mathematics

MDPI

Article

On the Structural Solvability of MLWE with Rank-Deficient Public Matrices

Nor Siti Khadijah Arunah [1,2], Amir Hamzah Abd Ghafar [1,3,4]*, Muhammad Asyraf Asbullah [1,3,5] and Muhammad Rezal Kamel Ariffin [1,3,4]

1 Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia; norsi830@uitm.edu.my (N.S.K.A.); ma_asyraf@upm.edu.my (M.A.A.); rezal@upm.edu.my (M.R.K.A.)
2 Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Cawangan Johor Kampus Segamat, Segamat 85000, Johor, Malaysia
3 Malaysia Cryptology Technology and Management Centre (PTPKM), c/o Universiti Putra Malaysia, UPM Serdang, Serdang 43400, Selangor, Malaysia
4 Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia
5 Centre for Foundation Studies in Science, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia
* Correspondence: amir_hamzah@upm.edu.my

Abstract

The security of Module Learning With Errors (MLWE) relies on the assumption that the public matrix is sampled uniformly and forms a full-rank operator. In this work, we examine the structural consequences of relaxing this assumption by considering public matrices that demonstrate slot-wise rank deficiency under the Number Theoretic Transform (NTT). Focusing on the case where each NTT slot matrix has rank 1, we show that this leads to enlarged left nullspace, which allows the elimination of the secret component s_1, reducing the original relation to a linear system consisting only of s_2. Given partial knowledge of s_2, this projected system admits a unique solution once a sufficient number of independent constraints is available. After recovering s_2, the problem of determining s_1 reduces to solving a bounded linear system, which can be viewed as a structured instance of the Short Integer Solution (SIS) problem. These results provide a dimension-based characterization of solvability under slot-wise rank-deficient public matrices. Using ML-DSA as a concrete instantiation, we illustrate how such structural deviations affect the behavior of the system and discuss simple safeguards, such as rank verification during key generation, to mitigate these issues.

Keywords: Module-LWE; lattice-based cryptography; CRYSTALS-Dilithium; ML-DSA; rank deficiency; structural analysis; post-quantum cryptography

MSC: 94A60; 11T71; 68P25

Academic Editor: Jonathan Blackledge

Received: 15 April 2026
Revised: 8 May 2026
Accepted: 13 May 2026
Published: 19 May 2026

1. Introduction

The Learning With Errors (LWE) problem is a central hardness assumption in lattice-based post-quantum cryptography (PQC). It was introduced by Regev in 2009 [1], extended to Ring-LWE by Lyubashevsky et al. in 2010 [2], and later generalized to the module setting by Langlois and Stehlé in 2015 [3]. In the Module-LWE (MLWE) formulation, matrices and secret vectors are defined over the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, yielding structured linear relations of the form

$$\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{t},$$

where $\mathbf{A} \in R_q^{k \times \ell}$ is public and $\mathbf{s}_1, \mathbf{s}_2$ are small-norm secret vectors. This algebraic structure underlies modern lattice-based schemes such as Saber [4], CRYSTALS-Kyber [5], and CRYSTALS-Dilithium [6].

The security of MLWE-based constructions relies on the assumption that the public matrix $\mathbf{A}$ is sampled uniformly at random from $R_q^{k \times \ell}$ [5,7,8]. In this setting, the public matrix $\mathbf{A}$ is generated with overwhelming probability as a full-rank operator. As a result, nullspace projections cannot be used to separate the secret components. Under these standard assumptions, recovering $\mathbf{s}_1$ or $\mathbf{s}_2$ from a single instance reduces to solving a bounded linear system related to the Short Integer Solution (SIS) problem, which is believed to be computationally hard [8,9]. Neither the CRYSTALS-Dilithium specification [6] nor the ML-DSA standard [10] includes an explicit rank verification step during key generation.

The situation changes when the rank profile of $\mathbf{A}$ deviates from this ideal setting. If $\mathbf{A}$ has slot-wise rank deficiency under the NTT decomposition, its nullspace becomes larger and creates an additional algebraic structure. In this case, projection operators can be used to eliminate the $\mathbf{s}_1$ component, turning the original relation into a linear system in the coefficients of $\mathbf{s}_2$. From a linear-algebraic point of view, whether this system can be solved depends on the balance between the number of independent constraints and the dimension of the unknown space.

In this work, we analyze how MLWE behaves under slot-wise rank-1 public matrices in the NTT domain and study when recovery is possible. When each NTT slot matrix has rank 1, the enlarged left nullspace permits elimination of the $\mathbf{s}_1$ component. In this setting, partial knowledge of $\mathbf{s}_2$ already becomes useful, and unique recovery is possible once enough independent projected equations are available. After recovering $\mathbf{s}_2$, the remaining task is to determine $\mathbf{s}_1$, which reduces to solving a linear system of dimension $(k-1)n$. This corresponds to a structured SIS-type search problem in the single-instance setting. We also provide explicit recovery procedures and a corresponding complexity analysis, using ML-DSA as a concrete instantiation to illustrate the resulting algebraic behavior. Overall, these results emphasize the importance of verifying the rank of the public matrix $\mathbf{A}$ during key generation in cryptographic standards.

2. Related Works

Research on MLWE-based constructions has largely focused on several areas, including hardness reductions, concrete security evaluation, and parameter selection for post-quantum cryptography. Recent work has strengthened the theoretical foundations of LWE and MLWE through refined worst-case-to-average-case reductions [11], structural hardness analyses [12,13], and studies under more general leakage or distribution models [14]. In most of these works, the public matrix is assumed to be sampled uniformly at random, and the analysis is carried out within this model.

Complementary work has looked at implementation-level threats, such as side-channel and fault-injection attacks on CRYSTALS-Dilithium [15–17]. These research approaches exploit auxiliary leakage or induced faults to reveal partial information about secret-dependent values. However, they still assume honest parameter generation and therefore do not change the algebraic structure of the public matrix.

Beyond this standard setting, a number of works have considered structural deviations in LWE-type systems. In particular, adversarially generated public parameters [18] show that departures from the usual sampling procedure can introduce an exploitable algebraic structure. Related studies on structured parameter constructions [19,20] further illustrate how carefully designed matrices can affect computational behavior, even when

they appear as pseudorandom. In contrast, the setting considered in this work does not rely on embedded trapdoors or hidden bases, but instead examines how slot-wise rank deficiency in the NTT domain affects the algebraic structure of the MLWE system.

Rank-related phenomena also play an important role in algebraic cryptanalysis. In multivariate cryptography and MinRank-based attacks [21,22], rank deficiencies reduce the effective solution space and make dimension-driven attacks possible. These results illustrate how the rank profile of a matrix can directly influence the solvability of structured algebraic systems.

When the solution space becomes smaller, efficient search techniques are expected to become more relevant. Meet-in-the-middle (MiTM) methods are a common approach for reducing the complexity of bounded search problems by splitting the search space into smaller parts. Such techniques have been used in several lattice-based cryptanalytic settings, including hybrid attacks that combine lattice reduction with combinatorial search in NTRU-type and related systems [23–25]. In this work, MiTM is used as a supporting tool to explore the bounded solution space that arises after eliminating $\mathbf{s}_2$.

Finally, the remainder of this paper is organized as follows. Section 3 reviews the MLWE framework, the Number Theoretic Transform (NTT), and the CRYSTALS-Dilithium setting. Section 4 formalizes the adversarial parameter authority model. Section 5 analyzes slot-wise rank-1 public matrices and presents the recovery of $\mathbf{s}_2$, together with complexity analysis and a numerical illustration. Section 6 examines the resulting linear structure of $\mathbf{s}_1$ and studies recovery using meet-in-the-middle techniques. Section 7 discusses simple safeguards for detecting and preventing slot-wise low-rank public matrices during key generation, and Section 8 concludes this paper.

3. Preliminaries

3.1. Module Learning with Errors (MLWE) Framework

Let $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ be the polynomial ring with modulus q. The MLWE problem is defined over modules of the form $R_q^{k\times\ell}$. A public matrix $\mathbf{A} \in R_q^{k\times\ell}$ is sampled according to the prescribed distribution, and secret vectors $\mathbf{s}_1 \in R_q^{\ell}$ and $\mathbf{s}_2 \in R_q^{k}$ are drawn from bounded small-coefficient distributions.

The public relation is given by

$$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \pmod{q}. \tag{1}$$

The MLWE hardness assumption states that, for appropriate parameters, recovering $\mathbf{s}_1$ (or distinguishing $(\mathbf{A}, \mathbf{t})$ from uniform) is computationally infeasible when $\mathbf{A}$ is sampled according to the standard distribution.

3.2. Number Theoretic Transform (NTT)

Let $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, where q is chosen such that a primitive $2n$-th root of unity exists modulo q. The Number Theoretic Transform (NTT) provides a ring isomorphism

$$R_q \cong \mathbb{Z}_q^n,$$

under which polynomial multiplication corresponds to component-wise multiplication.

Applying the NTT entry-wise to Equation (1) decomposes the MLWE relation into a collection of n linear systems:

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} + \mathbf{s}_2^{(i)} = \mathbf{t}^{(i)}, \quad i = 0, \ldots, n-1,$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{k\times\ell}$ denotes the i-th NTT component.

3.3. CRYSTALS-Dilithium as an Instantiation

CRYSTALS-Dilithium instantiates the MLWE relation

$$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \pmod{q},$$

where $\mathbf{A} \in R_q^{k\times\ell}$ is deterministically generated from a public seed, and $\mathbf{s}_1, \mathbf{s}_2$ are sampled from bounded small-coefficient distributions.

For illustrative purposes, our simulations use reduced parameters $q = 769$ and $n = 16$ instead of the standard values $8,380,417$ and 256, respectively. Since the structural conditions derived in this work depend only on linear dimension counts that scale proportionally with n, these reduced-parameter experiments preserve the structural solvability behavior analyzed in this study.

During signature generation, the long-term secret vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are reused across multiple operations. From a security perspective, recovery of $\mathbf{s}_1$ compromises the signing key, while knowledge of $\mathbf{s}_2$ alone does not necessarily break the scheme but may simplify subsequent algebraic analysis. In this work, we study the setting in which structural properties of the public matrix allow $\mathbf{s}_2$ to be recovered and examine how this affects the remaining problem of determining $\mathbf{s}_1$.

To formalize the structural setting considered in this work, we first describe the decomposition of the MLWE public matrix under the NTT and introduce the notion of NTT slot matrices.

Definition 1 (NTT slot matrices)**.** *Let $\mathbf{A} \in R_q^{k\times\ell}$ and apply the NTT to each polynomial entry of $\mathbf{A}$. For each index $i = 0, \ldots, n-1$, we denote by*

$$\mathbf{A}^{(i)} \in \mathbb{Z}_q^{k\times\ell}$$

*the matrix formed by the i-th NTT coefficients of the polynomial entries of $\mathbf{A}$. We refer to $\mathbf{A}^{(i)}$ as the i-*th NTT slot matrix *of $\mathbf{A}$.*

Under this decomposition, the MLWE relation

$$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$$

becomes a collection of n linear systems

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} + \mathbf{s}_2^{(i)} = \mathbf{t}^{(i)} \pmod{q}, \quad i = 0, \ldots, n-1.$$

We define the *slot-wise rank* of $\mathbf{A}$ as the rank of the NTT slot matrices $\mathbf{A}^{(i)}$ over $\mathbb{Z}_q$. We say that $\mathbf{A}$ is *slot-wise rank-r* if

$$\operatorname{rank}(\mathbf{A}^{(i)}) = r \quad \text{for all } i = 0, \ldots, n-1.$$

In our analysis, recovery of $\mathbf{s}_2$ relies on partial knowledge of its coefficients. The following definition formalizes the notion of coefficient exposure used throughout this work.

Definition 2 (Coefficient exposure of $\mathbf{s}_2$)**.** *Let $\mathbf{s}_2 \in R_q^k$, where each entry is a polynomial of degree at most $n-1$. We say that one polynomial component of $\mathbf{s}_2$ is fully known if n independent coefficients of $\mathbf{s}_2$ are revealed. The revealed coefficients may correspond to all coefficients of a single polynomial entry or to any collection of coefficients across multiple entries whose total number equals n.*

Since the derived structural conditions depend only on linear dimension counts that scale proportionally with n, the same solvability behavior holds at practical parameter scales. Reduced-parameter instances are used solely for numerical illustration.

4. Adversarial Parameter Authority Model

In the standard ML-DSA specification, the public matrix $\mathbf{A} \in R_q^{k \times \ell}$ is deterministically derived from a uniformly sampled seed via an XOF-based expansion procedure. Security analyses assume that this process produces a matrix that is computationally indistinguishable from uniform sampling.

We consider a structural deviation model in which the distributed matrix $\mathbf{A}$ deviates from this prescribed generation process and exhibits controlled slot-wise rank deficiency under the NTT decomposition.

Definition 3 (Adversarial Parameter Generation Model)**.** *An authority responsible for parameter distribution may publish a matrix* $\mathbf{A}$ *that appears syntactically valid under the key generation specification but exhibits controlled slot-wise rank deficiency under the NTT decomposition.*

Since neither the CRYSTALS-Dilithium specification nor the ML-DSA standard [6,10] includes an explicit rank verification step during key generation, such matrices may not be detected by the standard generation process. Our analysis therefore characterizes MLWE solvability under adversarially chosen rank profiles, without contradicting hardness guarantees under honest parameter generation.

4.1. Malicious Generation of the Public Matrix via `ExpandA`

To illustrate how slot-wise rank-deficient public matrices may arise under an adversarial parameter generation process, we consider a malicious implementation of the `ExpandA` procedure in the NTT domain.

In particular, for each slot $i = 0, \ldots, n-1$, we construct the matrix $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{k \times \ell}$ as an outer product of two vectors,

$$\mathbf{A}^{(i)} = \mathbf{u}^{(i)} (\mathbf{v}^{(i)})^{\mathsf{T}},$$

where $\mathbf{u}^{(i)} \in \mathbb{Z}_q^k$ and $\mathbf{v}^{(i)} \in \mathbb{Z}_q^\ell$.

Because $\mathbf{A}^{(i)}$ is expressed as the product of a column vector and a row vector, every row of $\mathbf{A}^{(i)}$ is a scalar multiple of $(\mathbf{v}^{(i)})^{\mathsf{T}}$. Hence,

$$\operatorname{rank}(\mathbf{A}^{(i)}) = 1 \quad \text{for all } i.$$

This structured construction does not reflect the standard parameter generation process in ML-DSA, where entries are sampled to approximate uniform randomness. Instead, it serves as a deterministic framework for analyzing the algebraic consequences of slot-wise rank deficiency in MLWE systems.

From an adversarial perspective, subtle structural modifications are often preferable to highly disruptive parameter manipulations, since they are less likely to trigger implementation-level suspicion. In particular, a modified `ExpandA` procedure that preserves the overall deterministic generation flow while introducing hidden slot-wise rank deficiencies may appear syntactically valid under standard recomputation checks. Although such a setting falls outside the intended ML-DSA deployment model, it provides a useful framework for studying how unverified structural properties can influence the algebraic solvability of MLWE systems.

4.2. Leakage Model for $\mathbf{s}_2$

The recovery of $\mathbf{s}_2$ in our analysis assumes partial knowledge of its coefficients. In particular, we consider the case where one polynomial component of $\mathbf{s}_2$ is fully known, corresponding to n revealed coefficients.

Such partial exposure does not arise in the standard threat model of ML-DSA. However, it is well established that implementation-level attacks may leak partial information about secret-dependent values. In particular, side-channel attacks (e.g., power analysis or timing leakage), fault-injection attacks, and memory exposure scenarios have been shown to reveal subsets of secret coefficients in lattice-based cryptographic implementations.

In this work, we do not assume full disclosure of $\mathbf{s}_2$, but rather partial leakage of one component. This level of exposure is sufficient to illustrate how structural properties of the public matrix $\mathbf{A}$ interact with leaked information to affect the solvability of the underlying MLWE system.

Therefore, our analysis should be interpreted as a structural study under combined conditions of non-ideal parameter generation and partial leakage, rather than a direct attack on standard ML-DSA deployments.

5. Structural Analysis Under Slot-Wise Rank-Deficient Public Matrices

Assume that, under the NTT decomposition,

$$\operatorname{rank}(\mathbf{A}^{(i)}) = 1 \quad \forall i = 0, \ldots, n-1.$$

Each slot, therefore, admits a left nullspace of dimension 3.

5.1. Slot-Wise Rank-1 Case (4×4)

Under the NTT decomposition, the MLWE relation becomes

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} + \mathbf{s}_2^{(i)} = \mathbf{t}^{(i)} \pmod{q}, \quad i = 0, \ldots, n-1.$$

Since $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{4\times 4}$ has rank 1, its left nullspace has dimension

$$\dim\left(\mathcal{N}_L(\mathbf{A}^{(i)})\right) = 4 - 1 = 3.$$

Consequently, for each slot i, there exists a nonzero matrix $\mathbf{U}^{(i)} \in \mathbb{Z}_q^{4\times 4}$ whose rows lie in the left nullspace of $\mathbf{A}^{(i)}$, and hence satisfy

$$\mathbf{U}^{(i)}\mathbf{A}^{(i)} = \mathbf{0}.$$

5.2. Projection and Elimination of $\mathbf{s}_1$

Multiplying the slot-wise MLWE relation by $\mathbf{U}^{(i)}$ and using $\mathbf{U}^{(i)}\mathbf{A}^{(i)} = \mathbf{0}$ yields

$$\mathbf{U}^{(i)}\mathbf{s}_2^{(i)} = \mathbf{U}^{(i)}\mathbf{t}^{(i)} \pmod{q}, \qquad i = 0, \ldots, n-1. \tag{2}$$

Therefore, the influence of $\mathbf{s}_1^{(i)}$ is eliminated in every slot, and the projected equations involve only $\mathbf{s}_2^{(i)}$ and public data.

5.3. Structural Conditions for $\mathbf{s}_2$ *Recovery Under Rank-Deficient MLWE*

Theorem 1 (Recovery of $\mathbf{s}_2$ under slot-wise rank-1 public matrices)**.** *Let*

$$\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \pmod{q}, \qquad \mathbf{A} \in R_q^{4\times 4},$$

and suppose that

$$\operatorname{rank}(\mathbf{A}^{(i)}) = 1 \quad \textit{for all } i = 0, \ldots, n-1$$

under the NTT decomposition.

Assume that one polynomial component of $\mathbf{s}_2$ *is fully known. Let*

$$\mathbf{P}\mathbf{y} = \mathbf{x} \pmod{q}$$

be the global projected system obtained from the left nullspace projection

$$\mathbf{U}^{(i)}\mathbf{A}^{(i)} = \mathbf{0},$$

where $\mathbf{U}^{(i)} \in \mathbb{Z}_q^{3\times 4}$ *has rows forming a basis of the left nullspace of* $\mathbf{A}^{(i)}$.

Then $\mathbf{s}_2$ *is uniquely recoverable if and only if*

$$\operatorname{rank}(\mathbf{P}) = 3n.$$

Proof. Applying the NTT entry-wise to the MLWE relation gives, for each slot $i = 0, \ldots, n-1$,

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} + \mathbf{s}_2^{(i)} = \mathbf{t}^{(i)} \pmod{q}.$$

Since

$$\operatorname{rank}(\mathbf{A}^{(i)}) = 1$$

and $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{4\times 4}$, the left nullspace of $\mathbf{A}^{(i)}$ has dimension

$$4 - \operatorname{rank}(\mathbf{A}^{(i)}) = 3.$$

Hence, there exists a matrix $\mathbf{U}^{(i)} \in \mathbb{Z}_q^{3\times 4}$ whose rows form a basis of the left nullspace of $\mathbf{A}^{(i)}$, so that

$$\mathbf{U}^{(i)}\mathbf{A}^{(i)} = \mathbf{0}.$$

Multiplying the slot-wise relation by $\mathbf{U}^{(i)}$ eliminates $\mathbf{s}_1^{(i)}$:

$$\mathbf{U}^{(i)}\mathbf{s}_2^{(i)} = \mathbf{U}^{(i)}\mathbf{t}^{(i)} \pmod{q}.$$

Since $\mathbf{U}^{(i)}$ has rank 3, each slot contributes exactly three linearly independent projected equations. Stacking all n slots therefore yields a global projected system containing $3n$ independent equations.

Because one polynomial component of $\mathbf{s}_2$ is assumed to be known, the remaining unknown part of $\mathbf{s}_2$ contains exactly $3n$ coefficients. These coefficients are collected in the vector $\mathbf{y} \in \mathbb{Z}_q^{3n}$.

The stacked projected equations therefore take the form

$$\mathbf{P}\mathbf{y} = \mathbf{x} \pmod{q}, \qquad \mathbf{P} \in \mathbb{Z}_q^{3n\times 3n}.$$

If

$$\operatorname{rank}(\mathbf{P}) = 3n,$$

then $\mathbf{P}$ is invertible over $\mathbb{Z}_q$. Hence, the system admits a unique solution

$$\mathbf{y} = \mathbf{P}^{-1}\mathbf{x} \pmod{q}.$$

Substituting the known polynomial component recovers the full vector $\mathbf{s}_2$.

https://doi.org/10.3390/math14101749

Conversely, if $\mathrm{rank}(\mathbf{P}) < 3n$, then the columns of $\mathbf{P}$ are linearly dependent, so the projected system does not uniquely determine the unknown vector $\mathbf{y}$. Hence, $\mathbf{s}_2$ cannot be uniquely recovered from the projected system alone. □

Remark 1. *Since* $\mathrm{rank}(\mathbf{A}^{(i)}) = 1$, *the left nullspace has dimension* $\dim \mathcal{N}_L(\mathbf{A}^{(i)}) = 3$. *Although the projected system yields four scalar equations per slot, at most three are linearly independent. Consequently, the assembled matrix* $\mathbf{P}$ *has maximal rank* $3n$, *and unique recovery occurs precisely when* $\mathrm{rank}(\mathbf{P}) = 3n$.

Complexity in Finding $\mathbf{s}_2$

In the slot-wise rank-1 setting with one-component leakage, the recovery of $\mathbf{s}_2$ reduces to solving an overdetermined linear system $Py = x \pmod q$, where

$$P \in \mathbb{Z}_q^{4n \times 3n}, \qquad y \in \mathbb{Z}_q^{3n}.$$

The vector y contains the remaining unknown coefficients of $\mathbf{s}_2$ after one polynomial component has been exposed.

The cost of constructing the system consists of three main steps. First, applying the NTT to a constant number of polynomial entries costs $O(n \log n)$.

Second, for each of the n NTT slots, the rank test and the computation of a left-nullspace basis involve only constant-size 4×4 matrices. Hence, the total cost of this step is $O(n)$.

Third, assembling the global matrix P and the right-hand side x requires forming $4n$ linear equations in $3n$ unknowns, giving cost $O(n^2)$.

The dominant step is solving the modular linear system. Since the system is overdetermined, we first select a full-rank square subsystem of size $3n \times 3n$ and solve it by Gaussian elimination over $\mathbb{Z}_q$. This gives the complexity

$$O((3n)^3) = O(27n^3).$$

Equivalently, if elimination is applied directly to the rectangular $4n \times 3n$ matrix, the cost is bounded by

$$O((4n)(3n)^2) = O(36n^3).$$

Thus, the recovery of $\mathbf{s}_2$ is polynomial in n once the rank-deficient structure and the one-component leakage assumption are present. Table 1 summarizes the complexity comparison.

Table 1. Complexity comparison between the toy instantiation and standard Dilithium parameters (slot-wise rank-1, one-component leakage).

Quantity	**Toy ($n = 16$, $q = 769$)**	**Dilithium2 ($n = 256$, $q = 8,380,417$)**
Unknowns ($3n$)	48	768
Equations ($4n$)	64	1024
System size ($4n \times 3n$)	64×48	1024×768
NTT preprocessing †	$O(n \log n)$	$O(n \log n)$
Linear-system assembly	$O(n^2)$	$O(n^2)$
Dominant solve cost ‡	$36n^3 \approx 1.47 \times 10^5$	$36n^3 \approx 6.04 \times 10^8$
Approx. power-of-two (solve)	$\approx 2^{17}$	$\approx 2^{29}$

† Entry-wise NTT is applied to a constant number of polynomials (4×4 matrix and 4-vector), so the total cost is $O(n \log n)$. ‡ Solve cost corresponds to modular elimination on a $4n \times 3n$ system (e.g., via reduction to a solvable $4n \times 3n$ subsystem).

The recovery procedure for $\mathbf{s}_2$ under the rank-1 setting is summarized in Algorithm 1.

Algorithm 1 Recovery of $\mathbf{s}_2$ via Global Projection and Coefficient Extraction (Rank-1, 4×4)

Input: Public $(\mathbf{A}, \mathbf{t}) \in R_q^{4\times 4} \times R_q^4$, modulus q, degree n, and leakage of $s_{2,1} \in R_q$
Output: Recovered $\mathbf{s}_2$ or FAIL

Compute the NTT-domain representations $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{4\times 4}$ and $\mathbf{t}^{(i)} \in \mathbb{Z}_q^4$ for all $i = 0, \ldots, n-1$.
▷ (1) Weak-key detection: slot-wise rank-1
for $i = 0$ to $n-1$ **do**
  if $\mathrm{rank}(\mathbf{A}^{(i)}) \neq 1$ **then**
    return FAIL
  end if
end for
▷ (2) Compute slot-wise projectors and assemble a global projector $\mathbf{U}$
for $i = 0$ to $n-1$ **do**
  Compute $\mathbf{U}^{(i)} \in \mathbb{Z}_q^{4\times 4}$ such that $\mathbf{U}^{(i)}\mathbf{A}^{(i)} = \mathbf{0}$.
end for
Assemble $\mathbf{U} := (\mathbf{U}^{(0)}, \ldots, \mathbf{U}^{(n-1)})$, i.e., $\mathbf{U}$ is a 4×4 matrix whose entries are length-n slot-vectors.
▷ (3) Unknown representation in coefficient domain
Let $s_{2,2}(X) = \sum_{j=0}^{n-1} y_{j+1} X^j$, $s_{2,3}(X) = \sum_{j=0}^{n-1} y_{n+j+1} X^j$, $s_{2,4}(X) = \sum_{j=0}^{n-1} y_{2n+j+1} X^j$.
Define $\mathbf{s}_2'(X) := (s_{2,1}(X), s_{2,2}(X), s_{2,3}(X), s_{2,4}(X))^T$ with unknown vector $\mathbf{y} \in \mathbb{Z}_q^{3n}$.
▷ (4) Global projection and equation extraction
Compute the projected target $\mathbf{Ut}$ in the NTT domain:

$$(\mathbf{Ut})^{(i)} \leftarrow \mathbf{U}^{(i)}\mathbf{t}^{(i)} \pmod q, \qquad i = 0, \ldots, n-1.$$

Compute the projected unknown expression:

$$(\mathbf{Us}_2')^{(i)} \leftarrow \mathbf{U}^{(i)} \cdot NTT(\mathbf{s}_2')^{(i)} \pmod q, \qquad i = 0, \ldots, n-1.$$

Flatten the 4×1 projected vectors across all slots into $4n$ scalar equations $p^{(1)}, \ldots, p^{(4n)}$ by row-wise indexing:

$$p^{(4i+r)}: \quad (\mathbf{Us}_2')_r^{(i)} = (\mathbf{Ut})_r^{(i)}, \qquad i = 0, \ldots, n-1,\ r = 1, \ldots, 4.$$

▷ (5) Assemble $\mathbf{Py} = \mathbf{x} \pmod q$
Initialize $\mathbf{P} \in \mathbb{Z}_q^{4n\times 3n}$ and $\mathbf{x} \in \mathbb{Z}_q^{4n}$.
for $j = 1$ to $4n$ **do**
  Set $\mathbf{x}_j \leftarrow$ right-hand side of $p^{(j)}$.
  Extract the coefficients of the left-hand linear form in variables $\mathbf{y}$ from $p^{(j)}$ and store them as row j of $\mathbf{P}$.
end for
▷ (6) Rank test, solve, and verify
Compute $r \leftarrow \mathrm{rank}(\mathbf{P})$ over $\mathbb{Z}_q$.
if $r < 3n$ **then**
  return FAIL ▷ Insufficient independent constraints
end if
Select $3n$ rows of $\mathbf{P}$ forming a full-rank square subsystem $\mathbf{P}_{\mathrm{sub}}\mathbf{y} = \mathbf{x}_{\mathrm{sub}}$.
Solve $\mathbf{y} \leftarrow \mathbf{P}_{\mathrm{sub}}^{-1}\mathbf{x}_{\mathrm{sub}} \pmod q$ using modular Gaussian elimination.
Substitute $\mathbf{y}$ into $\mathbf{s}_2'(X)$ to obtain $\mathbf{s}_2(X)$.
Verify that all $4n$ equations hold modulo q.
if verification fails **then**
  return FAIL
end if
return $\mathbf{s}_2$

5.4. Numerical Illustration for $n = 16$

We illustrate the rank–leakage threshold phenomenon under reduced parameters $q = 769$ and $n = 16$.

Let $\mathbf{A} \in R_q^{4\times 4}$ and $\mathbf{t} \in R_q^4$ denote a public key satisfying the MLWE relation as in Equation (1).

$$\mathbf{A} = \begin{pmatrix} [50, 244, \cdots, 282] & [308, 611, \cdots, 599] & [267, 657, \cdots, 337] & [199, 602, \cdots, 15] \\ [768, 251, \cdots, 696] & [486, 285, \cdots, 104] & [656, 679, \cdots, 210] & [319, 676, \cdots, 217] \\ [425, 712, \cdots, 147] & [311, 18, \cdots, 75] & [347, 480, \cdots, 462] & [538, 496, \cdots, 16] \\ [737, 513, \cdots, 505] & [172, 607, \cdots, 650] & [229, 294, \cdots, 159] & [211, 150, \cdots, 395] \end{pmatrix}$$

The NTT-matrix representation of $\mathbf{t}$ is explicitly given as follows:

$$\mathbf{t} = \begin{pmatrix} [421, 126, 268, 564, 102, 139, 700, 472, 264, 607, 650, 463, 743, 739, 281, 120] \\ [138, 137, 237, 93, 197, 97, 435, 736, 351, 406, 43, 301, 119, 293, 583, 69] \\ [56, 333, 260, 739, 10, 83, 632, 207, 150, 641, 593, 442, 222, 461, 119, 653] \\ [729, 714, 21, 166, 473, 606, 533, 91, 587, 330, 154, 555, 282, 604, 608, 28] \end{pmatrix}$$

Assume that $\mathbf{A}$ satisfies the slot-wise rank-1 condition in the NTT domain:

$$\text{rank}(\mathbf{A}^{(i)}) = 1 \qquad \text{for all } i = 0, \ldots, 15.$$

For example, we check the rank of the first slot of $\mathbf{A}$

$$\mathbf{A}^{(0)} = \begin{pmatrix} 50 & 308 & 267 & 199 \\ 768 & 486 & 656 & 319 \\ 425 & 311 & 347 & 538 \\ 737 & 172 & 229 & 211 \end{pmatrix} \tag{3}$$

and

$$\text{rank}(\mathbf{A}^{(0)}) = 1.$$

We further assume that one component of $\mathbf{s}_2$ is fully known, namely

$$s_2^{(1)}(x) = 2 + x + x^2 + x^3 - 2x^4 + 2x^5 + x^7 - 2x^9 + x^{10} - x^{11} + 2x^{12} - x^{14} + x^{15}$$

and the full matrix $\mathbf{s}_2$ is as follows:

$$\mathbf{s}_2' = \begin{pmatrix} 2 + x + x^2 + \cdots + x^{15} \\ y_1 + y_2 x + y_3 x^2 + \cdots + y_n x^{n-1} \\ y_{n+1} + y_{n+2} x + \cdots + y_{2n} x^{n-1} \\ y_{2n+1} + y_{2n+2} x + \cdots + y_{3n} x^{n-1} \end{pmatrix}.$$

5.5. Slot-Wise Projection

For each slot i, compute a left-nullspace basis matrix $\mathbf{U}^{(i)} \in \mathbb{Z}_q^{4\times 4}$; for example, our first slot of $\mathbf{U}$ is

$$\mathbf{U}^{(0)} = \begin{pmatrix} 157 & 271 & 418 & 93 \\ 712 & 159 & 123 & 338 \\ 89 & 597 & 235 & 550 \\ 325 & 19 & 197 & 336 \end{pmatrix}$$

satisfying

$$\mathbf{U}^{(i)} \mathbf{A}^{(i)} = \mathbf{0}. \tag{4}$$

Multiplying the slot-wise MLWE relation by $\mathbf{U}^{(i)}$ eliminates the $\mathbf{s}_1^{(i)}$ component and yields

$$\mathbf{U}^{(i)} \mathbf{s}_2^{(i)} = \mathbf{U}^{(i)} \mathbf{t}^{(i)} \pmod{q}. \tag{5}$$

Stacking all slots produces a linear system over $\mathbb{Z}_q$. Our first slots of $p^{(1)}$ are as follows:

$$\begin{aligned} p^{(1)} :& 101y_1 + 56y_2 + 663y_3 + 695y_4 + 195y_5 + 702y_6 + 374y_7 + 116y_8 + 110y_9 + 396y_{10} + 349y_{11} + 26y_{12} + 555y_{13} \\ &+ 460y_{14} + 118y_{15} + 271y_{16} + 187y_{17} + 58y_{18} + 55y_{19} + 198y_{20} + 559y_{21} + 13y_{22} + 662y_{23} + 230y_{24} + 59y_{25} \\ &+ 520y_{26} + 334y_{27} + 741y_{28} + 53y_{29} + 37y_{30} + 287y_{31} + 418y_{32} + 588y_{33} + 425y_{34} + 761y_{35} + 125y_{36} + 450y_{37} \\ &+ 82y_{38} + 449y_{39} + 386y_{40} + 313y_{41} + 204y_{42} + 273y_{43} + 60y_{44} + 216y_{45} + 470y_{46} + 154y_{47} + 93y_{48} + 721 = 143. \end{aligned}$$

Since each slot contributes four projected equations and there are $n = 16$ slots, the projected system yields 64 scalar equations $p^{(1)}, \ldots, p^{(64)}$. However, since each projected block has rank at most 3, the number of independent constraints is at most 48.

After substituting the known component $s_2^{(1)}$, the remaining unknown coefficients are represented by a vector $\mathbf{y} \in \mathbb{Z}_q^{3n}$.

The projected equations across all slots therefore assemble into the overdetermined system

$$\mathbf{P}\mathbf{y} = \mathbf{x} \pmod{q}, \tag{6}$$

where $\mathbf{P} \in \mathbb{Z}_q^{4n \times 3n}$.

$$\mathbf{P} \cdot \mathbf{y} = \begin{pmatrix} 101 & 56 & 663 & 695 & \cdots & 93 \\ 264 & 370 & 437 & 327 & \cdots & 502 \\ 340 & 513 & 48 & 760 & \cdots & 78 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 78 & 330 & 568 & 451 & \cdots & 274 \\ 31 & 564 & 413 & 72 & \cdots & 172 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{48} \end{pmatrix} = \begin{pmatrix} 191 \\ 678 \\ 713 \\ \vdots \\ 156 \\ 689 \end{pmatrix}$$

The assembled system takes the form $\mathbf{P}\mathbf{y} = \mathbf{x} \pmod{q}$. In this instance, the system admits a unique solution over $\mathbb{Z}_q$, which we compute via modular Gaussian elimination.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{45} \\ y_{46} \\ y_{47} \\ y_{48} \end{pmatrix} = \begin{pmatrix} 768 \\ 767 \\ 1 \\ 0 \\ \vdots \\ 767 \\ 767 \\ 2 \\ 0 \end{pmatrix}.$$

The full polynomial vectors of $\mathbf{s}_2$ are as follows:

$$\mathbf{s}_2 = \begin{pmatrix} x^{15} + 768x^{14} + 0x^{13} + 2x^{12} + 768x^{11} + x^{10} + 767x^9 + 0x^8 + x^7 + 0x^6 + \ldots \\ \ldots 2x^5 + 767x^4 + x^3 + x^2 + x + 2 \\ 768x^{15} + 767x^{14} + x^{13} + 0x^{12} + 0x^{11} + x^{10} + 767x^9 + 767x^8 + 2x^7 + 0x^6 + \ldots \\ \ldots x^5 + x^4 + x^3 + 0x^2 + x + 1 \\ 2x^{15} + 0x^{14} + x^{13} + x^{12} + 768x^{11} + 2x^{10} + 0x^9 + 0x^8 + 768x^7 + 768x^6 + \ldots \\ \ldots 767x^5 + 767x^4 + 0x^3 + x^2 + x + 0 \\ x^{15} + x^{14} + 768x^{13} + x^{12} + x^{11} + x^{10} + x^9 + 0x^8 + 2x^7 + x^6 + \ldots \\ \ldots 768x^5 + 768x^4 + 767x^3 + 767x^2 + 2x + 0 \end{pmatrix}.$$

5.6. Recovery Consequence

The recovered $\mathbf{s}_2$ collapses the MLWE relation into a linear system

$$\mathbf{A}\mathbf{s}_1 = \mathbf{t} - \mathbf{s}_2 \pmod{q}.$$

Since $\mathrm{rank}(\mathbf{A}^{(i)}) = 1$ in each slot, the solution space of $\mathbf{s}_1$ is affine with dimension governed by the nullity of $\mathbf{A}$. Consequently, the residual ambiguity in $\mathbf{s}_1$ corresponds precisely to the nullspace dimension induced by the slot-wise rank deficiency.

Under uniform parameter generation, such a rank-1 slot-wise collapse occurs with negligible probability. This confirms the rank–leakage threshold behavior predicted by the theoretical analysis.

6. Residual Secret Recovery of $\mathbf{s}_1$

Once $\mathbf{s}_2$ has been recovered, the MLWE relation reduces to

$$\mathbf{A}\mathbf{s}_1 = \mathbf{b} \pmod{q}, \qquad \mathbf{b} := \mathbf{t} - \mathbf{s}_2. \tag{7}$$

Under the slot-wise representation in the NTT domain, this system decomposes into n independent linear relations

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} = \mathbf{b}^{(i)}, \qquad i = 0, \ldots, n-1. \tag{8}$$

We analyze the structure of this reduced system under the same slot-wise rank-degradation model.

6.1. Slot-Wise Linear Reduction

Assume that each NTT slice satisfies

$$\mathrm{rank}(\mathbf{A}^{(i)}) = 1.$$

Proposition 1 (Rank-1 slice induces a scalar constraint)**.** *Let* $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{4\times 4}$ *satisfy* $\mathrm{rank}(\mathbf{A}^{(i)}) = 1$ *and suppose*

$$\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} = \mathbf{b}^{(i)} \pmod{q}.$$

Then there exist vectors $\mathbf{u}^{(i)}, \mathbf{v}^{(i)} \in \mathbb{Z}_q^4$ *and a scalar* $\alpha_i \in \mathbb{Z}_q$ *such that*

$$\mathbf{A}^{(i)} = \mathbf{u}^{(i)}(\mathbf{v}^{(i)})^\top, \qquad \mathbf{b}^{(i)} = \alpha_i\, \mathbf{u}^{(i)} \pmod{q},$$

and consequently, the relation reduces to the scalar constraint

$$(\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} = \alpha_i \pmod{q}.$$

Proof. Since $\mathrm{rank}(\mathbf{A}^{(i)}) = 1$, the matrix admits a rank-1 factorization

$$\mathbf{A}^{(i)} = \mathbf{u}^{(i)}(\mathbf{v}^{(i)})^\top$$

for some nonzero vectors $\mathbf{u}^{(i)}, \mathbf{v}^{(i)} \in \mathbb{Z}_q^4$.

Substituting this representation into $\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} = \mathbf{b}^{(i)}$ yields

$$\mathbf{u}^{(i)}(\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} = \mathbf{b}^{(i)} \pmod{q}.$$

Define the scalar

$$\alpha_i := (\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} \in \mathbb{Z}_q.$$

https://doi.org/10.3390/math14101749

Then the left-hand side becomes

$$\mathbf{u}^{(i)}(\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} = \mathbf{u}^{(i)}\alpha_i.$$

Hence

$$\mathbf{b}^{(i)} = \alpha_i\, \mathbf{u}^{(i)} \pmod q,$$

which shows that $\mathbf{b}^{(i)}$ lies in the one-dimensional image space $\mathrm{Im}(\mathbf{A}^{(i)}) = \mathrm{span}\{\mathbf{u}^{(i)}\}$. Consequently, the relation reduces to the scalar constraint

$$(\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} = \alpha_i \pmod q.$$

□

Proposition 1 shows that under rank-1 slices, the MLWE relation reduces to a system of scalar constraints. As a result, recovering the secret vector $\mathbf{s}_1$ can be formulated as a bounded modular linear system.

Despite this reduction, the problem remains difficult in general. The resulting system $\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod q$ has significantly fewer equations than unknowns, and the solution space is constrained only by the bounded domain $\mathbf{z} \in [-\eta, \eta]^{4n}$. This corresponds to a structured instance of the Short Integer Solution (SIS) problem, which is widely believed to be hard. In particular, the low rank of $\mathbf{A}^{(i)}$ reduces each slot to a single scalar constraint, leaving a large nullspace that preserves substantial ambiguity in $\mathbf{s}_1$.

Let $\mathbf{z}$ denote the coefficient vector of $\mathbf{s}_1$. Stacking all slot-wise constraints obtained from Proposition 1 yields the global linear system

$$\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod q, \tag{9}$$

where $\mathbf{M} \in \mathbb{Z}_q^{n\times 4n}$ and

$$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})^\top.$$

In CRYSTALS-Dilithium and ML-DSA, the secret coefficients satisfy

$$z_j \in [-\eta, \eta], \qquad \eta = 2.$$

Recovering $\mathbf{s}_1$ therefore requires solving the bounded modular system

$$\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod q, \qquad \mathbf{z} \in [-\eta, \eta]^{4n}. \tag{10}$$

6.1.1. Meet-in-the-Middle Strategy

A direct enumeration of all bounded vectors $\mathbf{z}$ requires $(2\eta+1)^{4n}$ candidates and is computationally infeasible. To reduce the search space, we employ a meet-in-the-middle (MiTM) strategy.

Partition the vector $\mathbf{z}$ into two halves

$$\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_R),$$

where each part contains $2n$ coefficients.

Equation (10) can then be rewritten as

$$\mathbf{M}_L\mathbf{z}_L + \mathbf{M}_R\mathbf{z}_R = \boldsymbol{\alpha} \pmod q.$$

In this setting, exhaustive search remains infeasible, and practical recovery requires exploiting an additional structure in the bounded solution space.

The MiTM procedure enumerates all bounded values of $\mathbf{z}_L$ and stores the corresponding partial sums $\mathbf{M}_L\mathbf{z}_L$ in a hash table. For each candidate $\mathbf{z}_R$, the algorithm checks whether

$$\boldsymbol{\alpha} - \mathbf{M}_R\mathbf{z}_R$$

appears in the table, thereby producing candidate solutions $\mathbf{z}$. The meet-in-the-middle recovery framework for $\mathbf{s}_1$ is summarized in Algorithm 2.

Algorithm 2 MiTM Recovery Framework for $\mathbf{s}_1$

Input: Public matrix $\mathbf{A}$, public vector $\mathbf{t}$, recovered secret vector $\mathbf{s}_2$
Output: Candidate solutions for $\mathbf{s}_1$

Compute $\mathbf{b} \leftarrow \mathbf{t} - \mathbf{s}_2 \pmod q$
for $i = 0, \ldots, n-1$ **do**
    Extract the slot-wise relation $\mathbf{A}^{(i)}\mathbf{s}_1^{(i)} = \mathbf{b}^{(i)} \pmod q$
    Using the rank-1 structure of $\mathbf{A}^{(i)}$, determine $\mathbf{v}^{(i)}$ and scalar α_i such that

$$(\mathbf{v}^{(i)})^\top \mathbf{s}_1^{(i)} = \alpha_i \pmod q$$

end for
Stack all slot constraints to obtain

$$\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod q, \qquad \boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})^\top$$

Split the unknown vector as

$$\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_R), \qquad \mathbf{z}_L, \mathbf{z}_R \in [-\eta, \eta]^{2n}$$

for all $\mathbf{z}_L \in [-\eta, \eta]^{2n}$ **do**
    Compute the left partial sum $\mathbf{c}_L = \mathbf{M}_L\mathbf{z}_L \pmod q$
    Store $(\mathbf{c}_L, \mathbf{z}_L)$ in a hash table
end for
for all $\mathbf{z}_R \in [-\eta, \eta]^{2n}$ **do**
    Compute the target value $\mathbf{c}_R = \boldsymbol{\alpha} - \mathbf{M}_R\mathbf{z}_R \pmod q$
    if $\mathbf{c}_R$ exists in the hash table **then**
        Recover the matching $\mathbf{z}_L$ and form candidate $\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_R)$
        if $\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod q$ **then**
            Reconstruct the corresponding $\mathbf{s}_1$
            Store $\mathbf{s}_1$ as a valid candidate
        end if
    end if
end for
Return all valid candidates for $\mathbf{s}_1$

6.1.2. Numerical Illustration of the MiTM Search

To illustrate the meet-in-the-middle recovery framework, we use the same reduced parameter setting as in Section 5, namely $q = 769$ and $n = 16$. After recovering $\mathbf{s}_2$, the MLWE relation reduces to

$$\mathbf{A}\mathbf{s}_1 = \mathbf{b} \pmod q, \qquad \mathbf{b} := \mathbf{t} - \mathbf{s}_2.$$

In the NTT domain, this gives

$$\mathbf{b} = \begin{pmatrix} [436, 385, 280, 356, 160, 91, 400, 532, 471, 436, 418, 206, 160, 217, 319, 222] \\ [668, 21, 546, 74, 484, 376, 157, 722, 256, 348, 637, 138, 358, 71, 304, 597] \\ [630, 657, 463, 459, 101, 286, 598, 541, 162, 52, 325, 586, 628, 126, 673, 83] \\ [613, 239, 45, 338, 84, 193, 54, 496, 475, 105, 464, 549, 367, 711, 516, 463] \end{pmatrix}.$$

The coefficients of the secret vector $\mathbf{s}_1$ are represented by unknown variables $z_1, \ldots, z_{64}$, forming the coefficient vector

$$\mathbf{z} = (z_1, \ldots, z_{64})^\top.$$

After applying the NTT, each slot $\mathbf{s}_1^{(i)}$ becomes a linear expression in these variables, with rows corresponding respectively to $z_1 : z_{16}$, $z_{17} : z_{32}$, $z_{33} : z_{48}$, and $z_{49} : z_{64}$.

For the reduction in Proposition 1, we consider the first NTT slot $\mathbf{A}^{(0)}$ as in Equation (3). Since $\operatorname{rank}(\mathbf{A}^{(0)}) = 1$, the matrix admits a rank-1 factorization

$$\mathbf{A}^{(0)} = \mathbf{u}^{(0)} (\mathbf{v}^{(0)})^\top \pmod{769},$$

with

$$\mathbf{u}^{(0)} = \begin{pmatrix} 50 \\ 768 \\ 425 \\ 737 \end{pmatrix}, \qquad \mathbf{v}^{(0)} = \begin{pmatrix} 1 \\ 283 \\ 113 \\ 450 \end{pmatrix}.$$

Let

$$\alpha_0 := (\mathbf{v}^{(0)})^\top \mathbf{s}_1^{(0)}.$$

Substituting the NTT representation of $\mathbf{s}_1^{(0)}$ gives

$$\alpha_0 = \begin{pmatrix} 1 & 283 & 113 & 450 \end{pmatrix} \begin{pmatrix} 304z_1 + 633z_2 + 587z_3 + \cdots + 43z_{15} + z_{16} \\ 304z_{17} + 633z_{18} + 587z_{19} + \cdots + 43z_{31} + z_{32} \\ 304z_{33} + 633z_{34} + 587z_{35} + \cdots + 43z_{47} + z_{48} \\ 304z_{49} + 633z_{50} + 587z_{51} + \cdots + 43z_{63} + z_{64} \end{pmatrix}.$$

$$\begin{aligned} \alpha_0 = {} & 304z_1 + 633z_2 + 587z_3 + 729z_4 + 625z_5 + 712z_6 + 410z_7 + 707z_8 + 392z_9 + 27z_{10} + 251z_{11} + 596z_{12} + 300z_{13} \\ & + 311z_{14} + 43z_{15} + z_{16} + 673z_{17} + 731z_{18} + 17z_{19} + 215z_{20} + 5z_{21} + 18z_{22} + 680z_{23} + 141z_{24} + 200z_{25} + 720z_{26} \\ & + 285z_{27} + 257z_{28} + 310z_{29} + 347z_{30} + 634z_{31} + 283z_{32} + 516z_{33} + 12z_{34} + 197z_{35} + 94z_{36} + 646z_{37} + 480z_{38} \\ & + 190z_{39} + 684z_{40} + 463z_{41} + 744z_{42} + 679z_{43} + 445z_{44} + 64z_{45} + 538z_{46} + 245z_{47} + 113z_{48} \\ & + 687z_{49} + 320z_{50} + 383z_{51} + 456z_{52} + 565z_{53} + 496z_{54} + 709z_{55} + 553z_{56} + 299z_{57} + 615z_{58} \\ & + 676z_{59} + 588z_{60} + 425z_{61} + 761z_{62} + 125z_{63} + 450z_{64}. \end{aligned}$$

Since $\mathbf{b}^{(0)} = \alpha_0 \mathbf{u}^{(0)} \pmod{769}$, each component satisfies $b_j^{(0)} \equiv \alpha_0 \, u_j^{(0)} \pmod{769}$. Hence,

$$\alpha_0 \equiv b_j^{(0)} (u_j^{(0)})^{-1} \pmod{769}, \qquad j = 1, \ldots, 4.$$

All components yield the same value of α_0; hence, we use the first entry

$$\alpha_0 \equiv b_1^{(0)} (u_1^{(0)})^{-1} \pmod{769}.$$

Thus,

$$\alpha_0 = 436 \cdot 323 \pmod{769} = 101.$$

Repeating the same procedure for each slot $i = 0, \ldots, 15$ produces 16 scalar equations. Stacking these constraints yields

$$\mathbf{M}\mathbf{z} = \boldsymbol{\alpha} \pmod{769}, \qquad \mathbf{M} \in \mathbb{Z}_{769}^{16 \times 64}.$$

where

$$\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{15})^\top$$

and $\mathbf{M} \in \mathbb{Z}_{769}^{16 \times 64}$.

$$\mathbf{M} \cdot \mathbf{z} = \begin{pmatrix} 304 & 633 & 587 & 729 & \cdots & 450 \\ 587 & 712 & 392 & 596 & \cdots & 450 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 726 & 458 & 469 & 173 & \cdots & 450 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{63} \\ z_{64} \end{pmatrix} = \begin{pmatrix} 101 \\ 717 \\ 541 \\ \vdots \\ 593 \\ 508 \end{pmatrix}.$$

To apply the meet-in-the-middle strategy, we split the coefficient vector as

$$\mathbf{z} = (\mathbf{z}_L, \mathbf{z}_R),$$

where

$$\mathbf{z}_L = (z_1, \ldots, z_{32})^\top, \qquad \mathbf{z}_R = (z_{33}, \ldots, z_{64})^\top.$$

Accordingly, the matrix $\mathbf{M}$ is partitioned as

$$\mathbf{M} = (\mathbf{M}_L \mid \mathbf{M}_R),$$

with

$$\mathbf{M}_L \in \mathbb{Z}_{769}^{16\times 32}, \qquad \mathbf{M}_R \in \mathbb{Z}_{769}^{16\times 32}.$$

Hence, the system $\mathbf{Mz} = \alpha \pmod{769}$ becomes

$$\mathbf{M}_L\mathbf{z}_L + \mathbf{M}_R\mathbf{z}_R = \alpha \pmod{769}.$$

For this experiment, the recovered solution is split as

$$\mathbf{z}_L = \begin{pmatrix} 1 \\ 1 \\ 2 \\ \vdots \\ 2 \\ 1 \\ 2 \end{pmatrix}, \qquad \mathbf{z}_R = \begin{pmatrix} -2 \\ 1 \\ 1 \\ \vdots \\ -1 \\ 1 \\ 0 \end{pmatrix}.$$

Thus, the first row of the system decomposes as

$$(M_L)_{1,*}z_L + (M_R)_{1,*}z_R = \alpha_0 \pmod{769}.$$

Combining the recovered $\mathbf{s}_1$ and $\mathbf{s}_2$, we verify that

$$\mathbf{As}_1 + \mathbf{s}_2 = \mathbf{t} \pmod{769},$$

which confirms the correctness of the recovered secret pair.

This numerical example illustrates how slot-wise rank degeneration transforms the MLWE relation into a reduced linear system that can be explored using meet-in-the-middle search under bounded secret coefficients.

6.1.3. Complexity Under ML-DSA Parameters

We now estimate the cost of recovering $\mathbf{s}_1$ after $\mathbf{s}_2$ has been recovered. Under ML-DSA parameters, we have $n = 256$ and $\eta = 2$. Hence, each coefficient of $\mathbf{s}_1$ lies in

$$[-\eta, \eta] = [-2, 2],$$

which contains $2\eta + 1 = 5$ possible values.

Since $\mathbf{s}_1 \in R_q^4$, the total number of unknown coefficients is $4n$. A direct bounded search therefore requires

$$(2\eta + 1)^{4n} = 5^{4n}$$

candidates. For $n = 256$, this becomes

$$5^{1024} \approx 2^{2377}.$$

A meet-in-the-middle strategy splits the unknown vector into two halves,

$$z = (z_L, z_R),$$

where each half contains $2n$ coefficients. The algorithm enumerates all possibilities for one half and stores the corresponding partial sums, then searches for matching values from the other half. Therefore, the time and memory cost are approximately

$$(2\eta + 1)^{2n} = 5^{2n}.$$

For $n = 256$, this gives

$$5^{512} \approx 2^{1189}.$$

The above estimates assume that the coefficients are uniformly distributed over $[-2, 2]$. However, in ML-DSA, the secret coefficients are sampled from a centered binomial distribution (CBD), which has lower entropy than the uniform distribution. For $\eta = 2$, the coefficient probabilities are

$$P(0) = \frac{6}{16}, \qquad P(\pm 1) = \frac{4}{16}, \qquad P(\pm 2) = \frac{1}{16}.$$

Let

$$H_{\mathrm{CBD}} = -\sum_i p_i \log_2 p_i$$

denote the Shannon entropy [26] of the centered binomial distribution used in ML-DSA secret sampling; we obtain

$$H_{\mathrm{CBD}} \approx 2.03 \text{ bits}.$$

Under this entropy-based heuristic model, the MiTM complexity can be approximated by

$$2^{2nH_{\mathrm{CBD}}}.$$

For $n = 256$, this gives approximately

$$2^{2(256)(2.03)} \approx 2^{1039}.$$

If additional leakage reveals m coefficients of $\mathbf{s}_1$, then only $4n - m$ coefficients remain unknown. A direct search has complexity

$$5^{4n-m},$$

while the meet-in-the-middle search has approximate complexity

$$5^{(4n-m)/2}.$$

Under the CBD entropy model, the corresponding heuristic complexity becomes

$$2^{(4n-m)H_{\mathrm{CBD}}/2}.$$

For example, if $m = 256$ coefficients are known, then the MiTM complexity becomes

$$5^{(1024-256)/2} = 5^{384} \approx 2^{892},$$

while the entropy-based estimate becomes

$$2^{(1024-256)(2.03)/2} \approx 2^{780}.$$

These estimates should be interpreted as heuristic search-space estimates rather than exact cryptanalytic costs. They show that even after recovering $\mathbf{s}_2$, the remaining recovery of $\mathbf{s}_1$ remains computationally infeasible at standard ML-DSA parameters unless substantially more information or an additional exploitable structure are available. Table 2 summarizes the estimated search complexity for recovering $\mathbf{s}_1$ under ML-DSA parameters.

Table 2. Estimated search complexity for recovering $\mathbf{s}_1$ under ML-DSA parameters ($n = 256$, $\eta = 2$).

Method	Estimated Complexity
Brute-force bounded search	$5^{4n} = 5^{1024} \approx 2^{2377}$
MITM bounded search	$5^{2n} = 5^{512} \approx 2^{1189}$
MITM with CBD entropy model	$2^{2nH_{\mathrm{CBD}}} \approx 2^{1039}$
MITM with CBD entropy model and m leaked coefficients	$2^{(4n-m)H_{\mathrm{CBD}}/2}$

While the current analysis does not provide a practical method to recover $\mathbf{s}_1$ at standard ML-DSA parameters, it still shows how deviations in the rank structure of the public matrix affect the underlying problem. In particular, when the matrix becomes low-rank, the number of independent constraints is reduced. However, this does not remove the hardness of the bounded search problem.

These observations again point to simple defensive measures. In particular, verifying the rank of the public matrix during key generation is sufficient to prevent such unintended structural weaknesses.

7. Preventing Low-Rank Public Matrices in ML-DSA

The analysis in this work relies on slot-wise rank deficiencies in the public matrix $\mathbf{A}$ when examined in the NTT domain. In standard ML-DSA, the matrix $\mathbf{A}$ is deterministically generated from a public seed ρ through the procedure `ExpandA` [10]. Since the entries are derived from XOF output with rejection sampling, the resulting matrix is expected to behave similarly to a uniformly random element of $(T_q)^{k\times\ell}$, and slot-wise rank deficiency is therefore unlikely under honest parameter generation.

As a defensive safeguard, the matrix expansion procedure may be augmented with an explicit rank verification step to ensure that each slot matrix $\mathbf{A}^{(i)}$ has full rank.

Rank-Checked Matrix Expansion

A simple countermeasure is to validate the rank of each slot matrix after generating $\widehat{\mathbf{A}}$. If any slot fails the test, a fresh seed is derived and the matrix is regenerated until all slot matrices satisfy the required rank condition. Algorithm 3 summarizes the proposed rank-checked `ExpandA` procedure.

Algorithm 3 Rank-Checked `ExpandA`

Input: Public seed $\rho \in \{0,1\}^{256}$
Output: $\widehat{\mathbf{A}} \in (T_q)^{k \times \ell}$ such that $\text{rank}(\mathbf{A}^{(i)}) = \min(k, \ell)$ for all $i = 0, \ldots, n-1$

repeat
  Compute $\widehat{\mathbf{A}} \leftarrow \texttt{ExpandA}(\rho)$
  Extract slot matrices $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(n-1)}$
  valid $\leftarrow$ `true`
  for $i \leftarrow 0$ to $n-1$ **do**
    if $\text{rank}(\mathbf{A}^{(i)}) < \min(k, \ell)$ **then**
      Derive a fresh seed $\rho \leftarrow H(\rho)$
      valid $\leftarrow$ `false`
      break
    end if
  end for
until valid = `true`
return $\widehat{\mathbf{A}}$

This modification preserves the standard ML-DSA interface, since the procedure still maps a seed ρ to a public matrix $\widehat{\mathbf{A}}$. As a lightweight countermeasure, the standard `ExpandA` procedure can be augmented with a slot-wise rank validation step. If any NTT slot matrix fails to have full rank, the sampled seed is rejected and a fresh seed is derived. The additional cost is negligible, since the rank test is applied only to small $k \times \ell$ matrices. This provides a simple safeguard against unintended low-rank structures that could otherwise reduce the effective dimension of the underlying MLWE system.

8. Conclusions

This work studies how slot-wise rank degradation changes the dimensional structure of MLWE systems, using CRYSTALS-Dilithium as a representative setting. We focus on the case where the public matrix has rank 1 in every NTT slot. In this situation, the nullspace becomes large, which directly affects the balance between constraints and unknowns. When partial information on $\mathbf{s}_2$ is available, this imbalance makes the projected system algebraically tractable. The key point is that this effect does not rely on classical lattice trapdoors or hidden short bases, but arises purely from rank deficiency in the public matrix. Under honest parameter generation, such slot-wise rank collapse is unlikely. However, if the rank profile is deliberately manipulated, the solvability of the underlying MLWE relation can change significantly.

Even in an adversarial setting, such a matrix may still appear syntactically valid under the current specification, since no explicit verification of the slot-wise rank of $\mathbf{A}$ is required during key generation. From a defensive perspective, this suggests simple safeguards. One may recompute $\mathbf{A}$ from the public seed using the prescribed `ExpandA` procedure or perform lightweight rank checks directly in the NTT domain. These steps are sufficient to rule out such structural anomalies. Although the analysis focuses on the rank-1 case, the same dimensional viewpoint is likely to extend to higher or mixed-rank configurations.

Author Contributions: Conceptualization, N.S.K.A. and A.H.A.G.; Formal Analysis, N.S.K.A. and A.H.A.G.; Funding Acquisition, A.H.A.G. and M.R.K.A.; Methodology, N.S.K.A. and A.H.A.G.; investigation, N.S.K.A., A.H.A.G., M.A.A. and M.R.K.A.; writing—original draft preparation, N.S.K.A.; writing—review and editing, A.H.A.G.; visualization, N.S.K.A., A.H.A.G., M.A.A. and M.R.K.A.; supervision, A.H.A.G., M.A.A. and M.R.K.A.; project administration, A.H.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by Malaysia Cryptology Technology and Management Centre (PTPKM).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The first author would like to further express appreciation to the Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), and Malaysia Cryptology Technology and Management Centre for giving the opportunity to conduct this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LWE	Learning with errors
PQC	Post-quantum cryptography
MLWE	Module learning with errors
NTT	Number Theoretic Transform
SIS	Short integer solution
ML-DSA	Module-Lattice-Based Digital Signature Algorithm
MiTM	Meet in the Middle
XOF	eXtendable-Output Function

References

1. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **2009**, *56*, 34. [CrossRef]
2. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010*; Gilbert, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 1–23. [CrossRef]
3. Langlois, A.; Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.* **2015**, *75*, 565–599. [CrossRef]
4. D'Anvers, J.-P.; Karmakar, A.; Roy, S.S.; Vercauteren, F. *SABER: Module-LWR Based Key Exchange, CPA-Secure and CCA-Secure*; IACR Cryptology ePrint Archive; International Association for Cryptologic Research: Bellevue, WA, USA, 2018; Report 2018/230.
5. Bos, J.W.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*; IEEE: Piscataway, NJ, USA, 2018; pp. 353–367. [CrossRef]
6. 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)*; NIST Post-Quantum Cryptography Standardization Round ; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021.
7. Ducas, L.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: Digital signatures from module lattices. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*; IEEE: Piscataway, NJ, USA, 2018; pp. 238–254.
8. Micciancio, D.; Peikert, C. Hardness of SIS and LWE with small parameters. In *Advances in Cryptology–CRYPTO 2013*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8042, pp. 21–39. [CrossRef]
9. Koo, Z.; Lee, Y.; Lee, J.-W.; No, J.-S.; Kim, Y.-S. Improved reduction between SIS problems over structured lattices. *IEEE Access* **2021**, *9*, 157083–157092. [CrossRef]
10. National Institute of Standards and Technology (NIST). *FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)*; FIPS 204; NIST: Gaithersburg, MD, USA, 2024.
11. Aggarwal, D.; Ming, L.J.; Veliche, A. *Worst-Case to Average-Case Hardness of LWE: An Alternative Perspective*; IACR Cryptology ePrint Archive; International Association for Cryptologic Research: Bellevue, WA, USA, 2024; Report 2024/603.
12. Newton, P.; Richelson, S. A lower bound for proving hardness of learning with rounding with polynomial modulus. In *Advances in Cryptology–CRYPTO 2023*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2023; pp. 805–835. [CrossRef]
13. Gupte, A.; Vafa, N.; Vaikuntanathan, V. Continuous LWE is as hard as LWE and applications to learning Gaussian mixtures. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1162–1173.

14. Boudgoust, K.; Jeudy, C.; Tairi, E.; Wen, W. *Hardness of M-LWE with General Distributions and Applications to Leaky Variants*; IACR Cryptology ePrint Archive; International Association for Cryptologic Research: Bellevue, WA, USA, 2025; Report 2025/1472.
15. Krahmer, E.; Pessl, P.; Land, G.; Güneysu, T. Correction fault attacks on randomized CRYSTALS-Dilithium. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2024**, *3*, 174–199. [CrossRef]
16. ElGhamrawy, M.; Azouaoui, M.; Bronchain, O.; Renes, J.; Schneider, T.; Schönauer, M.; Seker, O.; van Vredendaal, C. From MLWE to RLWE: A differential fault attack on randomized and deterministic Dilithium. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2023**, *4*, 262–286. [CrossRef]
17. Islam, S.; Mus, K.; Singh, R.; Schaumont, P.; Sunar, B. Signature correction attack on Dilithium signature scheme. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*; IEEE: Piscataway, NJ, USA, 2022; pp. 647–663. [CrossRef]
18. Hemmert, T. *How to Backdoor LWE-Like Cryptosystems; Cryptology ePrint Archive*; Report 2022/1381; International Association for Cryptologic Research: Bellevue, WA, USA, 2022.
19. Vaikuntanathan, V.; Zamir, O. Improving algorithmic efficiency using cryptography: Trapdoored matrices and applications. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*; SIAM: Philadelphia, PA, USA, 2026; pp. 2554–2574.
20. Braverman, M.; Newman, S. Practical secure delegated linear algebra with trapdoored matrices. In *Proceedings of the Theory of Cryptography Conference (TCC)*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 97–118.
21. Smith-Tone, D.; Valenzuela, C. Cryptanalysis of the best HFE-LL′ constructions. In *Proceedings of the International Workshop on Security*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 167–186.
22. Bardet, M.; Briaud, P.; Bros, M.; Gaborit, P.; Tillich, J.-P. Revisiting algebraic attacks on MinRank and on the rank decoding problem. *Des. Codes Cryptogr.* **2023**, *91*, 3671–3707. [CrossRef]
23. Levina, A.; Kadykov, V.; Valluri, M.R. Security analysis of hybrid attack for NTRU-class encryption schemes. *IEEE Access* **2023**, *11*, 109939–109952. [CrossRef]
24. Boura, C.; David, N.; Derbez, P.; Leander, G.; Naya-Plasencia, M. Differential meet-in-the-middle cryptanalysis. In *Advances in Cryptology—CRYPTO 2023*; Springer: Cham, Switzerland, 2023; pp. 240–272. [CrossRef]
25. Wenger, E.; Saxena, E.; Malhou, M.; Thieu, E.; Lauter, K. Benchmarking attacks on learning with errors. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2025; pp. 279–297.
26. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef] [PubMed]