal{P}}$ the transformed MQ map restricted to $\langle \mathbf{e}_1, \ldots, \mathbf{e}_m \rangle$. After finding a solution to the *restricted* MQ problem, we pad it with $n-m$ zeros to obtain a solution in $\mathbb{F}_q^n$. This implies that we are now facing a $\mathrm{MQ}(m, m)$ problem. In Algorithm 2, we provide pseudocode for the traditional formulation of the Extraction phase.

Structure of the Matrices. It is not hard to see that, thanks to the isotropicity and orthogonality properties of the pseudo-oil subspace O, the matrices corresponding to $\tilde{\mathcal{P}}$ have the shape depicted in Figure 1. In particular, the fact that O_0 is isotropic for $\mathcal{P}_{\mathcal{L}}$ creates the zero square block on the top-left corner

Algorithm 2: Traditional Extraction

Input: MQ(n, m) problem $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$; pseudo-oil decomposition $O = O_0 \oplus \tilde{O}$; non-negative integers k and b such that $k + b + 1 \leq m$.

Output: Solution $\mathbf{s} \in \mathbb{F}_q^n$ to $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$, or $\perp$.

1 Compute the change of variables S sending $\mathbf{e}_i$ to $\mathbf{o}_i$ for $i \in \{1, \ldots, m\}$.
2 $\tilde{\mathcal{P}} \leftarrow \mathcal{P} \circ S$ restricted to $\langle \mathbf{e}_i \rangle_{i=1}^{m}$.
3 **for** *all assignments* $\mathbf{k} \in \mathbb{F}_q^k$ *of the last* k *variables* `// Guess step`
4 **do**
5 **if** *the MQ*(b, b) *problem admits a solution* $\mathbf{o}_1$ `// Square step`
6 **then**
7 Eliminate one variable using the linear equation.
8 **if** *the remaining MQ*$(m - b - k - 1, m - b - 1)$ *problem admits a solution* $\mathbf{o}_0$ `// Overdetermined step`
9 **then**
10 Set $\mathbf{u} = (\mathbf{o}_1, \mathbf{o}_0, \mathbf{k})^t$.
11 **return** $\mathbf{s} = S(\mathbf{u}, \mathbf{0}^{n-m})^t$. `// Final Solution`
12 **else**
13 Backtrack to Line 6 and choose an alternative solution $\mathbf{o}_1$ to the MQ(b, b) problem.
14 **return** $\perp$.

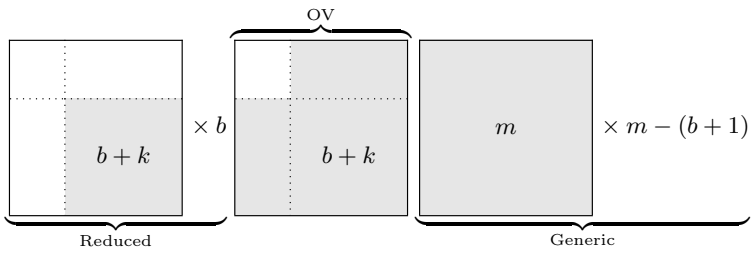

Fig. 1: Structure of the m-by-m matrices of transformed MQ map $\tilde{\mathcal{P}}$.

of the first $b+1$ matrices of $\tilde{\mathcal{P}}$, while the fact that $\tilde{O}$ is orthogonal to O_0 with respect to $\mathcal{P}_{\mathcal{B}}$ creates the rectangular zero blocks in the first b matrices.

In Figure 1, there are three types of matrices, which we call *Reduced, OV* and *Generic*, respectively. We now briefly describe their structure. For notational convenience, let us fix the formal vector of variables $\mathbf{x} = (x_1, \ldots, x_m)^t$, which we additionally split into $\mathbf{x} = (\mathbf{y}, \mathbf{z})^t$, where $\mathbf{y} = (x_1, \ldots, x_{m-(b+k)})^t$ and $\mathbf{z} = (x_{m-(b+k)+1}, \ldots, x_m)^t$.

Starting from the left, Reduced matrices correspond to polynomials in the last $b+k$ variables only. Indeed, the first $m-(b+k)$ variables $\mathbf{y}$ do not appear in the polynomial because all the corresponding coefficients are zero. This implies that the first b matrices of $\tilde{\mathcal{P}}$ together with the first b entries of $\mathbf{t}$ can be treated as a separate $\mathrm{MQ}(b+k, b)$ problem solely in the last $b+k$ variables $\mathbf{z}$, see Figure 2.

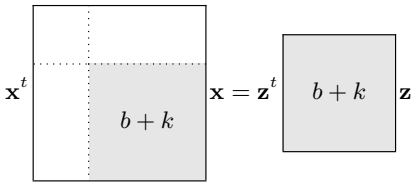

Fig. 2: The structure of Reduced matrices.

Let us now consider OV matrices. Explicitly computing the corresponding polynomial shows that, while the $\mathbf{z}$ variables appear quadratically, the $\mathbf{y}$ variables do not mix with each other, but only linearly with $\mathbf{z}$, see Figure 3. In particular,

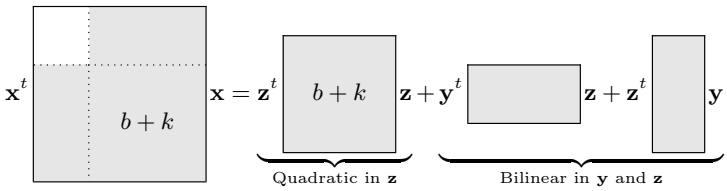

Fig. 3: The structure of OV matrices.

this means that after having assigned the $\mathbf{x}_2$ variables, the polynomial becomes linear in the $\mathbf{x}_1$ variables. This is essentially the same phenomenon that gives rise to OV-based signatures [Pat97] and allows the owner of the secret key to sign efficiently.

Finally, Generic matrices are, as the name suggests, totally generic and do not exhibit any particular structure. Accordingly, the corresponding quadratic polynomial is simply generic.

Having understood the structure of the matrices of $\tilde{\mathcal{P}}$, the Extraction phase is rather straight forward. We divide it in Guess, Square and Overdetermined steps.

Guess Step. The Guess step consists of enumerating assignments for the last k variables of $\mathbf{x}$ until all the subsequent steps are successful. After this variable assignment, we are left with an MQ$(m-k, m)$ problem, i.e. $m-k$ free variables and m equations.

Square Step. As already observed, the first b matrices (Reduced) together with the first b entries of $\mathbf{t}$ form a separate MQ$(b+k, b)$ problem solely in the last $b+k$ variables $\mathbf{z}$. Since during the Guess step k variables of $\mathbf{z}$ have been assigned, we have a separate MQ(b, b) problem in the remaining b variables of $\mathbf{z}$. The Square step precisely consists in solving this MQ(b, b) subproblem using a Gröbner Basis algorithm. If the MQ(b, b) problem does not admit a solution, then we can discard the current assignment for the last k variables and move to the next one. Otherwise, we have found a set of assignments for all the $b+k$ variables of $\mathbf{z}$ satisfying the first b equations. We select one such assignment $\mathbf{o}_1$ and we move to the next step. After the Square step, we still need to solve the last $m-b$ equations, and find an assignment for the $m-b-k$ variables of $\mathbf{y}$.

Overdetermined Step. Since we have assigned all the variables of $\mathbf{z}$ to $\mathbf{o}_1$, the polynomial corresponding to the OV matrix of Figure 1 becomes linear in the $\mathbf{y}$ variables, as shown in Figure 3. Using this linear relation we eliminate one of the variables of $\mathbf{y}$ (say x_{m-b-k}). We are then left with $m-b-1$ equations and $m-b-k-1$ variables of $\mathbf{y}$.

Finally, the last $m-b-1$ equations together with the $m-b-k-1$ free variables of $\mathbf{y}$ form an overdetermined MQ$(m-b-k-1, m-b-1)$ problem. The Overdetermined step consists in solving this problem using a Gröbner Basis algorithm. If the problem admits a solution, then we have found an assignment $\mathbf{u} \in \mathbb{F}_q^m$ for all the variables of $\mathbf{y}$ and $\mathbf{z}$, and we can return the solution $\mathbf{s} = S(\mathbf{u}, \mathbf{0}^{n-m})^t$ to the original MQ(n, m) problem, see Line 9. If the problem does not admit a solution, we then backtrack to the Square step and select an alternative solution $\mathbf{o}_1$ to the MQ(b, b) problem.

The following lemma proves the correctness of Algorithm 2.

Lemma 3. *A vector $\mathbf{x} \in \mathbb{F}_q^n$ solves the transformed MQ(n, m) problem $(\tilde{\mathcal{P}}, \mathbb{F}_q^n, \mathbf{t})$ if and only if the vector $\mathbf{s} = S\mathbf{x}$ solves the original MQ(n, m) problem $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$.*

Proof. The vector $\mathbf{x} \in \mathbb{F}_q^n$ is a solution to $(\tilde{\mathcal{P}}, \mathbb{F}_q^n, \mathbf{t})$ if $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{t}$, and by definition of $\tilde{\mathcal{P}}$ this is equivalent to $\mathcal{P}(S\mathbf{x}) = \mathbf{t}$. This exactly means that $\mathbf{s} = S\mathbf{x}$ is a solution to $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$, the original MQ problem.

3.3 Extraction: Geometric Formulation

Abusing the notation, in this section we denote the MQ map restricted to the pseudo-oil subspace O computed in the Centrifugation phase by $\mathcal{P} : O \to \mathbb{F}_q^m$. As in its traditional formulation, the Extraction phase is divided in Guess step, Square step and Overdetermined step. We give the pseudocode for the geometric formulation of Extraction phase in Algorithm 3.

Algorithm 3: GEOMETRIC EXTRACTION

Input: Pseudo-oil subspaces O_0 and $\tilde{O}$; MQ(m, m) problem $(\mathcal{P}, O, \mathbf{t})$; non-negative integers k and b such that $k + b + 1 \leq m$.
Output: Solution to $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$, or $\perp$.

```
1  Decompose Õ as O_1 ⊕ G_k of dimensions b and k respectively.
2  for k ∈ G_k                                                    // Guess
3  do
4      if the MQ problem in Equation (5) admits a solution o_1 ∈ O_1   // Square
5      then
6          if the MQ problem (P_{o_1+k}, O_0, t) admits a solution o_0 ∈ O_0
               // Overdetermined
7          then
8              return o_0 + o_1 + k.                              // Final Solution
9          else
10             Choose an alternative solution o_1 to the MQ problem in
                Equation (5) and backtrack to Line 6.
11 return ⊥.                                                      // No solution found
```

Guess Step. The Guess step starts by additionally decomposing the pseudo-oil summand $\tilde{O}$ as the direct sum of O_1 of dimension b and G_k of dimension k. We call the subspace G_k the *guessing subspace*. A canonical choice is taking O_1 and G_k to be spans of the first b and the last k vectors of the basis of $\tilde{O}$, respectively. Then, until all subsequent steps succeed, we enumerate vectors $\mathbf{k} \in G_k$.

Square Step. The Square step consists of solving the MQ problem with input

$$(\mathcal{P}_{\mathbf{k},\mathcal{B}}, O_1, \mathbf{t}_{\mathcal{B}})\,. \tag{5}$$

Recall that $\mathcal{P}_{\mathbf{k},\mathcal{B}}$ is defined as the map $\mathbf{x} \mapsto \pi_{\mathcal{B}}(\mathcal{P}(\mathbf{k}+\mathbf{x}))$ and $\mathbf{t}_{\mathcal{B}} = \pi_{\mathcal{B}}(\mathbf{t})$. This is an MQ$(b, b)$ problem, since both the domain and the codomain are b-dimensional vector spaces. If it does not admit a solution, we discard the current vector $\mathbf{k}$ and move to the next one in the Guess step; otherwise, we found a set of solutions and we denote by $\mathbf{o}_1$ one of these.

Overdetermined Step. The overdetermined step amounts to solve the MQ problem with input $(\mathcal{P}_{\mathbf{o}_1+\mathbf{k}}, O_0, \mathbf{t})$. By decomposing the MQ system into the contributions to $\pi_{\mathcal{B}}(\mathbb{F}_q^m)$, $\pi_{\{b+1\}}(\mathbb{F}_q^m)$ and $\pi_{[m]\setminus\mathcal{B}}(\mathbb{F}_q^m)$, we have

$$\begin{cases} \mathcal{P}_{\mathcal{B}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \mathbf{t}_{\mathcal{B}} \\ \mathcal{P}_{\{b+1\}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \mathbf{t}_{\{b+1\}} \\ \mathcal{P}_{[m]\setminus\mathcal{L}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \mathbf{t}_{[m]\setminus\mathcal{B}} \end{cases} \tag{6}$$

The first set of b equations $\mathcal{P}_{\mathcal{B}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \pi_{\mathcal{B}}(\mathbf{t})$ is automatically fulfilled, for any $\mathbf{x} \in O_0$. Indeed, since $\mathbf{o}_1+\mathbf{k} \in \tilde{O}$ and $\mathbf{o}_1$ solves the MQ problem $(\mathcal{P}_{\mathbf{k},\mathcal{B}}, O_1, \mathbf{t}_{\mathcal{B}})$, we can rewrite as

$$\mathcal{P}_{\mathcal{B}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \underbrace{\mathcal{P}'_{\mathcal{B}}(\mathbf{x}, \mathbf{o}_1+\mathbf{k})}_{0 \text{ for orthogonality}} + \underbrace{\mathcal{P}_{\mathcal{B}}(\mathbf{x})}_{0} + \mathcal{P}_{\mathcal{B}}(\mathbf{o}_1+\mathbf{k}) = \mathbf{t}_{\mathcal{B}}.$$

The equation $\mathcal{P}_{\{b+1\}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \mathbf{t}_{\{b+1\}}$ is actually linear. Indeed, we can rewrite the left hand side as

$$\mathcal{P}_{\{b+1\}}(\mathbf{x}+\mathbf{o}_1+\mathbf{k}) = \underbrace{\mathcal{P}_{\{b+1\}}(\mathbf{x})}_{0} + \underbrace{\mathcal{P}_{\{b+1\}}(\mathbf{o}_1+\mathbf{k})}_{\text{constant}} + \underbrace{\mathcal{P}_{\{b+1\}}(\mathbf{x}, \mathbf{o}_1+\mathbf{k})}_{\text{linear in } \mathbf{x}}.$$

Using this linear relation, we can eliminate one variable from the MQ problem. The last set of $m-(b+1)$ equations are quadratic and still to be solved. Thus, the MQ system in Equation (6) is a MQ$(m-b-k-1, m-b-1)$ problem, which is overdetermined whenever $k > 0$. We solve it using a Gröbner Basis algorithm. If it admits a solution $\mathbf{o}_0$, then we return $\mathbf{o}_0+\mathbf{o}_1+\mathbf{k}$ as a solution to the original MQ problem; otherwise, we backtrack to the Square step and select an alternative solution $\mathbf{o}_1$ to the MQ problem in Equation (5).

3.4 Complexity Analysis

The runtime complexity of Hashimoto's algorithm is the result of a discrete optimisation problem, where the goal is to find the values of the parameters b and k that minimise the overall runtime, which is the sum of the runtimes of the Centrifugation and Extraction phases. We start by deriving the constraint of this optimisation problem. Then, we individually analyse the runtimes of the two phases.

Existence of Pseudo-Oil Structures. The first requirement comes from the Overdetermined step. More precisely, in Section 3.3, we eliminate one degree of freedom using the linear relation, therefore O_0 must be at least one-dimensional. This yields the structural constraint $m-(k+b) \geq 1$.

Next, we derive more subtle constraints coming from the Centrifugation. In the Centrifugation, we look for the pseudo-oil summand O_0 and $\tilde{O}$ satisfying certain isotropicity and orthogonality properties. The constraints we derive in

what follows ensure the existence on average of such special linear subspaces for random MQ maps. More precisely, the existence of such pseudo-oil decomposition is equivalent to the solvability of the systems of equations that arise in the Centrifugation phase. In turn, the solvability of these systems of equations can be analysed by counting the number of variables and equations. Since the number of expected solutions to a random system of m equations in n variables is q^{n-m}, we expect that such a system admits a solution whenever $n > m$.

Accordingly, it turns out that the most restrictive requirements show up when computing the last pseudo-oil vector $\mathbf{o}_{m-b-k}$ of O_0 and the last pseudo-oil vector $\mathbf{o}_m$ of $\tilde{O}$. In particular, in Section 3.1 we have counted the number of variables and equations at these steps. Therefore, we have the following inequalities, with the number of equations on the right and the number of variables available on the left.

$$\begin{aligned} n - (m - k - b - 1)(b + 2) &> b + 1 \\ n - m + 1 &> b(m - b - k) \end{aligned} \tag{7}$$

Remark 2. The inequalities in Equation (7) ensure the existence of the pseudo-oil subspaces O_0 and $\tilde{O}$ *in expectation* for random MQ maps. Indeed, if $\mathcal{P}$ is random, then the systems of equations solved during the Centrifugation phase are random as well, and the inequalities guarantee that, on average, each such system admits a non-zero solution.

However, even when these inequalities are satisfied, Algorithm 1 can still fail at some intermediate step. Modelling each system as random, the probability that a system in n variables and m equations is unsolvable is roughly $e^{-q^{n-m}} > 0$.

A failure of this kind should be interpreted as a *local obstruction*: the partial set of pseudo-oil vectors $\mathbf{o}_1, \ldots, \mathbf{o}_{i-1}$ already computed cannot be extended to a basis of O_0 (or, analogously, of $\tilde{O}$), even though pseudo-oil decompositions of the prescribed dimensions typically do exist for $\mathcal{P}$, just supported on a different subspace. In this situation, Algorithm 1 can be *rerandomised* by simply selecting an alternative non-zero solution $\mathbf{o}_1' \notin \langle \mathbf{o}_1 \rangle$ of the projected MQ problem $(\mathcal{P}_{\mathcal{L}}, \mathbb{F}_q^n, 0)$ and recomputing all subsequent vectors. Since each $\mathbf{o}_i$ with $i \geq 2$ is determined by a system that depends on the previously chosen vectors, this single change effectively rerandomises every downstream system, moving the search to a different candidate pseudo-oil subspace. The procedure can be iterated until a complete pseudo-oil basis is found. Using our proof of concept implementation [3], we experimentally tested this analysis and confirm its validity. Indeed, the Centrifugation phase succeeded for all the MQ maps satisfying the inequalities in Equation (7).

Remark 3. We point out that the link between our notation and the one used in the original paper by Hashimoto is $b = a - 1$. Thanks to our more geometric formulation of the Centrifugation phase, our inequalities are even slightly less restrictive than the one in Hashimoto's original paper, where one additional variable was unnecessarily fixed. Indeed, translating our inequalities in Equation (7) in terms of k and a yields the constraints given in [Has21], up to a $+1$.

[3] See `https://github.com/maxostuzzi/MQ_chemistry`.

Centrifugation Phase. An upper bound for the runtime of the Centrifugation phase is

$$(m - (k + b)) \cdot \mathrm{GB}(b + 1, b + 1),$$

ignoring negligible terms coming from Gaussian Elimination. Indeed, one needs to compute $m - (k + b)$ isotropic vectors for $\mathcal{P}_{\mathcal{L}}$ to form the basis of the oil subspace O_0, which amounts to solve $m - (k + b)$ MQ problems as in Equation (3). On the other hand, computing a basis for $\tilde{O}$ only requires solving a linear system of equations, which can be done efficiently in polynomial time. Then, the complexity claim follows.

Remark 4. One might notice that while computing the isotropic vectors of O_0 there are more than $b+1$ variables available. Indeed, to simplify the above complexiy formula, we ignore the additional available variables and use the inequality

$$\mathrm{GB}(n, m) \leq \mathrm{GB}(m, m), \quad \text{for } n \geq m.$$

Extraction Phase. Finally, we analyse the Extraction phase under the following mild heuristic assumption.

Heuristic 1 *Fixing assignments in the Guess or Square steps produces unfiformly random MQ problems.*

Theorem 1. *Under Heuristic 1, the Extraction phase (Algorithms 2 and 3) produces one solution in expected time*

$$q^k \cdot (GB(b, b) + GB(m - b - k - 1, m - b - 1)).$$

Proof. Under Heuristic 1, the MQ problem in the Square and Overdetermined steps are uniformly random. In particular, the Square step produces 1 solution in expectation, since the corresponding MQ problem has the same number of variables and equations. This implies that each guess in the Guess steps leads to one attempt of the Overdetermined step, in expectation. Being uniformly random, the MQ problem in the Overdetermined step admits a solution with probability roughly q^{-k}, since it is overdetermined by k equations. Therefore, one needs q^k independent attempts of the Overdetermined step to find a solution in expectation. Since there are q^k vectors $\mathbf{k}$ in the guessing subspace G_k, the claim follows.

Remark 5. In case of failure of Algorithm 3, one can restart the Centrifugation phase, compute alternative pseudo-oil decompositions as described in Remark 2, and proceed again with another Extraction phase.

4 Enriching Hashimoto's Algorithm

In this section, we show that our novel geometric framework reveals room for improvement in Hashimoto's algorithm. The key insight is that computing a *richer*

pseudo-oil structure with more summands and more linearised equations directly reduces the complexity of both phases. Accordingly, we enrich Hashimoto's algorithm along two independent axes.

The first enrichment allows for more than two pseudo-oil summands, isotropic for different projected MQ maps. This follows the general cryptanalytic principle of distributing and balancing the workload: rather than solving a single square MQ problem of size b, we solve p smaller square MQ problems of sizes $b_1, \ldots, b_p$, reducing the algebraic cost of each step at the price of a richer Centrifugation. We introduce p optimisation parameters $b_1, \ldots, b_p$ governing the sizes of the pseudo-oil decomposition.

The second enrichment imposes a wider orthogonality condition. We introduce the parameter ℓ that allows to eliminate ℓ variables during the Overdetermined step by linearising ℓ equations instead of just one. Our algorithm recovers Hashimoto's as the special case $p = \ell = 1$.

We start by describing the *Enriched Centrifugation* phase and deriving the constraints on the parameters for the existence of the pseudo-oil decomposition. Then, we give both the traditional and geometric formulations of the *Enriched Extraction* phase, which are equivalent.

4.1 Enriched Centrifugation

Let us fix some notation. Let $k, \ell, b_1, \ldots, b_p \in \mathbb{Z}$ be non-negative integers. Let $\mathcal{L} \supset \mathcal{B}_1 \supset \mathcal{B}_2 \supset \cdots \supset \mathcal{B}_p$ be a sequence of index sets defined as

$$\mathcal{L} = \left[\sum_{i=1}^{p} b_i + \ell\right] \quad \text{and} \quad \mathcal{B}_i = \left[\sum_{j=i}^{p} b_j\right], \text{ for } i \in \{1, \ldots, p\}.$$

Let us also denote $B_i = \max \mathcal{B}_i$ and $L = \max \mathcal{L}$.

In the Enriched Centrifugation phase, our goal is to compute a m-dimensional pseudo-oil subspace $O \subset \mathbb{F}_q^n$ admitting a direct sum decomposition as

$$O = \bigoplus_{i=0}^{p-1} O_i \oplus \tilde{O},$$

with $\dim O_i = b_i$ for $i \in \{1, \ldots, p-1\}$ and $\dim \tilde{O} = b_p + k$, that satisfies the properties we describe in the following.

Richer Pseudo-Oil Structures. In the following, we list the four properties that the pseudo-oil subspaces must satisfy to enable the Enriched Extraction phase. There are two isotropicity and two orthogonality properties.

On the isotropicity side, the first is that the subspace O_0 must be isotropic for the projected MQ map $\mathcal{P}_{\mathcal{L}}$, i.e. $\mathcal{P}_{\mathcal{L}}(O_0) = 0$. The second one is that, for each $i \in \{1, \ldots, p-1\}$, the subspace $\bigoplus_{j=0}^{i} O_j$ must be isotropic for the projected MQ

map $\mathcal{P}_{\mathcal{B}_{i+1}}$, namely

$$\mathcal{P}_{\mathcal{B}_{i+1}}\left(\bigoplus_{j=0}^{i} O_j\right) = 0.$$

Remark 6. This second requirement is strictly stronger than asking for the isotropicity of each O_i for $\mathcal{P}_{\mathcal{B}_{i+1}}$ alone. In fact, the isotropicity of $\bigoplus_{j=0}^{i} O_j$ comprises the isotropicity of each O_j for $\mathcal{P}_{\mathcal{B}_{i+1}}$ for $j \in \{0, \ldots, i\}$, but also the pair-wise orthogonality of these pseudo-oil summands with respect to $\mathcal{P}_{\mathcal{B}_{i+1}}$.

Remark 7. The chain of pseudo-oil direct sums $\bigoplus_{j=0}^{i} O_j$ is ascending, while the chain of index sets $\mathcal{B}_{i+1}$ is descending. This ultimately allows for a balanced distribution of the complexity of the Enriched Centrifugation phase: as the index i grows, we are computing larger isotropic subspaces for smaller projected MQ maps.

Regarding orthogonality, $\bigoplus_{j=0}^{i} O_j$ must be orthogonal to $\bigoplus_{j=i+1}^{p-1} O_j \oplus \tilde{O}$ with respect to $\mathcal{P}_{\mathcal{B}_{i+1}}$, for $i \in \{0, \ldots, p-2\}$. Lastly, $\bigoplus_{j=0}^{p-1} O_j$ must be orthogonal to $\tilde{O}$ with respect to $\mathcal{P}_{\mathcal{B}_p}$.

Remark 8. Note that, for $p = \ell = 1$, these properties reduce to the ones in Section 3. In particular, the first two properties collapse into the single isotropicity condition for O_0, while the last one collapses into the single orthogonality condition between O_0 and $\tilde{O}$.

Computing Richer Pseudo-Oil Structures. We give the pseudocode for the Enriched Centrifugation phase in Algorithm 4. The strategy to compute

Algorithm 4: Enriched Centrifugation

Input: MQ map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$.
Output: Pseudo-oil subspaces $O, O_0, \ldots, O_{p-1}, \tilde{O}$ satisfying the defining conditions.

for $i = 0, \ldots, p-1$ **do**
  Compute a basis of O_i by solving MQ problems as in Equation (9).
Compute a basis of $\tilde{O}$ by solving the linear system of equations in Equation (10).
$O \leftarrow \bigoplus_{i=0}^{p-1} O_i \oplus \tilde{O}$.
return $O, \{O_i\}_{i=0}^{p-1}, \tilde{O}$.

a pseudo-oil subspace satisfying these properties is essentially the same as the one presented in Section 3.1, adapted to the properties described in the previous subsubsection. More precisely, one still needs to proceed by sequentially constructing bases for each subspace, starting from O_0 and ending with $\tilde{O}$.

Starting from O_0, one solves for a nonzero solution the MQ problem $(\mathcal{P}_{\mathcal{L}}, \mathbb{F}_q^n, 0)$, i.e. computes any isotropic vector for $\mathcal{P}_{\mathcal{L}}$, while the subsequent pseudo-oil vectors in O_0 must additionally satisfy the orthogonality property and be linear independent. In particular, the last pseudo-oil vector of O_0 must be a solution to the system of equations

$$\begin{cases} \mathcal{P}_{\mathcal{L}}(\mathbf{x}) = 0 \\ \mathcal{P}_{\mathcal{L}}(\mathbf{x}, \mathbf{o}_j) = 0 \quad 1 \leq j < m - k - B_1, \end{cases} \tag{8}$$

where the pseudo-oil vectors $\mathbf{o}_j$ for $1 \leq j < m-k-B_1$ are the already determined basis elements of O_0.

Proceeding with the first vector in O_1, one solves the system of equations

$$\begin{cases} \mathcal{P}_{\mathcal{B}_2}(\mathbf{x}) = 0 \\ \mathcal{P}_{\mathcal{B}_1}(\mathbf{x}, O_0) = 0, \end{cases} \tag{9}$$

and computes the subsequent ones by adding the orthogonality condition with respect to $\mathcal{B}_2$ that, by Lemma 2, ensures that O_1 is isotropic with respect to $\mathcal{P}_{\mathcal{B}_2}$.

After computing the pseudo-oil summands O_i for $i = 1, \ldots, p-1$, one computes a basis of $\tilde{O}$ by finding $b_p + k$ linear independent vectors solving the linear system of equations

$$\mathcal{P}_{\mathcal{B}_{i+1}}\left(\mathbf{x}, \bigoplus_{j=0}^{i} O_j\right) = 0 \quad i \in \{0, \ldots, p-1\}. \tag{10}$$

Existence of Pseudo-Oil Structures. Now we discuss the constraints to ensure the existence of the pseudo-oil decomposition described above for random generic MQ maps. As in Section 3.4, we start with the structural constraint

$$m - (k + B_1) \geq \ell, \tag{11}$$

coming from the Overdetermined step and ensuring that the first pseudo-oil summand O_0 has at least dimension ℓ. This requirement should be compared with the structural constraint of the original Hashimoto's algorithm, where $p = \ell = 1$. Next, we count the number of variables and equations of the equation systems solved during the Enriched Centrifugation phase to compute the pseudo-oil bases. It turns out that the most restrictive constraints are the ones relative to the last pseudo-oil vectors $\mathbf{o}_{m-k-B_1} \in O_0$ and $\mathbf{o}_m \in \tilde{O}$. As in Section 3.4, we write these constraints with the number of variables on the left side and the number of equations on the right side.

$$\begin{aligned} n - \big(m - B_1 - k - 1\big)(L+1) &> L \\ n - m + 1 &> \sum_{j=1}^{p} b_j (m - \sum_{i=1}^{j} b_i - k) \end{aligned} \tag{12}$$

Indeed, to compute $\mathbf{o}_{m-k-B_1}$, one looks for a non-zero solution in $\mathbf{x} \in \mathbb{F}_q^n/\langle \mathbf{o}_1, \ldots, \mathbf{o}_{m-k-B_1-1} \rangle$ to the system of equations Equation (8). After eliminating $(m-k-B_1-1)L$ variables using the orthogonality relations, the number of variables is $n-(m-k-B_1-1)(L+1)$, while the number of equations is L. This yields the first constraint in Equation (12).

To compute $\mathbf{o}_m \in \tilde{O}$, one looks for a nonzero vector solving the system of linear equations Equation (10) that is linear independent from the already determined $m-1$ pseudo-oil vectors. This is a linear system of $\sum_{j=1}^{p} b_j(m - \sum_{i=1}^{j} b_i - k)$ equations in $n-m+1$ variables, which yields the second constraint in Equation (12).

Remark 9. As for the original algorithm by Hashimoto, if one incurs in *local obstructions* at any step of the Enriched Centrifugation phase, it is possible to rerandomise the computation as described in Remark 2. We tested this approach with our proof of concept implementation: the Enriched Centrifugation phase succeeded for all tested MQ maps satisfying the inequalities Equation (12).

4.2 Enriched Extraction: Traditional Formulation

The traditional formulation of our Enriched Extraction is a natural extension of the one of Hashimoto's algorithm. We start by defining a linear change of variables $S \in \mathrm{GL}(\mathbb{F}_q)$ that sends the first m vectors of the ordered canonical basis of $\mathbb{F}_q^n$ to the ordered basis of O constructed during the Enriched Centrifugation phase. Then, we consider the transformed MQ map $\tilde{\mathcal{P}} = \mathcal{P} \circ S$, restricted to the subspace $\langle \mathbf{e}_1, \ldots, \mathbf{e}_m \rangle \subset \mathbb{F}_q^n$. In Figure 4, we depict the structure of the matrices of the transformed MQ map $\mathcal{P}$, which is a direct consequence of the isotropicity and orthogonality properties of the pseudo-oil decomposition. As for the original algorithm, we have Generic, OV and Reduced matrices, see Section 3.2.

Guess Step. The Guess step is identical to the one in the original algorithm by Hashimoto: we enumerate assignments for the last k variables

$$x_{m-k+1}, \ldots, x_m,$$

until all subsequent steps are successful.

Square Step. In the Square step of the Enriched Extraction phase, there are p square MQ problem to solve sequentially instead of just one, each of them corresponding to a block of Reduced matrices of Figure 4. More precisely, we first solve the first $B_p = b_p$ quadratic equations, which form an $\mathrm{MQ}(b_p, b_p)$ problem, thereby finding a set of assignments for the b_p variables

$$x_{m-b_p-k+1}, \ldots, x_{m-k}$$

that satisfy the first b_p equations. Indeed, the equations corresponding to the first b_p Reduced matrices only involve these variables, see Section 3.2. If this

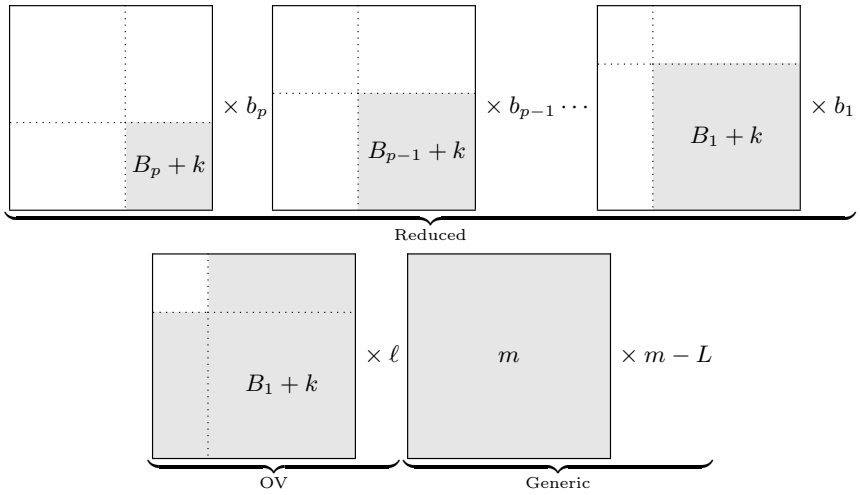

Fig. 4: Structure of the m-by-m matrices of transformed MQ map $\tilde{\mathcal{P}}$, restricted to $\langle \mathbf{e}_1, \ldots, \mathbf{e}_m \rangle$.

MQ problem does not admit a solution, we discard the current guess of the last k variables and proceed with the subsequent one.

Assume that the first MQ problem admits a solution. After choosing one assignment from the set of solutions, we move to the next block of $B_{p-1} - B_p = b_{p-1}$ equations in b_{p-1} variables, which form an MQ(b_{p-1}, b_{p-1}) problem. If this does not admit a solution, we backtrack to the previous square MQ problem and select an alternative assignment. If this MQ(b_{p-1}, b_{p-1}) problem admits a solution, we then compute its solution set of assignments.

We repeat this process to solve all square MQ(b_i, b_i) problems until we solve the last block of b_1 equations in b_1 variables, thereby computing its solution set. If any of these MQ problems does not admit a solution, we backtrack to the previous one and select an alternative assignment.

Overdetermined Step. In the Overdetermined step, we eliminate ℓ variables using the ℓ linear equations coming from the OV matrices in Figure 4. This leaves us with an overdetermined MQ$(m - k - L, m - L)$ problem. If it does not admit a solution, we backtrack to the last square MQ problem of the Square step. If it does admit a solution, we find an assignment for the remaining $m - k - L$ variables, which, together with the k variables assigned in the Guess step and the B_1 variables fixed in the Square step, gives us a full solution for the original MQ problem.

We summarise the process with the pseudocode in Algorithm 5. The correctness of the algorithm follows directly from Lemma 3, where now the linear change of variables S is the one associated with our Enriched pseudo-oil structure.

Algorithm 5: Traditional Enriched Extraction

Input: MQ(n, m) problem $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$; pseudo-oil decomposition of O; non-negative integers $k, \ell, b_1, \ldots, b_p$ such that $m - (k + B_1) \geq \ell$.

Output: Solution $\mathbf{s} \in \mathbb{F}_q^n$ to $(P, \mathbb{F}_q^n, \mathbf{t})$, or $\perp$.

Compute the change of variables $S \in \mathrm{GL}_n(\mathbb{F}_q)$ sending $\mathbf{e}_i$ to $\mathbf{o}_i$ for $i \in \{1, \ldots, m\}$.
$\tilde{\mathcal{P}} \leftarrow \mathcal{P} \circ S$ restricted to $\langle \mathbf{e}_i \rangle_{i=1}^m$.
for *all assignments* $\mathbf{k} \in \mathbb{F}_q^k$ *of the last* k *variables* `// Guess step`
do
  if *all the consecutive MQ*(b_i, b_i) *problems admit a solution* $\mathbf{o}_i$ `// Square step`
  then
    Eliminate ℓ variables using the ℓ linear equations.
    if *the overdetermined MQ*$(m - k - L, m - L)$ *problem admits a solution* $\mathbf{o}_0$ `// Overdetermined step`
    then
      Set $\mathbf{u} = (\mathbf{o}_0, \mathbf{o}_1, \ldots, \mathbf{o}_p, \mathbf{k}, \mathbf{0}^{n-m})^t$.
      return $\mathbf{s} = S\mathbf{u}$. `// Final solution`
  else
    Backtrack to the last square MQ problem admitting multiple solutions and select an alternative assignment.
return $\perp$.

4.3 Enriched Extraction: Geometric Formulation

As in Section 3.3, we abuse the notation by using $\mathcal{P}$ for the MQ map restricted to the linear subspace O.

Guess Step. The Guess step of the Enriched Extraction phase coincides exactly with the Guess step in the Extraction phase of Hashimoto's algorithm. In particular, we decompose the pseudo-oil summand $\tilde{O}$ as the direct sum of O_p of dimension b_p and G_k of dimension k, and we enumerate vectors $\mathbf{k} \in G_k$. Then, Algorithm 6 leverages the properties of the of pseudo-oil decomposition in order to progressively construct a preimage of the target $\mathbf{t}$.

First Square Step. Instead of solving only one MQ problem, the Enriched Extraction phase requires to solve p square MQ problems. The first one is the one with input

$$(\mathcal{P}_{\mathbf{k},\mathcal{B}_p}, O_p, \mathbf{t}_{\mathcal{B}_p}) \tag{13}$$

Recall that, according to our notation, we have defined the shifted and projected MQ map as

$$\begin{aligned} \mathcal{P}_{\mathbf{k},\mathcal{B}_p} : O_p &\to \pi_{\mathcal{B}_p}(\mathbb{F}_q^m) \\ \mathbf{x} &\mapsto \mathcal{P}_{\mathcal{B}_p}(\mathbf{x} + \mathbf{k}). \end{aligned}$$

This is a MQ(b_p, b_p) problem, since both domain and codomain have dimension b_p. If this problem admits no solution, we discard the current guess $\mathbf{k} \in G_k$ and proceed with the next one. Else, if there is a solution, we compute the solution set to Equation (13). Let us denote by $\mathbf{s}_p = \mathbf{o}_p + \mathbf{k}$, where $\mathbf{o}_p \in O_p$ is one solution to Equation (13). We call $\mathbf{s}_p$ a *partial solution*.

Additional Square Steps. For $i = 1, \ldots, p-1$, upon having solutions $\mathbf{o}_p, \ldots, \mathbf{o}_{p-i}$ for the first i MQ problems, let

$$\mathbf{s}_{p-i} = \mathbf{k} + \sum_{j=p-i}^{p} \mathbf{o}_j$$

be the partial solution. We now solve the MQ problem with input

$$(\mathcal{P}_{\mathbf{s}_{p-i}, \mathcal{B}_{p-i-1}}, O_{p-i-1}, \mathbf{t}_{\mathcal{B}_{p-i-1}}).$$

More explicitly, the MQ system one has to face is

$$\begin{cases} \mathcal{P}_{\mathcal{B}_{p-i}}(\mathbf{x} + \mathbf{s}_{p-i}) = \mathbf{t}_{\mathcal{B}_{p-i}} \\ \mathcal{P}_{\mathcal{B}_{p-i-1} \setminus \mathcal{B}_{p-i}}(\mathbf{x} + \mathbf{s}_{p-i}) = \mathbf{t}_{\mathcal{B}_{p-i-1} \setminus \mathcal{B}_{p-i}} \end{cases}$$

Since $\mathbf{x} \in O_{p-i-1}$ is isotropic for $\mathcal{P}_{\mathcal{B}_{p-i}}$ and $s_{p-i} \in \bigoplus_{j=p-i}^{p} O_j \oplus \tilde{O}$ is orthogonal to O_{p-i-1} with respect to $\mathcal{P}_{\mathcal{B}_{p-i+1}}$, the first B_{p-i} equations are fulfilled independently from the choice of $\mathbf{x} \in O_{p-i-1}$. Indeed, we have

$$\mathcal{P}_{\mathcal{B}_{p-i}}(\mathbf{x} + \mathbf{s}_{p-i}) = \underbrace{\mathcal{P}'_{\mathcal{B}_{p-i}}(\mathbf{x}, \mathbf{s}_{p-i})}_{0} + \underbrace{\mathcal{P}_{\mathcal{B}_{p-i}}(\mathbf{x})}_{0} + \mathcal{P}_{\mathcal{B}_{p-i}}(\mathbf{s}_{p-i}) = \mathbf{t}_{\mathcal{B}_{p-i}}.$$

Therefore, this is a MQ(b_{p-i-1}, b_{p-i-1}) problem because both the dimension of domain and the number of equations to be solved are $B_{p-i-1} - B_{p-i} = b_{p-i-1}$. If this problem admits no solution, we backtrack to the last square MQ problem admitting multiple solutions and select an alternative one. Else, we compute the solution set to this MQ problem and select one solution $\mathbf{o}_{p-i-1} \in O_{p-i-1}$, giving the new partial solution

$$\mathbf{s}_{p-i-1} = \mathbf{k} + \sum_{j=p-i-1}^{p} \mathbf{o}_j.$$

Overdetermined Step. At this point, we have the partial solution

$$s_1 = \mathbf{k} + \sum_{j=1}^{p} \mathbf{o}_j$$

that, by construction, fulfils all but the first $m - B_1$ equations of the original MQ problem $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$. In the Overdetermined step, we want to solve the MQ

problem with input $(\mathcal{P}_{\mathbf{s}_1}, O_0, \mathbf{t})$, which is essentially the original MQ problem on the restricted domain O_0 and shifted by $\mathbf{s}_1$. Explicitly, the MQ system of equations is

$$\begin{cases} \mathcal{P}_{\mathcal{B}_1}(\mathbf{x}+\mathbf{s}_1) = \mathbf{t}_{\mathcal{B}_1} \\ \mathcal{P}_{\mathcal{L}\setminus\mathcal{B}_1}(\mathbf{x}+\mathbf{s}_1) = \mathbf{t}_{\mathcal{L}\setminus\mathcal{B}_1} \\ \mathcal{P}_{[m]\setminus\mathcal{L}}(\mathbf{x}+\mathbf{s}_1) = \mathbf{t}_{[m]\setminus\mathcal{L}} \end{cases} \tag{14}$$

Again, by isotropicity and orthogonality of $\mathbf{x} \in O_0$, the first $\mathcal{B}_1$ equations are automatically satisfied. The intermediate set of $L - B_1 = \ell$ equations are all linear, because we can rewrite the left side as

$$\mathcal{P}_{\mathcal{L}\setminus\mathcal{B}_1}(\mathbf{x}+\mathbf{s}_1) = \underbrace{\mathcal{P}_{\mathcal{L}\setminus\mathcal{B}_1}(\mathbf{x})}_{0} + \underbrace{\mathcal{P}_{\mathcal{L}\setminus\mathcal{B}_1}(\mathbf{s}_1)}_{\text{constant}} + \underbrace{\mathcal{P}'_{\mathcal{L}\setminus\mathcal{B}_1}(\mathbf{x},\mathbf{s}_1)}_{\text{linear in } \mathbf{x}}$$

These ℓ linear relations can be used to eliminate ℓ out of the $\dim O_0 = m-k-B_1$ variables variables, leaving us with $m-k-B_1-\ell = m-k-L$ free variables and $m-L$ quadratic equations, which form an overdetermined $\mathrm{MQ}(m-k-L, m-L)$ problem.

Let us denote by $\mathbf{o}_0$ a solution to this MQ problem. We are now able to form the *full solution*

$$\mathbf{s} = \mathbf{k} + \sum_{j=0}^{p} \mathbf{o}_j,$$

which, by construction, is a preimage of $\mathbf{t}$ through $\mathcal{P}$, as it fulfils all equations in the system 14, and thus solves our initial $\mathrm{MQ}(n, m)$ problem.

We give the pseudocode of the geometric Enriched Extraction in Algorithm 6.

5 Complexity Analysis

In this section, we analyse the time complexity of our enriched algorithm. As for Hashimoto's original algorithm, one needs to solve an optimisation problem, where the constraints are the ones given in Equation (12) and the function to be optimised is the total runtime complexity of our algorithm. The latter is given by the sum of the runtime complexities of the Enriched Centrifugation phase and the Enriched Extraction phase. We now discuss the runtime complexities of the two phases individually.

5.1 Enriched Centrifugation phase

Following Remark 4, the complexity of the Enriched Centrifugation phase is

$$(m-k-B_1)\mathrm{GB}(L,L) + \sum_{i=1}^{p-1} b_i \mathrm{GB}\left(B_{i+1}, B_{i+1}\right). \tag{15}$$

Algorithm 6: Geometric Enriched Extraction

Input: Pseudo-oil decomposition as $O = \bigoplus_{i=0}^{p-1} O_i \oplus \tilde{O}$; MQ$(m, m)$ problem $(\mathcal{P}, O, \mathbf{t})$; non-negative integers $k, \ell, b_1, \ldots, b_p$ such that $k + L \leq m$.

Output: Solution to $(\mathcal{P}, \mathbb{F}_q^n, \mathbf{t})$, or $\perp$.

Decompose $\tilde{O}$ as $O_p \oplus G_k$ of dimensions b_p and k respectively.
for $\mathbf{k} \in V_k$ `// Guess`
do
  if *the all the square MQ problems of the Square step admit a solution* $\mathbf{o}_i$ `// Square`
  then
    $\mathbf{s}_1 = \mathbf{k} + \sum_{j=1}^{p} \mathbf{o}_j$
    if *the overdetermined MQ problem* $(\mathcal{P}_{\mathbf{s}_1}, O_0, \mathbf{t})$ *admits a solution* $\mathbf{o}_0 \in O_0$ `// Overdetermined`
    then
      return $\mathbf{s}_1 + \mathbf{o}_0$. `// Final Solution`
  else
    Backtrack to the last square MQ problem admitting multiple solutions and select an alternative one.
return $\perp$.

Indeed, we start by computing a basis for O_0. In particular, one solves $\dim O_0 = m - k - B_1$ problems of type MQ(L, L), which corresponds to the first summand in the complexity formula. Then, in order to compute a basis for O_i for $i = 1, \ldots, p-1$, one has to solve $\dim O_i = b_i$ times an MQ(B_{i+1}, B_{i+1}) problem, one for each i, which correspond to the second summand in the complexity formula. Finally, computing $\tilde{O}$ can be done in polynomial time and is thus negligible.

5.2 Enriched Extraction phase

The multiplicity of solutions to square MQ problems in the Square step gives rise to a solution tree, and Algorithms 5 and 6 essentially perform a Depth First Search (DFS) on this tree. A similar tree appeared under the name *MQ Tree* in the context of [MOR25], where each level represents a MQ$(1, 1)$ problem, i.e. one single equation in one variable only. We generalise the concept of an MQ Tree to MQ problems of arbitrary size.

Definition 1. *Given a sequence of square MQ problems stemming from a guessed vector* $\mathbf{k} \in G_k$ *as in the Square step of the Enriched Extraction phase, the corresponding* MQ Tree *is a rooted tree defined as follows:*

1. *The root corresponds to the initial guess* $\mathbf{k} \in G_k$ *of the Guess step.*
2. *For* $i \in \{0, \ldots, p-1\}$*, each node at depth* i *has a number of children equal to the number of solutions* $N \in \{0, \ldots, q^{b_{p-i}}\}$ *to the MQ*(b_{p-i}, b_{p-i}) *problem corresponding to the current solution branch. In particular, each child corresponds to one of these solutions.*

We now prove that each Square step of the Enriched Extraction phase produces in expectation one partial solution $\mathbf{s}_1$ to be used in the subsequent Overdetermined step.

Lemma 4. *For $j \in \{1, \ldots, p\}$, let N_j be the number of nodes at depth j of the MQ Tree corresponding to a guessed vector $\mathbf{k} \in G_k$. Under Heuristic 1, we have $\mathbb{E}[N_j] = 1$ for every $j \in \{1, \ldots, p\}$.*

Proof. Under Heuristic 1, all the square MQ problems of the Square step are uniformly random after fixing the previous partial solution. Thus, each of these has one solution in expectation. In particular, it follows that $\mathbb{E}[N_1] = 1$. By the law of total expectation, we have

$$\mathbb{E}[N_j] = \mathbb{E}[\mathbb{E}[N_j | N_{j-1}]] = \mathbb{E}[\mathbb{E}[N_1] N_{j-1}] = \mathbb{E}[N_{j-1}].$$

By induction, we conclude that $\mathbb{E}[N_j] = 1$ for every $j \in \{1, \ldots, p\}$. □

Remark 10. Lemma 4 implies that in expectation, for each guess $\mathbf{k} \in G_k$, one solves exactly p square MQ problems, one for each depth of the MQ Tree. Intuitively, this average case analysis accounts for both shallow and extremely dense MQ Trees, which eventually balance out in expectation.

Theorem 2. *Let $k, \ell, b_1, \ldots, b_p$ be positive integers satisfying Equations* (11) *and* (12)*. Under Heuristic 1, the Enriched Extraction phase produces one full solution in expectation in time*

$$q^k \left(GB(m - k - L, m - L) + \sum_{j=1}^{p} GB(b_j, b_j) \right). \tag{16}$$

Proof. The proof is identical to the one of Theorem 1, since the Square step of the Enriched Extraction phase produces in expectation one partial solution $\mathbf{s}_1$ to be used in the subsequent Overdetermined step, which succeeds with probability q^{-k}. Under Heuristic 1, every new guess $\mathbf{k} \in G_k$ in the Guess step produces a uniformly random overdetermined MQ problem to be solved in the Overdetermined step. Therefore, after q^k iterations of the Guess step (and the subsequent steps), one finds one full solution to the original MQ(n, m) problem in expectation. □

We have implemented a proof of concept of the algorithm in SageMath, which can be found in the git repository `https://github.com/maxostuzzi/MQ_chemistry`. The repository also contains a script that generates toy examples, showing all the steps and printing the corresponding MQ tree.

6 Concrete Complexity Estimates

The bit complexity estimates are computed solving a discrete optimisation problem under the constraints in Equations (11) and (12), using a dynamic programming approach that can be found in our git repository . The script computes the

bit complexity of the Enriched Centrifugation phase and the Enriched Extraction phase separately according Equations (15) and (16), and returns the $\log_2$ of their sum.

To evaluate the algebraic solving complexities in Equations (15) and (16), the script internally uses the estimator [EVZB24] with default settings. The estimator returns the bit complexity of solving a $\mathrm{MQ}(n, m)$ problem for various algorithms, and we select the minimal one. We also provide a script that on input the parameters k, ℓ, b_i of our algorithm, checks the constraints in Equations (11) and (12) and outputs the corresponding bit complexity.

Our strategy is most effective when the number of variables is large, because this provides more leeway to fit richer pseudo-oil structures. To quantify this, we use the *u-rate* introduced in [MOR25] and defined as

$$\rho = \frac{2n}{m(m+1)}.$$

The u-rate measures the proximity of an MQ instance to the polynomial time regime of [MHT13]: by definition, instances with $\rho \geq 1$ fall in the heavily underdetermined territory and become efficiently solvable in polynomial time.

The u-rate is not, however, the only factor determining the performance of our algorithm. In fact, the size q of the underlying finite field plays a significant role: smaller fields allow to increase k for a more aggressive Guess step, artificially creating more room for richer pseudo-oil structures. In Figure 5, we show the estimates for $m = 100$ and three different field sizes $q \in \{4, 7, 16\}$. Our algorithm outperforms Hashimoto's starting from u-rate of roughly $0.03, 0.09$ and 0.25 for $q = 4, 7$ and 16 respectively.

Interestingly, our algorithm converges back to Hashimoto's for approximately $\rho \geq 0.51$, i.e. $n \geq 2600$, independently from the field size: in this regime, parameter choices that would reduce the Enriched Extraction phase complexity are still available, but only at the cost of a more expensive Enriched Centrifugation phase, which would dominate the overall complexity. In light of Remark 4, one could refine the dimension count for the MQ problems arising in the Centrifugation phase, thereby reducing its estimated complexity. We do not pursue this further, since the regime in question has not yet appeared in cryptographic applications.

Another critical parameter for GB-based algorithms is the memory footprint. As our algorithm needs to solve smaller instances of the MQ problem, we are able to achieve increasingly larger improvements by penalising heavier memory usage, which is arguably even more realistic than a memory-free comparison. The desired memory access model can be selected directly from our optimisation algorithm. The calculation is standard: assume that the time complexity of the algorithm is T and the memory complexity is M. Let f be a function that takes as input the memory complexity M and returns the cost $f(M)$ of a single memory access under the chosen model. Then, the bit complexity of the algorithm is computed as $\log_2(T \cdot f(M))$. We will use $f(M) = 1, \log_2(M), \sqrt[3]{M}, \sqrt{M}$ for constant, logarithmic, cube-root and square-root memory access models, respectively.

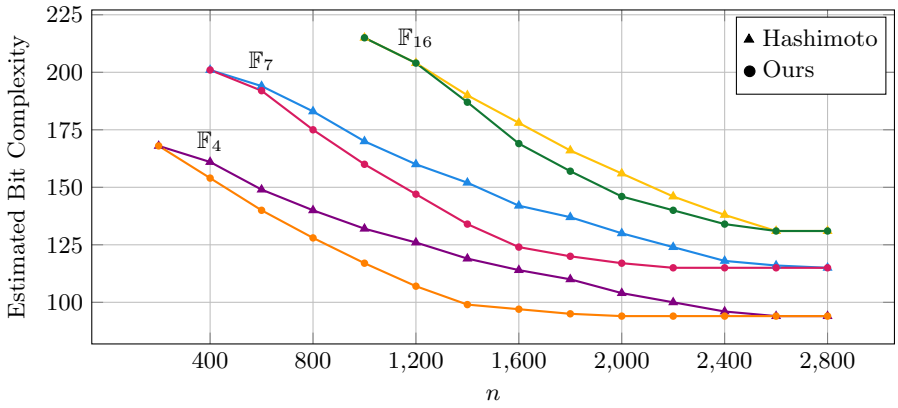

Fig. 5: Estimates for $m = 100$, increasing n and $q \in \{4, 7, 16\}$.

6.1 Attacking MQ-based Signature Schemes

In what follows, we apply our enriched algorithm to attack two of the MQ-based signature schemes submitted to the third round of the NIST competition: MAYO [BCC+24] and QR-UOV [FIH+24]. For the other two submissions, UOV [BCD+24] and SNOVA [WCD+24], the proposed parameter sets yield MQ problems with small u-rates and not small enough field sizes, so our enriched algorithm fails to improve on the state of the art. We therefore do not report complexity estimates for these two schemes. The situation is different for MAYO and QR-UOV, whose parameter sets feature significantly larger u-rates and smaller field sizes.

MAYO. We summarise the impact of our algorithm in the constant memory access model in Table 2. With our algorithm, we are able to reduce the bit complexity of forging a MAYO signature for the Security Level I parameter set by a factor of 2^7. The precise input parameters for the improved attack are

$$(k, \ell, b_1, \ldots, b_8) = (32, 14, 3, 3, 3, 3, 3, 3, 3, 3).$$

More precisely, the Enriched Centrifugation phase has bit complexity 103, which is not the dominant term. In the Enriched Extraction phase, for each of the 2^{128} guesses we solve in expectation eight MQ$(3, 3)$ square problems and one MQ$(8, 40)$ overdetermined problem, having bit complexities roughly 16 and 18 respectively. [4] Lastly, the polynomial factors of the Enriched Extraction phase yield a total of roughly 15 bits per guess. Therefore, the combination of the Guess step and the Overdetermined step is dominating the total bit complexity, and summing all these contributions yields

$$2^{128} \cdot (2^{15} + 2^{16} + 2^{18}) \approx 2^{146},$$

[4] These data can be reproduced using our scripts in the git repository `https://github.com/maxostuzzi/MQ_chemistry`.

as reported in Table 2.

For Security Levels III and V, our optimisation algorithm shows that the shortage of variables, indicated by the smaller u-rates, does not allow to take full advantage of our enriched algorithm. Indeed, the original algorithm ($p, \ell = 1$) remains the one with the best performance in the constant memory access model.

Table 2: Comparison for forging a MAYO signature in the constant memory access model.

Security	(n, m, q)	ρ	p	ℓ	Ours	Hashimoto
Level I	$(860, 78, 16)$	0.28	8	14	**146**	153
Level III	$(1298, 108, 16)$	0.22	1	1	223	223
Level V	$(1848, 142, 16)$	0.18	1	1	295	295

Considering memory access models that penalise heavy memory usage, we obtain speed-ups up to a factor of 2^{18} for the Security Level I parameters of MAYO and up to 2^9 for Security Level III, see Table 3. As in the constant model, the shortage of variables in Security Level V prevents our algorithm from improving over the state of the art.

Table 3: Comparison of the attack on MAYO SL1 under other memory access models.

	Level I		Level III		Level V	
Memory Model	Ours	Hashimoto	Ours	Hashimoto	Ours	Hashimoto
Logarithmic	**148**	157	226	226	299	299
Cube-Root	**149**	163	**229**	235	319	319
Square-Root	**151**	169	**231**	240	325	325

QR-UOV. Forging a QR-UOV [FIKT21,FIH$^+$24] signature can be reduced to solving an instance of the underdetermined MQ problem. Despite having u-rates smaller than MAYO, the smaller field size of some parameter sets allows to increase the parameter k, which in turn guarantees more leeway for choosing p and ℓ. We summarise the impact of our algorithm in the constant model in Table 4.

Most notably, we are able to reduce the complexity of the *direct attack* by a factor from 2^8 for Security Level I in the constant model, see Table 4. The precise input parameters for the improved attack are

$$(k, \ell, b_1, \ldots, b_5) = (54, 12, 4, 4, 5, 5, 5).$$

Table 4: Comparison for forging a QR-UOV signature.

Security	(n, m, q)	ρ	p	ℓ	Ours	Hashimoto
Level I	$(840, 100, 7)$	0.17	5	13	**172**	180
Level III	$(1240, 140, 7)$	0.13	1	1	**257**	260
Level V	$(1680, 190, 7)$	0.09	1	1	351	351

In the other memory models, our algorithm outperforms the original one, with a reduction of the bit complexity up to a factor of $2^{16}, 2^{15}$ and 2^4 for Security Level I, III and V respectively, see Table 5.

Table 5: Comparison of the attack on QR-UOV SL1 under other memory access models.

	Level I		Level III		Level V	
Memory Model	Ours	Hashimoto	Ours	Hashimoto	Ours	Hashimoto
Logarithmic	**172**	185	**259**	264	356	356
Cube-Root	**174**	188	**260**	272	373	373
Square-Root	**176**	192	**261**	276	**379**	383

7 Conclusions

In this paper, we introduced pseudo-oil subspaces as a geometric framework for understanding underdetermined MQ solving algorithms. We used the framework to enrich Hashimoto's algorithm and achieve concrete improvements for forging MAYO and QR-UOV signatures. This bit security reduction does not immediately threaten the security of MAYO or QR-UOV, whose designers targeted comfortable security margins.

The pseudo-oil framework opens several directions for future work. On the algorithmic side, it is natural to ask whether even richer or structurally different pseudo-oil decompositions could yield further concrete improvements for generic MQ maps. Our stratified construction is natural, but there is certainly room for different design choices.

Another open problem is understanding whether the structure of the MQ maps arising from MAYO and QR-UOV can be exploited to push the parameters ℓ and b_i beyond the limits imposed by Equation (12).

References

BCC+24. Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias J. Kannwischer. MAYO. Technical report, National Institute of Standards and

Technology, 2024. available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures`.

BCD+24. Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. UOV — Unbalanced Oil and Vinegar. Technical report, National Institute of Standards and Technology, 2024. available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures`.

Beu22. Ward Beullens. MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *SAC 2021*, volume 13203 of *LNCS*, pages 355–376. Springer, Cham, September / October 2022.

BFP09. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.

BFP12. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, ISSAC '12, page 67–74, New York, NY, USA, 2012. Association for Computing Machinery.

BG94. Mihir Bellare and Shafi Goldwasser. The complexity of decision versus search. *SIAM Journal on Computing*, 23(1):97–119, 1994.

CGMT02. Nicolas Courtois, Louis Goubin, Willi Meier, and Jean-Daniel Tacier. Solving underdefined systems of multivariate quadratic equations. In David Naccache and Pascal Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 211–227. Springer, Berlin, Heidelberg, February 2002.

CHMT14. Chen-Mou Cheng, Yasufumi Hashimoto, Hiroyuki Miura, and Tsuyoshi Takagi. A polynomial-time algorithm for solving a class of underdetermined multivariate quadratic equations over fields of odd characteristics. In Michele Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*, pages 40–58. Springer, Cham, October 2014.

CKPS00. Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, Berlin, Heidelberg, May 2000.

EVZB24. Andre Esser, Javier A. Verbel, Floyd Zweydinger, and Emanuele Bellini. {SoK: CryptographicEstimators - a Software Library for Cryptographic Hardness Estimation}. In *AsiaCCS*. {ACM}, 2024.

FIH+24. Hiroki Furue, Yasuhiko Ikematsu, Fumitaka Hoshino, Tsuyoshi Takagi, Kan Yasuda, Toshiyuki Miyazawa, Tsunekazu Saito, and Akira Nagai. QR-UOV. Technical report, National Institute of Standards and Technology, 2024. available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures`.

FIKT21. Hiroki Furue, Yasuhiko Ikematsu, Yutaro Kiyomura, and Tsuyoshi Takagi. A new variant of unbalanced Oil and Vinegar using quotient ring: QR-UOV. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 187–217. Springer, Cham, December 2021.

FNT21. Hiroki Furue, Shuhei Nakamura, and Tsuyoshi Takagi. Improving Thomae-Wolf algorithm for solving underdetermined multivariate quadratic poly-

nomial problem. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*, pages 65–78. Springer, Cham, 2021.

GJ79. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

Gro98. Lov K. Grover. A framework for fast quantum mechanical algorithms. In *30th ACM STOC*, pages 53–62. ACM Press, May 1998.

Has21. Yasufumi Hashimoto. An improvement of algorithms to solve under-defined systems of multivariate quadratic equations. Cryptology ePrint Archive, Paper 2021/1045, 2021.

HBD+20. Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS+. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

KPG99. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 206–222. Springer, Berlin, Heidelberg, May 1999.

LDK+20. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

MHT13. Hiroyuki Miura, Yasufumi Hashimoto, and Tsuyoshi Takagi. Extended algorithm for solving underdefined multivariate quadratic equations. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 118–135. Springer, Berlin, Heidelberg, June 2013.

MOR25. Alexander May, Massimo Ostuzzi, and Henrik Ressler. Just guess: Improved (quantum) algorithm for the underdetermined MQ problem. Cryptology ePrint Archive, Paper 2025/1788, 2025.

MS17. Rusydi H Makarim and Marc Stevens. M4gb: an efficient gröbner-basis algorithm. In *Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation*, pages 293–300, 2017.

Pat97. Jacques Patarin. The oil and vinegar signature scheme. In *Presented at the Dagstuhl Workshop on Cryptography September 1997*, 1997.

PFH+20. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

SAB+20. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute

of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

TW12. Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 156–171. Springer, Berlin, Heidelberg, May 2012.

WCD+24. Lih-Chung Wang, Chun-Yen Chou, Jintai Ding, Yen-Liang Kuan, Ming-Siou Li, Bo-Shu Tseng, Po-En Tseng, and Chia-Chun Wang. SNOVA. Technical report, National Institute of Standards and Technology, 2024. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.