Too Far Behind? Narrowing the Gap with a Dual-Enhanced Two-Stage Algebraic Algorithm for LWE

Rui-Jie Wang, Hong-Sen Yang, Zhong-Xiao Wang(✉), Qun-Xiong Zheng(✉), and Ye-Chen Li

Information Engineering University, Zhengzhou 450001, China
ruijie_wang1@163.com
moonlight0833@outlook.com
zhongxiao_wang@126.com
qunxiong_zheng@163.com
yechen_li2002@163.com

Abstract. The Learning with Errors (LWE) problem forms the foundation for numerous post-quantum cryptographic schemes, such as the NIST-selected CRYSTALS-KYBER and CRYSTALS-DILITHIUM. Algebraic analysis of LWE traditionally relies on solving the Arora-Ge system via Gröbner bases, yet its performance is far from satisfactory when only a limited number of samples is available. Meanwhile, recent dual attacks have proven highly effective against concrete LWE-based algorithms. This gap motivates us to investigate whether integrating techniques from dual attacks into algebraic analysis can have a positive effect.

We propose a novel, two-stage algebraic algorithm for LWE. First, dual lattice reduction is applied to transform the original samples into lower-dimensional samples. From an algebraic perspective, this stage reduces the number of variables at the cost of increasing the noise. Second, instead of solving the classic Arora-Ge system, we introduce a new polynomial construction that exploits the error distribution and solves it via a resultant-based method. When given $m = n$ samples, our two-stage algorithm yields better complexity estimates for CRYSTALS-KYBER than the Gröbner-basis estimates reported in Steiner's work (Eurocrypt 2024). As an independent contribution, we show that for LWE with a small secret, applying the resultant-based method directly to the Arora-Ge system provides a provable complexity estimate that achieves an exponential speed-up over the proven bounds established by Steiner.

Finally, we show how various forms of side information—namely, perfect hints, modular hints, and approximate hints—can be systematically incorporated into our two-stage algorithm.

Keywords: Learning with Errors · Algebraic analysis · Dual lattice reduction · Gröbner bases · Resultant · CRYSTALS-KYBER · Hints

1 Introduction

The proposal of Shor algorithm in 1994 showed that cryptographic algorithms based on integer factorization can be broken in polynomial time on a quantum computer. This has led to a significant shift toward developing post-quantum cryptography (PQC) over the past few decades. PQC refers to cryptographic algorithms designed to remain secure against both classical and quantum adversaries. Among these, lattice-based algorithms have attracted considerable attention. To date, the National Institute of Standards and Technology (NIST) has announced five standardization schemes for PQC, three of which are lattice-based. Among these, CRYSTALS-KYBER [40] and CRYSTALS-DILITHIUM [29] are directly based on the Learning with Errors (LWE) problem, which was introduced by Regev [37]. Therefore, a thorough understanding of the concrete complexity of LWE is crucial, motivating the cryptanalysis of LWE from diverse perspectives.

In this paper, two analytical approaches for LWE are considered. The first is **algebraic analysis**, proposed by Arora and Ge in ICALP 2011 [5]. By exploiting the fact that each error coordinate is bounded in an integer interval $[\![-T, T]\!]$, they construct a polynomial system regarding the secret vector, where each polynomial has a degree $d = 2T + 1$. The system is subsequently solved using a linearization technique. In ACISP 2020, Sun et al. [42] studied the linearization technique specifically for the Arora-Ge system of binary error LWE (coordinates of the error are chosen from $\{0, 1\}$).

Gröbner basis methods serve as a key tool for the algebraic analysis of LWE. In 2015, Albrecht et al. [2] used Gröbner bases to analyze the Arora-Ge polynomial system under the semi-regularity assumption [26,27]. They demonstrate that this method provides an exponential speed-up over the linearization method. For binary-error LWE (each coordinate of error is chosen from $\{0, 1\}$) with a quasi-linear sample size m (relative to the dimension n), it enables a subexponential-time solution. More recently, in Eurocrypt 2024, Steiner [41] proved that the Arora-Ge polynomial system is in generic coordinates, allowing for a proven complexity estimate of the Gröbner basis method. Moreover, the author formulated a conjecture on the degree of regularity for the system, leading to a "lowest achievable complexity estimate". Later, Cueto Noval et al. [21] extended their methodology for the Regular Syndrome Decoding (RSD) problem to a variant of LWE, where each error coordinate lies in a fixed set of constant size. They proved that, with high probability over $\mathbb{Z}_q$, when given $m \geq \binom{n+d-1}{d}$ samples, the Arora-Ge polynomial system achieves a degree of regularity equal to d, where d is the degree of the Arora-Ge polynomial. Based on this result, their improved analysis yields a complexity that is independent of the LWE modulus q and can achieve polynomial-time complexity when a sufficient number of samples are available.

The second approach is **dual attacks**, first introduced by Micciancio and Regev [31]. Given a pair $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, the attacker needs to decide whether $(\mathbf{A}, \mathbf{b})$ was sampled uniformly at random or is of the form $\mathbf{b} = \mathbf{As} + \mathbf{e} \mod q$, where $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^m$ is a short vector. This attack exploits the structure

of the lattice

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n,\ \mathbf{A}\mathbf{x} = \mathbf{y} \mod q\}$$

and its orthogonal lattice

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}^{\top}\mathbf{x} = \mathbf{0} \mod q\},$$

where $\mathbb{Z}$ denotes the integer ring. The attack proceeds by computing many short vectors $\mathbf{x}$ from $\Lambda_q^{\perp}(\mathbf{A})$ and testing the inner products $\langle \mathbf{x}, \mathbf{b} \rangle$. For the LWE case, it tends to be biased toward small values; for the uniform case, it is uniformly distributed over $\mathbb{Z}_q$. This comes from the fact that $\langle \mathbf{x}, \mathbf{b} \rangle = \langle \mathbf{x}, \mathbf{A}\mathbf{s} + \mathbf{e} \rangle = \langle \mathbf{x}, \mathbf{e} \rangle \mod q$.

In Eurocrypt 2017, this foundational idea was developed by Albrecht [1]. The author splits the matrix $\mathbf{A}$ and the secret $\mathbf{s}$ into $[\mathbf{A}_{\text{enu}} \parallel \mathbf{A}_{\text{lat}}]$ and $\begin{pmatrix} \mathbf{s}_{\text{enu}} \\ \mathbf{s}_{\text{lat}} \end{pmatrix}$, respectively. Lattice reduction algorithms are then used to find short vectors of the form $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ satisfying $\mathbf{A}_{\text{lat}}^{\top}\mathbf{x} = \mathbf{y} \mod q$. The attack proceeds by guessing $\mathbf{s}_{\text{enu}}$ and testing whether $\langle \mathbf{x}, \mathbf{b} \rangle - \langle \mathbf{y}_{\text{enu}}, \mathbf{s}_{\text{enu}} \rangle \mod q$ is small, where $\mathbf{y}_{\text{enu}} = \mathbf{A}_{\text{enu}}^{\top}\mathbf{x} \mod q$.

The evolution of dual attacks has seen successive improvements in efficiency. Espitau et al. [24] accelerated the guessing step by exploiting the recursive structure of the search space. Later, Guo and Johansson [28] used the Fast Fourier Transform (FFT) to search only a portion of the bits of $\mathbf{s}_{\text{enu}}$, which significantly reduces the search space. MATZOV [30] further replaced this search technique with "modulus-switching", which reduces the modulus from q to a much smaller p at the expense of a controlled increase in noise. They partition the matrix into

$$[\mathbf{A}_{\text{enu}} \parallel \mathbf{A}_{\text{fft}} \parallel \mathbf{A}_{\text{lat}}] \in \mathbb{Z}_q^{m \times n_{\text{enu}}} \times \mathbb{Z}_q^{m \times n_{\text{fft}}} \times \mathbb{Z}_q^{m \times n_{\text{lat}}}$$

with $n_{\text{enu}} + n_{\text{fft}} + n_{\text{lat}} = n$, and correspondingly $\mathbf{s} = \begin{pmatrix} \mathbf{s}_{\text{enu}} \\ \mathbf{s}_{\text{fft}} \\ \mathbf{s}_{\text{lat}} \end{pmatrix}$.

The algorithm operates in two phases. First, lattice reduction algorithms are used to generate a set ζ of short vectors $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ satisfying $\mathbf{A}_{\text{lat}}^{\top}\mathbf{x} = \mathbf{y} \mod q$. These short vectors are then used to generate a new LWE sample of reduced dimension:

$$(\mathbf{a}', b') = \left(\mathbf{A}_{\text{fft}}^{\top}\mathbf{x},\ \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\text{enu}}\mathbf{s}_{\text{enu}} \rangle\right),$$

with secret $\mathbf{s}_{\text{fft}}$ and error $e' = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\text{lat}} \rangle \mod q$.

To distinguish the correct guess of $\mathbf{s}_{\text{enu}}$, an evaluation function

$$E(\tilde{\mathbf{s}}_{\text{enu}}) = \max_{\mathbf{z} \in \mathbb{Z}_q^{n_{\text{fft}}}} \sum_{(\mathbf{x},\mathbf{y}) \in \zeta} \cos\left(\tfrac{2\pi}{q}\left(\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\text{enu}}\tilde{\mathbf{s}}_{\text{enu}} \rangle - \langle \mathbf{A}_{\text{fft}}^{\top}\mathbf{x}, \mathbf{z} \rangle\right)\right)$$

is introduced. For the correct guess, the inner expression $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\text{enu}}\mathbf{s}_{\text{enu}} \rangle - \langle \mathbf{A}_{\text{fft}}^{\top}\mathbf{x}, \mathbf{s}_{\text{fft}} \rangle$ is biased toward zero, making the sum large; for an incorrect guess

$\tilde{\mathbf{s}}_{\text{enu}} \neq \mathbf{s}_{\text{enu}}$ and any $\mathbf{z} \in \mathbb{Z}_q^{n_{\text{fft}}}$, the $\langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\text{enu}}\tilde{\mathbf{s}}_{\text{enu}} \rangle - \langle \mathbf{A}_{\text{fft}}^\top \mathbf{x}, \mathbf{z} \rangle$ is uniformly distributed over $\mathbb{Z}_q$, yielding a small sum. In practice, a threshold T is set, and a candidate $\tilde{\mathbf{s}}_{\text{enu}}$ is accepted if $E(\tilde{\mathbf{s}}_{\text{enu}}) > T$.

The evaluation can be accelerated via an FFT over $\mathbb{Z}_q^{n_{\text{fft}}}$. To further lower the FFT cost, modulus-switching reduces the modulus to a smaller $p < q$, shrinking the FFT domain to $\mathbb{Z}_p^{n_{\text{fft}}}$. The transformed evaluation function becomes

$$E'(\tilde{\mathbf{s}}_{\text{enu}}) = \max_{\mathbf{z} \in \mathbb{Z}_p^{n_{\text{fft}}}} \sum_{(\mathbf{x},\mathbf{y}) \in \zeta} \cos\Big(\tfrac{2\pi}{p} \big(\tfrac{p}{q} \langle \mathbf{x}, \mathbf{b} - \mathbf{A}_{\text{enu}}\tilde{\mathbf{s}}_{\text{enu}} \rangle - \langle \lfloor \tfrac{p}{q} \mathbf{A}_{\text{fft}}^\top \mathbf{x} \rceil, \mathbf{z} \rangle \big) \Big),$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. The function output still remains significantly larger for the correct $\mathbf{s}_{\text{enu}}$ than for incorrect candidates, preserving the distinguishing advantage.

Later, Ducas and Pulles [22] pointed out that the works of Guo and MATZOV are questionable due to an independence assumption, showing that their claimed parameters for KYBER-512 are in a contradictory regime. In response to this concern, Pouly and Shen [34] provided a provable dual attack on LWE in Eurocrypt 2024 without relying on any assumptions. Qu and Xu [35] followed in Asiacrypt 2025 with another provable dual attack that incorporated the modulus switching technique. In Crypto 2025, Carrier et al. [18] replaced modulus switching with an efficient polar-code decoding algorithm and got rid of the questionable independence assumption in their analysis. Their work provides complexity estimates for CRYSTALS-KYBER that fall below the NIST security requirements.

On the other hand, current research on the algebraic analysis of LWE reveals its limitations in practical scenarios—particularly when the number of available samples is limited (i.e., $m = n$)—where its effectiveness falls short of that achieved by lattice attacks. In several cryptographic areas beyond LWE, algebraic methods have been shown to be highly effective. Representative examples include multivariate cryptography [12,13,14,36], code-based cryptography, in particular the Rank syndrome decoding [4,6,7,25] and Regular syndrome decoding problems [15,21], and Arithmetization-Oriented primitives [9,43,44].

Moreover, nearly all existing algebraic approaches for LWE build upon the foundational framework introduced by Arora and Ge, highlighting the need for further methodological advances in this direction.

1.1 Contributions

A central question arising from the progress in dual attacks is whether they can enhance algebraic analysis of LWE. Algebraically, dual lattice reduction induces a trade-off: it reduces the number of variables in the subsequent polynomial system but simultaneously amplifies the noise. This amplification presents a significant challenge for classical algebraic methods, such as directly solving the Arora-Ge system via Gröbner bases.

In this paper, we answer this question affirmatively by introducing a new algebraic algorithm that integrates dual lattice reduction into algebraic analysis.

Our approach proceeds in two stages. First, we find short vectors from a lattice to generate a large set of dimension-reduced LWE samples. Second, instead of applying Gröbner basis methods to the classic Arora-Ge system, we propose a new polynomial construction that explicitly leverages the distribution of the error and then solve the resulting system using a resultant-based method. Our main contributions are as follows.

1. **A New Two-Stage Algorithm for LWE Integrating Dual Lattice Reduction into Algebraic Analysis.** We develop a two-stage algorithm that combines dual lattice reduction with algebraic cryptanalysis of LWE. The first stage involves applying dual lattice reduction to transform the original LWE samples into a large number of samples with reduced dimension. The second stage solves this dimension-reduced LWE using a method different from Gröbner bases and the linearization method. The key innovation of our algorithm is its explicit design to leverage the *dimension-noise trade-off* induced by the first stage. Rather than treating the reduction and solving stages as independent, our algorithm provides a methodology to algebraically exploit this trade-off, establishing a new avenue for enhancing algebraic analysis with lattice-based methods. To the best of our knowledge, such a two-stage algebraic algorithm for LWE has not appeared before.
2. **A New Polynomial System Construction and Resultant-Based Solving.** To overcome the limitations of applying Gröbner bases directly to the Arora-Ge system after dual lattice reduction, we introduce two interconnected algebraic advances.
   First, we design a new polynomial system tailored to the dimension-reduced LWE. The classical Arora-Ge construction requires the error to be bounded within an interval smaller than the modulus q, a restriction that severely limits the achievable dimension reduction in the first stage. Our new polynomial system explicitly exploits the distribution of the amplified error after the first stage, thereby enabling more effective use of the variable reduction obtained from the dual lattice reduction. We have validated the effectiveness of the proposed polynomial system under several different parameter sets through small-scale experiments (see Section 3.2.2).
   Second, we solve the constructed system using a resultant-based method. Although error amplification leads to a substantial growth in the total degree of the polynomials, the complexity of the resultant method is governed mainly by the number of variables and the partial degree of each individual variable. Because every coordinate of the secret $\mathbf{s}$ is bounded, we can impose a tight, explicit bound on the partial degree of each variable. This controllability constitutes a key advantage that our method actively exploits. We implemented the resultant-based solving procedure for the newly constructed system in Magma and verified it on small-scale parameter sets [1].
   For CRYSTALS-KYBER, a comparison between our two-stage algorithm and the "lowest achievable complexity estimates" from prior work [41] is

[1] https://github.com/latticewalker/Algebraic-Analysis-for-LWE.

presented in Table 2. Although our complexity estimates remain above the NIST security levels (143/207/272 bits), they represent a significant advance over prior algebraic analysis. Compared to the "lowest achievable" Gröbner-basis estimates in [41], our approach reduces the complexity by 860/621/698 bits for KYBER-512/768/1024, respectively.

3. **An Independent Contribution of the Resultant-Based Method for the Arora-Ge System.** As a separate contribution, we also show that the resultant-based method serves as a powerful algebraic solver for LWE. For the small-secret setting, applying the resultant-based method directly to the Arora-Ge system yields a provable complexity estimate that achieves an exponential speed-up over the proven bounds of Steiner [41] (Eurocrypt 2024). Concretely, for CRYSTALS-KYBER it delivers a provable complexity bound that is significantly tighter than the proven Gröbner-basis estimates reported in [41] (see Table 3 for the comparison).
4. **Integrating Hints into Our Two-Stage Algorithm.** We provide a method for incorporating various forms of side information—perfect hints, modular hints, and approximate hints—into our algorithm. This extension enhances practical applicability when partial knowledge is available and further reduces the overall complexity.

We stress that the aim of this work is not to contest the established security of LWE-based schemes, but to explore the possibility of integrating the techniques from dual attacks into algebraic analysis for LWE. Our results demonstrate the potential of this hybrid approach to refine the landscape of algebraic cryptanalysis for LWE.

1.2 Organization

The notation and foundational concepts used in this paper are introduced in Section 2. The overall two-stage algebraic algorithm is presented in Section 3: an analysis of dual lattice reduction is provided in Section 3.1; different polynomial constructions, along with solving methods, are described in Section 3.2. The independent contribution of the resultant-based method to the Arora-Ge system is demonstrated in Section 4. The incorporation of various forms of side information into the proposed algorithm is discussed in Section 5. Finally, the conclusions and some open questions are presented in Section 6.

2 Preliminaries

In this section, we present the notation and definitions used throughout the paper.

Vectors and matrices are denoted by bold lowercase and bold uppercase letters, respectively, e.g. $\mathbf{a}$ and $\mathbf{A}$. By default, all vectors in this paper are column vectors. A row vector is explicitly represented as the transpose of a column vector, e.g. $\mathbf{a}^\top$. For an integer $n \geq 1$, consider a vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$. Its

i-th entry is denoted by a_i for $i \in [\![1, n]\!]$, where $[\![a, b]\!]$ denotes the set of integers from a to b, inclusive.

Let $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ denote the set of natural numbers. For a set Ω, $|\Omega|$ denotes its cardinality. The ring of integers modulo q is denoted by $\mathbb{Z}_q$, which forms a finite field if q is prime.

For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^n$, their inner product is defined as $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i b_i$, which is equivalently expressed as $\mathbf{a}^\top \mathbf{b}$. The Euclidean norm of a vector $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{Z}_q^n$ is defined as $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} \min_{j \in \mathbb{Z}} (x_i + jq)^2}$.

Let χ be a probability distribution over $\mathbb{Z}_q$. The notation $\mathbf{a} \leftarrow \chi^n$ indicates that the vector $\mathbf{a}$ is sampled from χ^n (i.e., each entry is drawn independently from χ). The abbreviation *i.i.d.* stands for independent and identically distributed.

2.1 Lattice and Lattice Reduction

In this subsection, we will introduce some results about lattices and lattice reduction algorithms.

Definition 1. *A lattice is a discrete additive subgroup of $\mathbb{R}^m$. Specifically, given n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in \mathbb{R}^m$, the set*

$$\Lambda = \{\mathbf{v} \mid \mathbf{v} = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + \cdots + c_n\mathbf{b}_n, c_1, c_2, ..., c_n \in \mathbb{Z}\}$$

is called an m-dimensional lattice. The n linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ are called a basis of the lattice Λ and the integer n is called the rank of Λ. A lattice Λ is called a full-rank lattice if $n = m$.

Let $\mathbf{B}$ be the $m \times n$ matrix consisting of column vectors $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$. The determinant of the lattice Λ is defined by $\det(\Lambda) = \sqrt{\det(\mathbf{B}^\top \mathbf{B})}$, where $\mathbf{B}^\top$ is the transpose of $\mathbf{B}$. The most commonly used heuristic for lattice algorithms is the so-called Gaussian Heuristic (GH), which is extensively verified by experiments.

Definition 2 (Gaussian Heuristic [33]). *Let Λ be a full-rank lattice in $\mathbb{R}^m$, and C a measurable subset of $\mathbb{R}^m$ with volume $\mathrm{vol}(C)$. The Gaussian Heuristic "predicts" that the number of points of $\Lambda \cap C$ is roughly $\mathrm{vol}(C)/\det(\Lambda)$.*

Let $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ be a lattice basis of the full rank lattice Λ. Its Gram-Schmidt orthogonalized basis $\mathbf{B}^* = (\mathbf{b}_1^*, \mathbf{b}_2^*, \dots, \mathbf{b}_n^*)$ is defined as $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$, where $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$ for $1 \le j < i \le n$.

Lattice reduction is the process of taking a basis for a given Λ and finding subsequent bases of Λ with shorter and closer to orthogonal vectors. The BKZ algorithm [38,19] takes as input a lattice basis of Λ and a block size β, then outputs a BKZ-β reduced basis of Λ. Sieving algorithms build databases of lattice vectors, the sizes of which are exponential in the lattice dimension. In the simplest sieves, it is checked whether the sums or differences of any pair of database vectors is shorter than one of the summands or differands.

The geometric series assumption (GSA) is another widely used heuristic assumption in lattice-based cryptanalysis, and it is reasonably accurate for $\beta > 50$ and $\beta \ll n$ [33,19,45].

Definition 3 (Geometric Series Assumption [39]). *Let* $\mathbf{B}$ *be a BKZ-*β *reduced basis with sufficiently large* β*, then* $\|\mathbf{b}_i^*\|/\|\mathbf{b}_{i-1}^*\| \approx \delta(\beta)^{-2}$*, where* $\delta(\beta) = \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta}\right)^{\frac{1}{2(\beta-1)}}$ *is the root-Hermite factor.*

2.2 Learning with Errors

Definition 4 (LWE [37]). *Let* q *be a prime and* n, m *be positive integers, let* χ *be a probability distribution on* $\mathbb{Z}$ *and let* $\mathbf{s} \in \mathbb{Z}_q^n$ *be a secret vector. A pair*

$$(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \mod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$$

is called an LWE instance, where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ *is chosen uniformly at random and* $\mathbf{e} \leftarrow \chi^m$*.*

The LWE problem has two versions:

- **Decision-LWE**: Given a pair $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, which is drawn either from the LWE distribution $(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \mod q)$ or from the uniform distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, determine which distribution it comes from.
- **Search-LWE**: Given an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \mod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, recover the secret vector $\mathbf{s}$.

In most practical LWE-based cryptosystems, the secret and error vectors are both small, where their coordinates are chosen from a small, fixed interval. **We adopt and consistently rely on this common setting throughout this paper.**

In particular, CRYSTALS-KYBER employs a centered binomial distribution for both the secret and error coordinates, which is defined as follows.

Definition 5 (Centered Binomial Distribution). *The centered binomial distribution* B_η *with parameter* $\eta \in [\![0, \frac{q-1}{2}]\!]$ *is defined as* $B_\eta \sim \sum_{i=1}^{\eta}(X_i - Y_i)$*, where* X_i *and* Y_i *are i.i.d. uniform over* $\{0, 1\}$*.*

It is well-known that B_η has mean 0 and standard deviation $\sigma = \sqrt{\frac{\eta}{2}}$. The probability that $B_\eta = i$ is given by $\mathbb{P}(B_\eta = i) = 2^{-2\eta}\binom{2\eta}{\eta+i}$ for $i \in [\![-\eta, \eta]\!]$. In the KYBER schemes, $\eta = 3$ for KYBER-512 and $\eta = 2$ for KYBER-768/1024.

2.3 Modular Gaussian Distribution

Let $\sigma > 0$. For all $x \in \mathbb{R}$, the density of the centered Gaussian distribution with standard deviation σ is defined as $\rho_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$.

Definition 6 (Modular Gaussian Distribution). *The Modular Gaussian distribution modulo* q *is a probability distribution over* $\mathbb{Z}_q$*, defined by the probability mass function*

$$\rho_{q,\sigma}(k) = \frac{1}{C_{q,\sigma}} \sum_{x \in k + q\mathbb{Z}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{1}$$

where $C_{q,\sigma} = \sum_{k \in \mathbb{Z}} \exp\left(-\frac{k^2}{2\sigma^2}\right)$ *is a normalization constant.*

For a random variable X following a normal distribution with mean 0 and variance σ^2, one can utilize the well-known Mills' inequality

$$\mathbb{P}\big[\,|X| > t\sigma\,\big] \leq \frac{2}{t\sqrt{2\pi}}\,\exp\Big(-\frac{t^2}{2}\Big). \tag{2}$$

In Section 3.1, we utilize Eq. (2) to estimate bound on the error coordinates after dual lattice reduction.

2.4 Gröbner Bases

In this subsection, we give a brief introduction to the theory of Gröbner bases. For more details, refer to [20].

Let $\mathbb{K}$ be a field and $\mathbb{K}[x_1, x_2, \ldots, x_n]$ the polynomial ring over $\mathbb{K}$ with indeterminates $x_1, x_2, \ldots, x_n$. An expression of the form $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ with each $k_i \in \mathbb{N}$ is called a *term*. The set of all such terms is written as $T(x_1, x_2, \ldots, x_n)$. A *monomial* is a term multiplied by a nonzero coefficient $a \in \mathbb{K}$, i.e., $a x_1^{k_1} \cdots x_n^{k_n}$.

A *term order* on $T(x_1, x_2, \ldots, x_n)$ is a linear order $\preceq$ satisfying:

- For every term $X \in T(x_1, x_2, \ldots, x_n)$, $1 \preceq X$.
- For all terms $Y, X_1, X_2 \in T(x_1, x_2, \ldots, x_n)$, $X_1 \preceq X_2$ implies $YX_1 \preceq YX_2$.

For a polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$, its *leading term* with respect to $\preceq$, denoted by $\mathrm{LT}_{\preceq}(f)$, is the largest term appearing in f. For an ideal $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$, its *initial ideal* is defined as

$$\mathrm{in}_{\preceq}(I) = \langle\, \mathrm{LT}_{\preceq}(f) \mid f \in I \,\rangle.$$

Definition 7 (Gröbner basis). *Let $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ be an ideal and $\preceq$ a term order. A finite subset $\mathcal{G} = \{g_1, \ldots, g_m\} \subseteq I$ is said to be a* Gröbner basis *of I with respect to $\preceq$ if $0 \notin \mathcal{G}$ and*

$$\mathrm{in}_{\preceq}(I) = \langle \mathrm{LT}_{\preceq}(g_1), \ldots, \mathrm{LT}_{\preceq}(g_m) \rangle.$$

For a given polynomial system $\mathcal{F} = \{f_1, \ldots, f_m\} \subseteq \mathbb{K}[x_1, \ldots, x_n]$ and a given term order $\preceq$, the computation of a Gröbner basis of the ideal $\langle \mathcal{F} \rangle$ is often performed via linear algebra on large structured matrices, known as Macaulay matrices. Recall that a polynomial is said to be *homogeneous* if all its terms have the same degree. For a positive integer d, a *homogeneous Macaulay matrix* M_d is constructed as follows: its columns are indexed by all terms of degree d, ordered from left to right with respect to $\preceq$. The rows are indexed by all polynomials of the form $s \cdot f_i$, where $s \in T(x_1, x_2, \ldots, x_n)$ is a term satisfying $\deg(s \cdot f_i) = d$. The entry of a row $s \cdot f_i$ in column t is the coefficient of $s \cdot f_i$ at the term t. For an inhomogeneous system, one considers instead the matrix $M_{\leq d}$, whose columns are indexed by all terms of degree less than or equal to d and whose rows correspond to all $s \cdot f_i$ with $\deg(s \cdot f_i) \leq d$.

A key parameter governing the complexity of computing a Gröbner basis is the solving degree.

Definition 8 (Solving degree, [16]). *Let $\mathcal{F} = \{f_1, \ldots, f_m\} \subseteq \mathbb{K}[x_1, \ldots, x_n]$ and let $\preceq$ be a term order. The solving degree of $\mathcal{F}$, denoted by $\mathrm{sd}_{\preceq}(\mathcal{F})$, is the least integer d such that Gaussian elimination on the Macaulay matrix $M_{\leq d}$ produces a Gröbner basis of $\langle \mathcal{F} \rangle$ with respect to $\preceq$.*

In this paper, we use the solving degree with respect to the degree reverse lexicographic (DRL) order.

Definition 9 (Degree Reverse Lex Order [20]). *Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n) \in \mathbb{N}^n$. We say*

$$x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \preceq_{\mathrm{DRL}} x_1^{\beta_1} x_2^{\beta_2} \cdots x_n^{\beta_n},$$

if $\sum_{i=1}^n \alpha_i < \sum_{i=1}^n \beta_i$, or $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ and the right-most nonzero entry of $\boldsymbol{\alpha} - \boldsymbol{\beta}$ is positive.

For a system $\mathcal{F}$ of m polynomials in n variables with solving degree $\mathrm{sd} = \mathrm{sd}_{\mathrm{DRL}}(\mathcal{F})$, the cost of computing its Gröbner basis with respect to the DRL order is bounded by

$$O\left(m \cdot \mathrm{sd} \cdot \binom{n + \mathrm{sd} - 1}{\mathrm{sd}}^{\omega}\right), \tag{3}$$

where $2 \leq \omega < 3$ is the linear algebra exponent[2].

Denote the highest component of f as f^{top}. For a given finite set of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\} \subseteq \mathbb{K}[x_1, x_2, \ldots, x_n]$, define $\mathcal{F}^{\mathrm{top}} = \{f_1^{\mathrm{top}}, \ldots, f_m^{\mathrm{top}}\}$.

Another key parameter that characterizes the complexity of solving a polynomial system is the degree of regularity.

Definition 10 (Degree of Regularity, [8, Definition 4]). *Let $\mathbb{K}$ be a field and $\mathcal{F} \subseteq \mathbb{K}[x_1, \ldots, x_n]$. Assume that $\langle \mathcal{F}^{\mathrm{top}} \rangle_d = P_d$ for some integer $d \geq 0$, where $\langle \mathcal{F}^{\mathrm{top}} \rangle_d$ and P_d are the sets of homogeneous polynomials of degree d in the ideal $\langle \mathcal{F}^{\mathrm{top}} \rangle$ and $\mathbb{K}[x_1, \ldots, x_n]$, respectively. The degree of regularity is defined as*

$$d_{\mathrm{reg}}(\mathcal{F}) = \min\left\{d \geq 0 \mid \langle \mathcal{F}^{\mathrm{top}} \rangle_d = P_d\right\}.$$

In [41], the author employs a conjecture that sets a bound on the solving degree in relation to the degree of regularity.

Conjecture 1 ([41, Conjecture 25]). Let q be a prime number and let $\mathcal{F}_{\mathrm{AG}} \subseteq \mathbb{Z}_q[x_1, \ldots, x_n]$ be the Arora-Ge polynomial system. For the small secret LWE, if each coordinate of the error is chosen from $[\![-T, T]\!]$, then

$$\mathrm{sd}_{\mathrm{DRL}}(\mathcal{F}_{\mathrm{AG}}) \leq d_{\mathrm{reg}}(\mathcal{F}_{\mathrm{AG}}) + D - 1,$$

where D is the degree of polynomials in $\mathcal{F}_{\mathrm{AG}}$.

Consequently, the complexity of computing a Gröbner basis of $\langle \mathcal{F}_{\mathrm{AG}} \rangle$ with respect to the DRL order can be estimated by substituting the bound $d_{\mathrm{reg}}(\mathcal{F}_{\mathrm{AG}}) + 2T$ for solving degree in Eq. (3).

[2] We set $\omega = 2$ to maintain consistency with the complexity analysis used in [41].

2.5 Resultant

The resultant is a powerful tool for solving polynomial systems. Recent studies on Arithmetization-Oriented (AO) primitives have demonstrated that, in certain settings, resultant-based solvers can outperform traditional Gröbner basis methods [9,44,43]. Motivated by these findings, we adopt a resultant-based method for solving the new polynomial system in this paper.

Definition 11 (Resultant). *Let q be a prime number and let $f(\mathbf{x}, y), g(\mathbf{x}, y) \in \mathbb{Z}_q[\mathbf{x}, y]$ with $\mathbf{x} = (x_1, \ldots, x_s)$. The resultant of $f(\mathbf{x}, y)$ and $g(\mathbf{x}, y)$ with respect to the indeterminate y, denoted by $R(f, g, y)$, is defined as the determinant of the Sylvester matrix of $f(\mathbf{x}, y)$ and $g(\mathbf{x}, y)$ when considered as polynomials in the single indeterminate y. That is, if $f(\mathbf{x}, y) = \sum_{i=0}^{u} f_i y^i$ and $g(\mathbf{x}, y) = \sum_{i=0}^{v} g_i y^i$, where $f_i, g_i \in \mathbb{Z}_q[\mathbf{x}]$, then*

$$R(f,g,y) = \begin{vmatrix} f_u & f_{u-1} & \cdots & f_0 & & & \\ & f_u & f_{u-1} & \cdots & f_0 & & \\ & & \ddots & \ddots & & \ddots & \\ & & & f_u & f_{u-1} & \cdots & f_0 \\ g_v & g_{v-1} & \cdots & \cdots & g_0 & & \\ & g_v & g_{v-1} & \cdots & \cdots & g_0 & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & g_v & g_{v-1} & \cdots & g_0 \end{vmatrix}$$

It is well-known that the resultant of two polynomials is non-zero if and only if they have no common factors other than non-zero constants. In this case, the resultant yields a new polynomial $h(\mathbf{x})$, such that if $(\mathbf{x}_0, y_0)$ is a common root of both $f(\mathbf{x}, y)$ and $g(\mathbf{x}, y)$, then $h(\mathbf{x}_0) = 0$. In this way, we can remove one variable from two polynomials while retaining information about the roots of the original polynomials. For a system of ℓ polynomials in s variables, we can repeatedly compute resultants of the polynomials until we obtain a univariate polynomial. Solving for the roots of the univariate polynomial and repeatedly substituting them back, we can derive the roots that the polynomials have in common.

Since each resultant computation involves repeated multiplications of multivariate polynomials, the complexity of such multiplications is a basic ingredient in our later analysis. We therefore recall the following lemma.

Lemma 1 (Complexity of multivariate polynomial multiplication, [9]). *Let $f, g \in \mathbb{Z}_q[x_1, \ldots, x_n]$ be two polynomials with $\deg_{x_i}(f) = \alpha_i$ and $\deg_{x_i}(g) = \beta_i$ for $i = 1, \ldots, n$, where $\deg_{x_i}(f)$ is the partial degree of f with respect to x_i. Using the Kronecker substitution technique [32], their product $h = f \cdot g$ can be computed with $\mathcal{M}(\prod_{i=1}^{n}(\alpha_i + \beta_i + 1))$ operations in $\mathbb{Z}_q$, where $\mathcal{M}(d) = \mathcal{O}(d \log d \log\log d)$. The resulting polynomial satisfies $\deg_{x_i}(h) = \alpha_i + \beta_i$ and has at most $\prod_{i=1}^{n}(\alpha_i + \beta_i + 1)$ non-zero coefficients.*

Lemma 2 provides an estimation for finding the roots of a univariate polynomial over $\mathbb{Z}_q$.

Lemma 2 (Complexity of univariate polynomial root finding, [11, Section 3.1]). *Let $f(x) \in \mathbb{Z}_q[x]$ be a univariate polynomial of degree d. Using an FFT-based algorithm, all roots of $f(x)$ in $\mathbb{Z}_q$ can be found in*

$$\mathcal{O}\big(d \log(d)\,(\log(d) + \log(q))\,\log(\log(d))\big)$$

operations over $\mathbb{Z}_q$.

3 A Two-Stage Algebraic Algorithm for LWE

Recent advances in dual attacks have demonstrated their effectiveness for LWE. In contrast, existing algebraic attacks on LWE are far from satisfactory. This gap motivates us to investigate whether integrating dual lattice reduction into algebraic analysis can yield improvements.

In this section, we give a novel, two-stage hybrid algorithm that systematically integrates dual lattice reduction into algebraic cryptanalysis for LWE. In the first stage, short vectors are extracted from a lattice to produce a large set of dimension-reduced LWE samples. This transformation offers an algebraic advantage: it significantly decreases the number of unknown variables while simultaneously providing abundant samples. However, it also introduces a drawback: the noise is amplified, increasing the magnitude of the error. Consequently, applying the classic Arora-Ge method directly to those transformed samples is ineffective since it merely enumerates all possible bounded error values without exploiting their distribution, which naturally requires that the size of the error range be smaller than the modulus q. Moreover, the high total degree of the resulting polynomial system also makes Gröbner basis computations impractical. To overcome these limitations, in the second stage we propose a novel polynomial construction that explicitly exploits the distribution of the amplified error—in fact, the probability that it equals zero. This is achieved by selecting a random set of K distinct coordinates and multiplying them together (the details are provided in Section 3.2.2). This approach allows us to build the polynomial system without requiring that the size of the error range be smaller than the modulus q, which permits a larger dimension in the dual lattice reduction stage. This new system is solved using a resultant-based method, whose complexity is governed mainly by the number of variables and the partial degree in each variable—the latter being a parameter that is kept small and tightly bounded in our construction.

Our proposed two-stage algebraic algorithm is outlined in Algorithm 1.

Steps 1-3 constitute the first stage, dual lattice reduction, whose detailed analysis is provided in Section 3.1.

Step 4 forms the second stage, polynomial construction and solving, which is described in Section 3.2.

Once $\mathbf{s}_{\text{solve}}$ is recovered, the remaining secret coordinates can be obtained by solving a smaller sub-LWE instance (Step 5). We do not prescribe a specific subroutine for this final step, as its cost is dominated by the preceding stages of Algorithm 1.

Algorithm 1 A New Two-Stage Algebraic Algorithm for LWE

Input: An LWE instance $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, parameters n_{lat}, $n_{\text{solve}} = n - n_{\text{lat}}$, β_{sieve}, β_{bkz}.
Output: Secret vector $\mathbf{s}$.

1: Partition $\mathbf{A}$ column-wise into $[\mathbf{A}_{\text{solve}} \parallel \mathbf{A}_{\text{lat}}]$, where $\mathbf{A}_{\text{solve}} \in \mathbb{Z}_q^{m \times n_{\text{solve}}}, \mathbf{A}_{\text{lat}} \in \mathbb{Z}_q^{m \times n_{\text{lat}}}$, correspondingly split $\mathbf{s} = \begin{pmatrix} \mathbf{s}_{\text{solve}} \\ \mathbf{s}_{\text{lat}} \end{pmatrix}$.
2: $\zeta \leftarrow \text{ShortVectorSampling}(\mathbf{A}_{\text{lat}}, \beta_{\text{bkz}}, \beta_{\text{sieve}})$. ▷ $|\zeta| = N \approx (4/3)^{\beta_{\text{sieve}}/2}$
3: $\Omega \leftarrow \text{GenerateNewSamples}(\zeta, \mathbf{A}_{\text{solve}}, \mathbf{b})$.
4: $\mathbf{s}_{\text{solve}} \leftarrow \text{PolynomialSystemSolving}(\Omega)$.
5: $\mathbf{s}_{\text{lat}} \leftarrow \text{SolveSubLWE}(\mathbf{A}_{\text{lat}}, \mathbf{b}, \mathbf{s}_{\text{solve}})$. ▷ Recover remaining secret components

Return: $\mathbf{s} = (\mathbf{s}_{\text{solve}}^\top \parallel \mathbf{s}_{\text{lat}}^\top)^\top$.

In this section, we consider LWE with a small secret and a small error, wherein all coordinates are chosen from a fixed small interval $[\![-T, T]\!]$. This is the setting for most practical LWE-based schemes.

3.1 The First Stage: Constructing Dimension-reduced LWE via Dual Lattice Reduction

The first stage follows the methodology from dual attacks [18,28,30]. Specifically, we first apply lattice reduction algorithms to the lattice generated by the $r \times r$ matrix $\mathbf{B} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{A}_{\text{lat}}^\top & q\mathbf{I}_{n_{\text{lat}}} \end{bmatrix}$, where $r = m + n_{\text{lat}}$. This yields N short vectors of the form $\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ satisfying $\mathbf{A}_{\text{lat}}^\top \mathbf{x} = \mathbf{y} \mod q$ (see Algorithm 2 for details).

Algorithm 2 Short Vectors Sampling [28]

Input: A basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r]$ of a lattice, and integers $2 \le \beta_{\text{bkz}}, \beta_{\text{sieve}} \le r$.
Output: A set ζ of $N_{\text{sieve}}(\beta_{\text{sieve}})$ vectors from the lattice.

1: Run BKZ-β_{bkz} on $\mathbf{B}$ to obtain a reduced basis $\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_r$.
2: Run a sieve in dimension β_{sieve} on the sublattice spanned by $\mathbf{b}'_1, \ldots, \mathbf{b}'_{\beta_{\text{sieve}}}$ and collect the output into a set ζ.

Return: ζ.

The expected number of short vectors given by Algorithm 2 is $N = N_{\text{sieve}}(\beta_{\text{sieve}}) \approx \left(\sqrt{\frac{4}{3}}\right)^{\beta_{\text{sieve}}}$.

Each such vector is then used to construct a new dimension-reduced LWE sample $(\mathbf{a}', b')$ satisfying $b' = \langle \mathbf{a}', \mathbf{s}_{\text{solve}} \rangle + e' \mod q$, where

$$\mathbf{a}' = \mathbf{A}_{\text{solve}}^\top \mathbf{x}, \quad b' = \langle \mathbf{b}, \mathbf{x} \rangle, \quad e' = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\text{lat}} \rangle.$$

The expected length of the vectors produced by Algorithm 2 can be estimated by the following Lemma 3.

Lemma 3 (Length of vectors from Algorithm 2, [18, Lemma 2.8]). *Let Λ be the lattice generated by the $r \times r$ matrix $\mathbf{B}$ in Algorithm 2, and let β_{bkz} and β_{sieve} be the block sizes for BKZ and sieve, respectively. If the Gaussian Heuristic (GH) and the Geometric Series Assumption (GSA) hold for Λ, then Algorithm 2 outputs $N_{\mathrm{sieve}}(\beta_{\mathrm{sieve}}) \approx \left(\sqrt{\frac{4}{3}}\right)^{\beta_{\mathrm{sieve}}}$ vectors of expected length ℓ given by*

$$\ell = \det(\Lambda)^{1/r} \cdot \sqrt{\frac{4}{3}} \cdot \delta(\beta_{\mathrm{sieve}})^{\beta_{\mathrm{sieve}}-1} \cdot \delta(\beta_{\mathrm{bkz}})^{r-\beta_{\mathrm{sieve}}},$$

where $\delta(\beta) = \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta}\right)^{\frac{1}{2(\beta-1)}}$ is the root-Hermite factor.

To finish this subsection, we calculate the probability distribution of e' in Section 3.1.1, and then analyze the complexity of the first stage in Section 3.1.2.

3.1.1 The Probability Distribution of e'

We first invoke the following assumption regarding the distribution of the sampled short vectors.

Assumption 1 ([19,24,30]) *Let ℓ be the expected length of vectors returned by Algorithm 2. Then the coordinates of the returned vectors have approximately independent Gaussian distributions with mean 0 and standard deviation $\ell/\sqrt{r}$, where r is the dimension of the matrix $\mathbf{B}$.*

Now we give the probability distribution of e'.

Lemma 4 (Probability distribution of e'). *Under Assumption 1, the quantity $e' = \langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathrm{lat}} \rangle$ follows a modular Gaussian distribution with mean 0 and variance* [3]

$$\sigma^2 \approx \left(\|\mathbf{e}\|^2 + \|\mathbf{s}_{\mathrm{lat}}\|^2\right) \cdot \frac{\ell^2}{m + n_{\mathrm{lat}}}.$$

Proof. By Assumption 1, the coordinates of $\mathbf{x}$ and $\mathbf{y}$ are approximately independent Gaussians with mean 0 and variance ℓ^2/r, where $r = m + n_{\mathrm{lat}}$. Hence, $\langle \mathbf{x}, \mathbf{e} \rangle$ and $\langle \mathbf{y}, \mathbf{s}_{\mathrm{lat}} \rangle$ are multiple sums of independent Gaussian variables and are themselves modular Gaussian. Therefore, the mean of $\langle \mathbf{x}, \mathbf{e} \rangle + \langle \mathbf{y}, \mathbf{s}_{\mathrm{lat}} \rangle$ is 0 and

[3] In practice, verifying the independence among the coordinates of the vectors returned by Algorithm 2 is impractical. Therefore, in our subsequent analysis, we adopt the variance estimate $\sigma^2 = \left(\|\mathbf{e}\|^2 + \|\mathbf{s}_{\mathrm{lat}}\|^2\right) \cdot \frac{\ell^2}{m+n_{\mathrm{lat}}}$, as employed in [30].

the variance

$$\begin{aligned}\sigma^2 &\approx \mathrm{Var}\Big(\sum_{i=1}^{m} x_i e_i + \sum_{j=1}^{n_{\mathrm{lat}}} y_j (s_{\mathrm{lat}})_j\Big) \\ &= \mathbb{E}\Big(\sum_{i=1}^{m} x_i^2 e_i^2 + \sum_{j=1}^{n_{\mathrm{lat}}} y_j^2 (s_{\mathrm{lat}}^2)_j\Big) \\ &= \left(\|\mathbf{e}\|^2 + \|\mathbf{s}_{\mathrm{lat}}\|^2\right) \cdot \frac{\ell^2}{m + n_{\mathrm{lat}}}\end{aligned}$$

which is the stated formula. □

From the definition of the modular Gaussian distribution (see Definition 6), the probability that $e' = k \mod q$ can be given by Eq. (1). In Section 3.2.2, we will propose a polynomial construction that uses $\mathbb{P}(e' = k)$.

3.1.2 Complexity of the First Stage

The complexity of the first stage comprises two parts:

- **Short vector sampling**, the cost of which is stated in Lemma 5.
- **Generation of new LWE samples**, which costs $\mathcal{O}(\sqrt{\frac{4}{3}}^{\beta_{\mathrm{sieve}}})$.

Lemma 5. *The complexity of Algorithm 2 is*

$$T_{\mathrm{sample}} = T_{\mathrm{BKZ}} + T_{\mathrm{sieve}},$$

where T_{BKZ} and T_{sieve} denote the costs of the BKZ and the sieve, respectively.

In fact, from the subsequent analysis, the first stage is not the dominant cost, so we do not analyze differences among cost models for lattice reduction algorithms. Consequently, we adopt the "core-SVP" cost model [4] (denoted by the **C0** model in [18, Section 5]) for Algorithm 2. This model assumes that a single call to an SVP oracle suffices to reduce a lattice basis and that all lower-order terms in the exponent are negligible. Within this framework, the complexity of Algorithm 2 is given by $\mathcal{O}(2^{0.292\beta})$, where $\beta = \beta_{\mathrm{bkz}} = \beta_{\mathrm{sieve}}$.

3.2 The Second Stage: Polynomial Construction and Solving

After the first stage, we now have $N = N_{\mathrm{sieve}}(\beta_{\mathrm{sieve}})$ new dimension-reduced LWE samples

$$(\mathbf{a}'_i,\ b'_i = \langle \mathbf{a}'_i, \mathbf{s}_{\mathrm{solve}} \rangle + e'_i \mod q), i = 1, 2, \ldots, N,$$

where each coordinate of $\mathbf{s}_{\mathrm{solve}}$ is chosen from the integer interval $[\![-T, T]\!]$ and each e'_i is bounded (with high probability) within $[\![-T', T']\!]$ with $T' = \lceil t\sigma \rceil$. Here,

[4] For a more precise estimate, one can employ the "lattice estimator" of [3].

the parameter t is taken to be sufficiently large so that, with high probability, each e'_i lies within this bounded interval; and

$$\sigma \;=\; \sqrt{\left(\|\mathbf{e}\|^2 + \|\mathbf{s}_{\mathrm{lat}}\|^2\right) \cdot \frac{\ell^2}{m + n_{\mathrm{lat}}}}\;,$$

where ℓ is the expected length of the short vectors produced by Algorithm 2, given by Lemma 3. For convenience, $\mathbf{e}' = (e'_1, e'_2, \ldots, e'_N)$ is called the amplified error vector.

The second stage constructs a polynomial system from the dimension-reduced samples and solves it using two different algebraic approaches. In Section 3.2.1, we construct the polynomial system via the Arora-Ge method (hereafter referred to as the Arora-Ge system) and solve it with Gröbner bases; the corresponding complexity estimate follows the framework established in [41]. In Section 3.2.2, we propose a new polynomial system that exploits the error distribution and solve it using a resultant-based method. The results demonstrate that our new construction, combined with the resultant-based method, yields a more effective synergy with the preceding stage.

3.2.1 Construction of the Arora-Ge System and Solving by the Gröbner Basis Method

Although the first stage can produce substantially dimension-reduced samples, directly applying the Gröbner basis method to solve the resulting Arora-Ge system proves ineffective.

More precisely, by applying the polynomial construction introduced by Arora and Ge [5], the Arora-Ge system is defined as

$$\mathcal{F}_{\mathrm{AG}} \triangleq \left\{ f_1(b'_1 - \langle \mathbf{a}'_1, \mathbf{z} \rangle), \ldots, f_N(b'_N - \langle \mathbf{a}'_N, \mathbf{z} \rangle) \right\},$$

where each polynomial f_i is obtained by substituting the linear form $b'_i - \langle \mathbf{a}'_i, \mathbf{z} \rangle$ into the univariate polynomial $f(X) = X \prod_{j=1}^{\lceil t\sigma \rceil} (X + j)(X - j)$, which has a degree of $D = 2\lceil t\sigma \rceil + 1$. Since all coordinates of $\mathbf{s}$ are chosen from $[\![-T, T]\!]$, the degree in each variable s_i can be controlled by imposing the constraint

$$s_i \prod_{j=1}^{T} (s_i + j)(s_i - j) = 0, \qquad i = 1, 2, \ldots, n_{\mathrm{solve}}.$$

Consequently, the individual degree with respect to each variable can be kept strictly less than $2T + 1$.

Among the three distinct complexity-estimation methods presented in [41], we employ the "lowest achievable complexity estimate" for our analysis. This choice provides a fairer and more demanding comparison. Theorem 1 states the complexity of solving $\mathcal{F}_{\mathrm{AG}}$ using the Gröbner basis method under this "lowest estimate".

Theorem 1. *Let q be a prime, and let $m = n \geq 2$ be LWE parameters, with the secret $\mathbf{s}$ and error $\mathbf{e}$ having coordinates chosen from $[\![-T, T]\!]$. After performing dual lattice reduction with parameters n_{lat}, $n_{\text{solve}} = n - n_{\text{lat}}$, β_{bkz}, and β_{sieve}, let $\mathcal{F}_{\text{AG}}$ denote the Arora-Ge system constructed from the resulting N dimension-reduced samples. With a success probability of at least $1 - \epsilon$, one can compute a Gröbner basis for the ideal $\langle \mathcal{F}_{\text{AG}} \rangle$ with respect to the DRL order and its time complexity is* [5]

$$O\left(N \cdot \text{sd} \cdot \binom{n_{\text{solve}} + \text{sd} - 1}{\text{sd}}^{\omega}\right),$$

where $N = \left(\sqrt{4/3}\right)^{\beta_{\text{sieve}}}$, and $\text{sd} = d_{\min} + 2D - 1$. Here, $d_{\min}$ is the smallest integer d satisfying the inequality

$$N \cdot \binom{n_{\text{solve}} + d - 1}{d} \geq \binom{n_{\text{solve}} + D + d - 1}{D + d}, \tag{4}$$

with $D = \min\{2\lceil t\sigma \rceil + 1, 2T \cdot n_{\text{solve}}\}$. The standard deviation σ of the amplified error is given by

$$\sigma = \sqrt{\left(\|\mathbf{e}\|^2 + \|\mathbf{s}_{\text{lat}}\|^2\right) \cdot \frac{\ell^2}{m + n_{\text{lat}}}},$$

in which $\ell = \det(\Lambda)^{1/r} \cdot \sqrt{4/3} \cdot \delta(\beta_{\text{sieve}})^{\beta_{\text{sieve}}-1} \cdot \delta(\beta_{\text{bkz}})^{r-\beta_{\text{sieve}}}$ with $r = m + n_{\text{lat}}$. The parameter t is chosen such that the success probability

$$p_{\text{success}} \geq 1 - N \cdot \frac{2}{t\sqrt{2\pi}} \exp(-t^2/2) = 1 - \epsilon.$$

Proof. It immediately follows from Eq. (2) that t is chosen such that every coordinate of the amplified error vector $\mathbf{e}' = (e'_1, e'_2, \ldots, e'_N)$ falls within an interval $[\![-\lceil t\sigma \rceil, \lceil t\sigma \rceil]\!]$ with a probability of at least $1 - \epsilon$. Consequently, the degree of the polynomials in $\mathcal{F}_{\text{AG}}$ is $D = \min\{2\lceil t\sigma \rceil + 1, 2T \cdot n_{\text{solve}}\}$, and so the degree of regularity $d_{\text{reg}}(\mathcal{F}_{\text{AG}}) = d_{\min} + D$. Finally, applying Conjecture 1 provides the upper bound of the solving degree $\text{sd}_{\text{DRL}}(\mathcal{F}_{\text{AG}}) \leq d_{\text{reg}}(\mathcal{F}_{\text{AG}}) + D - 1$. Substituting this bound into Eq. (3) completes the proof. □

We estimate the time complexity for CRYSTALS-KYBER using Theorem 1. To minimize the adverse impact of the dual lattice reduction stage on the subsequent algebraic stage, we set the block-size parameters β_{BKZ} and β_{sieve} to their maximal feasible values, namely $\beta_{\text{bkz}} = \beta_{\text{sieve}} = m + n_{\text{lat}}$. We fix the overall success probability at 0.99. Under this setting, Figure 1 shows that the complexity first increases and then decreases as n_{lat} grows.

Initially, although $n_{\text{solve}} = n - n_{\text{lat}}$ decreases, this reduction is dominated by the rapid growth of D, which causes the overall complexity to increase. Once

[5] To recover the true secret from this basis, an additional step—such as the FGLM algorithm or the Gröbner Walk—is required to convert the DRL basis into a lexicographic one. In this paper, the complexity of this order conversion is omitted.

$2\lceil t\sigma \rceil + 1 > 2Tn_{\text{solve}}$, the degree can be reduced to $D = 2Tn_{\text{solve}}$. In this regime, n_{solve} decreases as n_{lat} increases, leading to a decrease in the overall complexity.

However, n_{lat} cannot be increased arbitrarily, since it must also satisfy $2\lceil t\sigma \rceil + 1 < q$. Consequently, as shown in Figure 1, the lowest complexity is still attained at $n_{\text{lat}} = 1$. This complexity remains prohibitively high and substantially exceeds the estimates in [41], namely 1584, 1588, and 2014 for KYBER-512, KYBER-768, and KYBER-1024, respectively. Therefore, constructing the Arora–Ge system and solving it via Gröbner bases in the second stage appears to be ineffective.

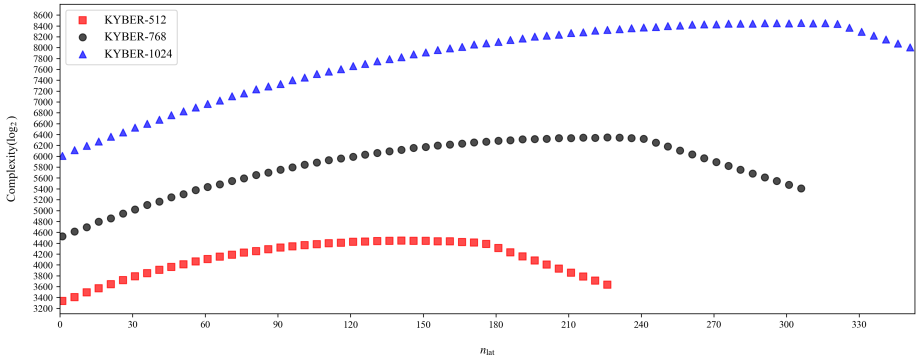


Fig. 1. Complexity estimation with the increase of n_{lat} for CRYSTALS-KYBER. The horizontal axis represents n_{lat}, and the vertical axis represents the complexity ($\log_2$). The parameter $\beta_{\text{bkz}} = \beta_{\text{sieve}} = r$, where r is the dimension of $\mathbf{B}$ (see Algorithm 2). The growth of n_{lat} is restricted by the condition that $2\lceil t\sigma \rceil + 1 < q$.

3.2.2 Construction of a New Polynomial System and Solving by the Resultant-Based Method

In Section 3.2.1, we have shown that constructing the Arora-Ge system and then solving it with the Gröbner basis method leads to prohibitively high complexity in the second stage. An analysis of the underlying causes reveals that the Arora-Ge system merely enumerates all possible values of the amplified error without exploiting their actual distribution; moreover, the rapid growth of the polynomial degree negatively affects the overall complexity. To exploit this specific distributional feature, we design a new polynomial system that explicitly incorporates the zero-error probability and solve it using a resultant-based method.

Construction of a New Polynomial System. Motivated by the observation that the amplified error follows a modular Gaussian distribution with a mean of zero, we construct a polynomial system that focuses on those high-probability values rather than enumerating all possibilities. This approach completely removes the requirement that the size of the error range must be smaller than

the modulus q. However, it also reduces the probability that the resulting equations hold for the true secret. To compensate, we leverage the large number of new LWE samples produced in the first stage and form each polynomial from multiple samples, thereby increasing the success probability.

We first present the general form of this new polynomial system. For an integer $0 \le R \le T'$, we define the polynomial

$$F = \prod_{i=1}^{K} X_i \cdot \prod_{i=1}^{K} (X_i + 1)(X_i - 1) \cdots \prod_{i=1}^{K} (X_i + R)(X_i - R). \tag{5}$$

Notably, setting $K = 1$ and $R = T'$ reduces Eq. (5) exactly to the original Arora-Ge polynomial.

In practice, we adopt a streamlined construction that directly exploits the highest probability, which is the zero-error probability. Specifically, we consider the polynomial $F = \prod_{i=1}^{K} X_i$. To obtain a concrete multivariate polynomial, we randomly select K distinct indices $\{j_1, j_2, \ldots, j_K\}$ from the sample set Ω and replace each variable X_i with the linear form $b'_{j_i} - \langle \mathbf{a}'_{j_i}, \mathbf{z} \rangle$ for $i = 1, 2, \ldots, K$. This yields

$$F = \prod_{i=1}^{K} \Big(b'_{j_i} - \langle \mathbf{a}'_{j_i}, \mathbf{z} \rangle \Big). \tag{6}$$

By selecting W such polynomials $F_1, F_2, \ldots, F_W$ from disjoint index sets (meaning that all selected samples are distinct), we construct the polynomial system

$$\mathcal{F}_{\text{new}} \triangleq \big\{ F_1, F_2, \ldots, F_W \big\}. \tag{7}$$

Two fundamental questions naturally arise: (1) What is the probability that the true secret satisfies the system $\mathcal{F}_{\text{new}}$? (2) What is the expected number of solutions to this system? Let $\mathbf{s}_{\text{solve}}$ be the true secret vector and $\tilde{\mathbf{s}} \neq \mathbf{s}_{\text{solve}}$ an arbitrary incorrect candidate.

Ducas and Pulles [22,23] showed that the Independent Score Heuristic used in previous dual-sieve analyses is problematic, because the target vector acts as a confounding variable. Their refined view is to first fix the target vector (or more precisely the target vector norm) and then model the short dual vectors output by the sieve as i.i.d. samples. In the following, we use this fixed-target heuristic to provide an estimate for the probability that the true secret satisfies $\mathcal{F}_{\text{new}}$.

Heuristic Claim 1 *Let $r = m + n_{\text{lat}}$ and $\mathbf{t} = (\mathbf{e}, \mathbf{s}_{\text{lat}})$. For the correct candidate $\mathbf{z} = \mathbf{s}_{\text{solve}}$, conditioned on a fixed value of $\|\mathbf{t}\|^2$, the probability that $\mathbf{s}_{\text{solve}}$ satisfies $\mathcal{F}_{\text{new}}$ is approximated by $\Phi(\|\mathbf{t}\|^2)$. Then, we obtain*

$$p_{\text{true}} \approx \mathbb{E}_{\|\mathbf{t}\|^2} \big[\Phi(\|\mathbf{t}\|^2) \big], \tag{8}$$

where

$$\Phi(\|\mathbf{t}\|^2) = \left(1 - \left(1 - \rho_{q,\frac{\|\mathbf{t}\|\,\ell}{\sqrt{r}}}(0)\right)^K\right)^W,$$

and ℓ is given by Lemma 3.

If the entries of $\mathbf{e}$ and $\mathbf{s}$ are sampled from centered discrete Gaussian distributions with variance σ_χ^2, then

$$p_{\text{true}} \approx \int_0^\infty \Phi(\sigma_\chi^2 x) f_{\chi_r^2}(x)\,dx, \qquad f_{\chi_r^2}(x) = \frac{x^{r/2-1}e^{-x/2}}{2^{r/2}\Gamma(r/2)}. \tag{9}$$

If they are sampled from the centered binomial distribution $\mathcal{B}_\eta$, then

$$p_{\text{true}} \approx \sum_{k=0}^{r\eta^2} \Phi(k) \cdot \Pr[\|\mathbf{t}\|^2 = k], \qquad \Pr[\|\mathbf{t}\|^2 = k] = [z^k]\left(\sum_{a=0}^{\eta^2} \Pr[Y = a] z^a\right)^r, \tag{10}$$

where Y denotes the square of one component of $\mathbf{t}$, and $[z^k]F(z)$ denotes the coefficient of z^k in $F(z)$.

Justification. Its justification consists of three parts: first, we derive Eq. (8) from the fixed-target heuristic; second, we specialize it to the discrete Gaussian case, yielding Eq. (9); finally, we specialize the estimate to the centered binomial distribution, yielding Eq. (10).

Let

$$r = m + n_{\text{lat}}, \qquad \mathbf{t} = (\mathbf{e}, \mathbf{s}_{\text{lat}}).$$

Justification of Eq. (8) Let $\mathbf{w}_i = (\mathbf{x}_i, \mathbf{y}_i)$ be a short vector produced in the first stage, satisfying

$$\mathbf{A}_{\text{lat}}^\top \mathbf{x}_i = \mathbf{y}_i \bmod q.$$

Recall that

$$\mathbf{a}'_i = \mathbf{A}_{\text{solve}}^\top \mathbf{x}_i, \qquad b'_i = \langle \mathbf{x}_i, \mathbf{b} \rangle.$$

Since $\mathbf{b} = \mathbf{A}_{\text{solve}}\mathbf{s}_{\text{solve}} + \mathbf{A}_{\text{lat}}\mathbf{s}_{\text{lat}} + \mathbf{e} \bmod q$, for the correct candidate $\mathbf{z} = \mathbf{s}_{\text{solve}}$, we have

$$\begin{aligned} b'_i - \langle \mathbf{a}'_i, \mathbf{s}_{\text{solve}} \rangle &= \langle \mathbf{x}_i, \mathbf{b} - \mathbf{A}_{\text{solve}}\mathbf{s}_{\text{solve}} \rangle \bmod q \\ &= \langle \mathbf{x}_i, \mathbf{A}_{\text{lat}}\mathbf{s}_{\text{lat}} + \mathbf{e} \rangle \bmod q \\ &= \langle \mathbf{y}_i, \mathbf{s}_{\text{lat}} \rangle + \langle \mathbf{x}_i, \mathbf{e} \rangle \bmod q \\ &= \langle \mathbf{w}_i, \mathbf{t} \rangle \bmod q. \end{aligned}$$

Therefore, a linear factor of the polynomial constructed in Eq. (6) vanishes at the correct candidate if and only if

$$\langle \mathbf{w}_i, \mathbf{t} \rangle = 0 \bmod q.$$

For a fixed value of $\|\mathbf{t}\|^2$, we define the corresponding conditional probability

$$p(\|\mathbf{t}\|^2) = \Pr\left[\langle \mathbf{w}, \mathbf{t} \rangle = 0 \bmod q \,\middle|\, \|\mathbf{t}\|^2\right],$$

where $\mathbf{w}$ denotes a short vector sampled in the first stage of Algorithm 1.

By Lemma 4, the variance of $\langle \mathbf{w}, \mathbf{t} \rangle = e'$ is computed by

$$\frac{\|\mathbf{t}\|^2 \ell^2}{r},$$

where ℓ is given by Lemma 3. Hence

$$p(\|\mathbf{t}\|^2) = \rho_{q, \frac{\|\mathbf{t}\|\ell}{\sqrt{r}}}(0),$$

where $\rho_{q,\sigma}$ is defined in Eq. (1).

Under the fixed-target heuristic, conditioned on $\|\mathbf{t}\|^2$, the values $\langle \mathbf{w}_i, \mathbf{t} \rangle \bmod q$ arising from distinct sieve outputs are modeled as mutually independent. Thus, a polynomial F_j constructed from K distinct samples vanishes at the correct candidate with probability approximately

$$1 - \left(1 - p(\|\mathbf{t}\|^2)\right)^K.$$

Since the W polynomials are constructed from disjoint sample blocks, their vanishing events are also modeled as conditionally independent. Consequently,

$$\Pr\left[\mathbf{s}_{\text{solve}} \text{ satisfies } \mathcal{F}_{\text{new}} \,\middle|\, \|\mathbf{t}\|^2\right] \approx \left(1 - \left(1 - p(\|\mathbf{t}\|^2)\right)^K\right)^W.$$

Finally, we obtain

$$p_{\text{true}} \approx \mathbb{E}_{\|\mathbf{t}\|^2}\left[\left(1 - \left(1 - p(\|\mathbf{t}\|^2)\right)^K\right)^W\right] = \mathbb{E}_{\|\mathbf{t}\|^2}\left[\Phi(\|\mathbf{t}\|^2)\right],$$

where

$$\Phi(\|\mathbf{t}\|^2) = \left(1 - \left(1 - \rho_{q, \frac{\|\mathbf{t}\|\ell}{\sqrt{r}}}(0)\right)^K\right)^W.$$

Justification of Eq. (9) We now specialize Eq. (8) to the case where the entries of $\mathbf{e}$ and $\mathbf{s}_{\text{lat}}$ are sampled from centered discrete Gaussian distributions with common variance σ_χ^2. Ignoring the effect of discretization, we model

$$\frac{\|\mathbf{t}\|^2}{\sigma_\chi^2} \sim \chi_r^2.$$

Therefore,

$$p_{\text{true}} \approx \int_0^\infty \Phi(\sigma_\chi^2 x) f_{\chi_r^2}(x)\, dx,$$

where $f_{\chi_r^2}$ denotes the density of the chi-square distribution with r degrees of freedom, and

$$\Phi(\sigma_\chi^2 x) = \left(1 - \left(1 - \rho_{q, \frac{\sqrt{\sigma_\chi^2 x}\ell}{\sqrt{r}}}(0)\right)^K\right)^W.$$

This gives Eq. (9).

Justification of Eq. (10) We now specialize Eq. (8) to the case where all coordinates of $\mathbf{e}$ and $\mathbf{s}_{\text{lat}}$ are sampled independently from the centered binomial distribution $\mathcal{B}_\eta$. Recall that

$$S = \|\mathbf{t}\|^2 = \|\mathbf{e}\|^2 + \|\mathbf{s}_{\text{lat}}\|^2.$$

For a random variable $X \sim \mathcal{B}_\eta$, we have

$$X = \sum_{i=1}^{\eta} U_i - \sum_{i=1}^{\eta} V_i,$$

where the U_i's and V_i's are independent and uniformly distributed over $\{0, 1\}$. Hence

$$\Pr[X = x] = 2^{-2\eta} \binom{2\eta}{\eta + x}, \qquad x = -\eta, \ldots, \eta.$$

Let $Y = X^2$. The probability mass function of Y is given by

$$\Pr[Y = a] = \sum_{\substack{x=-\eta \\ x^2=a}}^{\eta} 2^{-2\eta} \binom{2\eta}{\eta + x}, \qquad a \in \{0, 1, \ldots, \eta^2\}.$$

Since $S = \sum_{i=1}^{r} X_i^2$ for independent copies $X_i \sim \mathcal{B}_\eta$, the probability mass function of S is obtained by convolution. Equivalently,

$$\Pr[S = k] = [z^k] \left(\sum_{a=0}^{\eta^2} \Pr[Y = a] z^a \right)^r,$$

where $[z^k]F(z)$ denotes the coefficient of z^k in $F(z)$.

For a fixed value $S = k$, the variance of $\langle \mathbf{w}, \mathbf{t} \rangle$ is approximated by

$$\sigma_k^2 = \frac{\ell^2}{r} k.$$

Thus, we obtain

$$p_{\text{true}} \approx \sum_{k=0}^{r\eta^2} \Phi(k) \cdot \Pr[S = k],$$

where

$$\Phi(k) = \left(1 - \left(1 - \rho_{q, \ell\sqrt{k/r}}(0) \right)^K \right)^W.$$

This gives Eq. (10). □

We then estimate the expected number of solutions.

Lemma 6. *Let $\mathcal{F}_{new} = \{F_1, F_2, \ldots, F_W\}$ be the polynomial system defined as in Eq. (7), where each coordinate of $\mathbf{z}$ is restricted to an integer interval $[\![-T, T]\!]$. Let N_{sol} denote the number of solutions to $\mathcal{F}_{new}$. Then we have*

$$\mathbb{E}\big[N_{\mathrm{sol}}\big] \approx \big((2T+1)^{n_{\mathrm{solve}}} - 1\big) \cdot \Big(1 - \big(1 - \tfrac{1}{q}\big)^K\Big)^W + p_{\mathrm{true}}. \tag{11}$$

Proof. Consider a single polynomial $F_j = \prod_{i=1}^{K} \big(b'_{j_i} - \langle \mathbf{a}'_{j_i}, \mathbf{z}\rangle\big)$. For a candidate solution $\tilde{\mathbf{s}} \neq \mathbf{s}_{\mathrm{solve}}$ with coordinates lie in $[\![-T, T]\!]$, each linear factor appearing in $F_j(\tilde{\mathbf{s}})$ satisfies

$$\begin{aligned} b'_{j_i} - \langle \mathbf{a}'_{j_i}, \tilde{\mathbf{s}}\rangle &= \langle \mathbf{x}_{j_i}, \mathbf{b}\rangle - \big\langle \mathbf{A}_{\mathrm{solve}}^{\top}\mathbf{x}_{j_i}, \tilde{\mathbf{s}}\big\rangle \bmod q \\ &= \langle \mathbf{x}_{j_i}, \mathbf{A}_{\mathrm{solve}}(\mathbf{s}_{\mathrm{solve}} - \tilde{\mathbf{s}})\rangle + \langle \mathbf{x}_{j_i}, \mathbf{e}\rangle + \langle \mathbf{y}_{j_i}, \mathbf{s}_{\mathrm{lat}}\rangle \bmod q \\ &= \langle \mathbf{x}_{j_i}, \mathbf{A}_{\mathrm{solve}}(\mathbf{s}_{\mathrm{solve}} - \tilde{\mathbf{s}})\rangle + e'_{j_i} \bmod q. \end{aligned}$$

Because $\mathbf{s}_{\mathrm{solve}} - \tilde{\mathbf{s}} \neq 0$, the vector $\mathbf{A}_{\mathrm{solve}}(\mathbf{s}_{\mathrm{solve}} - \tilde{\mathbf{s}}) \bmod q$ is uniformly distributed over $\mathbb{Z}_q^m$. Therefore, $\langle \mathbf{x}_{j_i}, \mathbf{A}_{\mathrm{solve}}(\mathbf{s}_{\mathrm{solve}} - \tilde{\mathbf{s}})\rangle + e'_{j_i} \bmod q$ is uniformly distributed over $\mathbb{Z}_q$. Hence, the probability that the linear factor $b'_{j_i} - \langle \mathbf{a}'_{j_i}, \tilde{\mathbf{s}}\rangle$ equals zero can be approximated by $1/q$. Since the polynomials $F_1, \ldots, F_W$ are constructed from disjoint sample sets, the probability is $\left(1 - \left(1 - \frac{1}{q}\right)^K\right)^W$. Multiplying this probability by the number of candidates $\tilde{\mathbf{s}} \neq \mathbf{s}_{\mathrm{solve}}$ and adding the probability that the true secret satisfies the system gives the desired expectation. □

To successfully recover the secret, the polynomial system must satisfy two competing requirements simultaneously: it should contain the true solution with high probability, while keeping the expected total number of solutions very small. Balancing these two requirements leads to the following constraints on the parameters K and W:

$$p_{\mathrm{true}} \approx \mathbb{E}_{\|\mathbf{t}\|^2}\big[\Phi(\|\mathbf{t}\|^2)\big] = \theta, \tag{12}$$

$$\mathbb{E}(N_{\mathrm{sol}}) \approx p_{\mathrm{true}} + ((2T+1)^{n_{\mathrm{solve}}} - 1)\left(1 - \left(1 - \frac{1}{q}\right)^K\right)^W = \theta + \epsilon_{\mathrm{sol}}, \tag{13}$$

where $\mathbf{t} = (\mathbf{e}, \mathbf{s}_{\mathrm{lat}})$, θ is a prescribed threshold ensuring a high probability of retaining the true solution, and $\epsilon_{\mathrm{sol}} \ll 1$ denotes a positive quantity close to zero.

In practice, one can improve the reliability of recovering the true secret $\mathbf{s}_{\mathrm{solve}}$ by independently constructing multiple polynomial systems of the form (7) and solving each one separately. All candidate solutions are then ranked by their frequency of occurrence, and those whose scores exceed a predefined threshold [6]

[6] The threshold is determined by the probability parameter θ. For instance, with $\theta = 0.8$ and 8 independently constructed systems, the probability that the score of the true secret is at least 3 exceeds 99.8%; hence the threshold can be set to 3.

are retained. For each retained candidate, we substitute it back into the original LWE samples to obtain a sub-LWE problem of reduced dimension, solve it to recover the remaining part of the secret, and finally verify the full candidate by checking whether the resulting error vector $\tilde{\mathbf{e}} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{s}} \mod q$ follows the expected distribution. Only candidates that pass this distributional test are accepted as valid solutions.

Remark 1. After the first stage, an exponential number of new LWE samples is available. One may filter them, retaining only those whose vectors $\mathbf{a}'$ share a common position of zero entries. This approach implicitly eliminates the corresponding variables—we let γ denote the number of variables removed in this way. However, such filtering shrinks the pool of usable samples, thereby limiting the feasible choices for the parameters K and W. Consequently, a careful balance must be struck between the number of variables eliminated γ and the number of samples retained.

We experimentally verify the effectiveness of the proposed polynomial system $\mathcal{F}_{\text{new}}$ through several different parameter sets. The full verification code is available on GitHub [7].

Experimental verification of the polynomial-system construction. We experimentally verify two predictions used in the construction: the survival probability of the true secret in Eq. (8), and the expected number of solutions in Eq. (11). The verification is performed on **six parameter sets**, and the results are summarized in Table 1. We describe one representative example in detail:

$$(q, \eta, n, m, n_{\text{lat}}, \beta_{\text{bkz}}, \beta_{\text{sieve}}) = (3329, 1, 50, 50, 40, 40, 80).$$

Here $\eta = 1$, so both the secret and the error are sampled from the centered binomial distribution $\mathcal{B}_1$. After the dual-lattice reduction stage, this instance yields $N = 99438$ new samples.

For the polynomial-system parameters $(K, W) = (700, 8)$, our estimate gives $p_{\text{true}} \approx 0.81$, while Eq. (11) predicts $\mathbb{E}(N_{\text{sol}}) \approx 0.91$.

Verification of Eq. (8). We independently generate 20 original LWE instances with parameters $(q, \eta, n, m) = (3329, 1, 50, 50)$. For each instance, we first perform dual lattice reduction with parameter $(n_{\text{lat}}, \beta_{\text{bkz}}, \beta_{\text{sieve}}) = (40, 40, 80)$ to transform it into N lower-dimensional samples. Then, using the actual amplified errors, we determine which of the N new samples vanish at the true secret. We then perform 4000 independent simulation runs. In each run, we sample KW distinct new samples without replacement and split them into W disjoint blocks of size K. A run is declared successful if every block contains at least one sample vanishing at the true secret; equivalently, the resulting polynomial system contains the true secret as a solution. The empirical success probability, averaged over the 20 instances, is 0.82, with a standard deviation of 0.043. This agrees closely with the prediction $p_{\text{true}} \approx 0.81$.

[7] https://github.com/latticewalker/Algebraic-Analysis-for-LWE.

$(m, n, n_{\text{lat}}, \beta_{\text{bkz}},$ $\beta_{\text{sieve}}, K, W)$	Theory $(p_{\text{true}}, \mathbb{E}(N_{\text{sol}}))$	Empirical p_{true} mean/std	Empirical N_{sol} mean/std
$(45, 45, 35, 30, 80, 600, 7)$	$(0.85, 1.05)$	0.85/0.044	1.05/0.036
$(50, 50, 40, 40, 80, 700, 8)$	$(0.81, 0.91)$	0.82/0.043	0.91/0.064
$(60, 60, 50, 40, 80, 1000, 9)$	$(0.81, 1.13)$	0.80/0.037	1.14/0.072
$(65, 65, 55, 40, 80, 1200, 11)$	$(0.79, 0.91)$	0.77/0.066	0.88/0.089
$(70, 70, 60, 40, 80, 1600, 15)$	$(0.83, 0.87)$	0.82/0.068	0.86/0.077
$(80, 80, 70, 40, 90, 2100, 18)$	$(0.84, 0.91)$	0.82/0.052	0.90/0.079

Table 1. Experimental verification of the polynomial system construction. Here p_{true} denotes the probability that the true secret satisfies $\mathcal{F}_{\text{new}}$, and N_{sol} denotes the number of solutions.

Verification of Eq. (11). Using the same 20 original LWE instances and their transformed samples, we independently construct and solve 100 polynomial systems $\mathcal{F}_{\text{new}}$ for each instance with the same parameters $(K, W) = (700, 8)$, and record the number of solutions. Averaging over the 20 instances, the empirical mean number of solutions is 0.91, with a standard deviation of 0.064. Across all constructed systems, the number of solutions ranges from 0 to 3. Thus the observed solution count agrees with the prediction $\mathbb{E}(N_{\text{sol}}) \approx 0.91$.

Complexity of Solving Eq. (7) by the Resultant-based Method. To solve the system $\mathcal{F}_{\text{new}}$, we first consider the Gröbner basis methods. However, two main difficulties arise. First, it is unclear whether Conjecture 1, which underpins the "lowest achievable complexity estimates" from [41], still holds for our newly constructed system; consequently, the complexity analysis will likely need to rely on the semi-regularity assumption. Second, the polynomial degrees in our system remain relatively high—a property generally detrimental to efficient Gröbner basis computation. Therefore, we opt for a resultant-based method. This approach offers two key advantages in our context: (1) it yields a concrete complexity estimate for the second stage (polynomial system solving) without relying on any heuristic assumptions, and (2) its complexity is primarily governed by the number of variables and the partial degree of each variable—quantities that we can control tightly.

Specifically, we introduce a polynomial constraint of the following form for each variable $s_i \prod_{j=1}^{T}(s_i + j)(s_i - j) = s_i^d - h_i(s_i)$, where $d = 2T + 1$ and $h_i(s_i)$ is a polynomial of degree less than d. The **reduction system** is then defined as

$$\mathcal{P} = \left\{ s_1^d - h_1(s_1),\ s_2^d - h_2(s_2),\ \ldots,\ s_{n_{\text{solve}}}^d - h_{n_{\text{solve}}}(s_{n_{\text{solve}}}) \right\}. \tag{14}$$

Whenever the degree of any variable in a polynomial reaches or exceeds d, we reduce it modulo the system $\mathcal{P}$, thereby ensuring that the degree of each variable remains strictly below d. The reduction procedure and complexity analysis follow the standard recursive algorithm of [9]. We therefore defer the full pseudocode, given in Algorithm 3, together with the proof of Lemma 7.

Algorithm 3 $\mathrm{Reduce}_u(g(x_1, x_2, \ldots, x_u), \mathcal{P})$

Input: A polynomial $g \in \mathbb{Z}_q[x_1, x_2, \ldots, x_u]$, where $u \leq n$ and $\deg_{x_i}(g) \leq 2d-2$ for $1 \leq i \leq u$, and a reduction system $\mathcal{P}$.
Output: A polynomial whose degree in each variable is less than d.

if $u = 0$ **then**
  return g
end if
Express $g = \sum_{i=0}^{2d-2} g_i(x_1, x_2, \ldots, x_{u-1}) x_u^i$.
$\rho \leftarrow 0$.
for $i = 0$ to $d-2$ **do**
  $\rho \leftarrow \rho + \mathrm{Reduce}_{u-1}(g_i, \mathcal{P}) \cdot x_u^i + \mathrm{Reduce}_{u-1}(g_{i+d}, \mathcal{P}) \cdot \big(x_u^{i+d} \bmod (x_u^d - h_u)\big)$.
end for
$\rho \leftarrow \rho + \mathrm{Reduce}_{u-1}(g_{d-1}, \mathcal{P}) \cdot x_u^{d-1}$.
return ρ

Lemma 7. *Let $C(u,d)$ denote the worst-case time complexity, measured in $\mathbb{Z}_q$-operations, of the reduction procedure* $\mathrm{Reduce}_u(g(x_1, x_2, \ldots, x_u), \mathcal{P})$ *of Algorithm 3, where g is a polynomial in u variables whose partial degree with respect to each variable is at most $2d-2$. Then reducing g to a polynomial whose partial degree with respect to each variable is less than d costs*

$$C(u,d) = \mathcal{O}\big((2d-1)^{u+1}\big).$$

Proof. When $u = 0$, no reduction is required. For $u \geq 1$, Algorithm 3 makes $2d-1$ recursive calls to Reduce_{u-1}. In addition, it performs $d-1$ multiplications involving the precomputed reductions $x_u^{i+d} \bmod (x_u^d - h_u)$.

After reducing the first $u-1$ variables, each polynomial $\mathrm{Reduce}_{u-1}(g_{i+d}, \mathcal{P})$ contains at most d^{u-1} monomials. Therefore, each such multiplication costs $\mathcal{O}(d^u)$ operations over $\mathbb{Z}_q$. Hence,

$$C(0,d) = 0, \qquad C(u,d) = (2d-1)C(u-1,d) + (d-1)d^u.$$

Expanding the recurrence gives

$$C(u,d) = (d-1)\sum_{i=0}^{u-1}(2d-1)^i d^{u-i} < \sum_{i=0}^{u-1}(2d-1)^i d^{u-i+1} = \mathcal{O}\big((2d-1)^{u+1}\big). \square$$

Theorem 2 establishes the time complexity of solving $\mathcal{F}_{\mathrm{new}}$ via the resultant-based method. We have implemented the solution procedure for small-scale parameter cases in Magma code, and the code is available in the GitHub repository mentioned in the introduction.

Theorem 2. *Let q be a prime, $m = n \geq 2$ be LWE parameters, with the secret $\mathbf{s}$ and error $\mathbf{e}$ having coordinates chosen from $[\![-T, T]\!]$. After performing dual lattice reduction with parameters n_{lat}, $n_{\mathrm{solve}} = n - n_{\mathrm{lat}}$, β_{bkz}, and β_{sieve}, let $\mathcal{F}_{\mathrm{new}} = \{F_1, F_2, ..., F_W\}$ be the polynomial system constructed according to*

Eq. (7) *and satisfying the two constraints given in Eqs.* (12) *and* (13)*. Denote* $d = 2T + 1$*. Then, with a success probability of at least* θ*,* $\mathcal{F}_{\text{new}}$ *can be solved by the resultant-based method, and its time complexity (in* $\mathbb{Z}_q$*-operations) is upper bounded by*

$$\mathcal{O}\Big(T_{\text{gen}} + (2d-1)!(d-1) + W2^{2d-1}(d-1)\big(v \log v \log\log v + (2d-1)^{\,n_{\text{solve}}}\big)\Big), \tag{15}$$

where $v = (2d-1)^{\,n_{\text{solve}}-1}$ *and* $T_{\text{gen}} = \mathcal{O}\left(KW(d+1)^{n_{\text{solve}}+1}\right)$.

Proof. Fix θ, choose K, W satisfying Eqs. (12) and (13), and let $\mathcal{P}$ be the reduction system in Eq. (14). The total cost comprises generating $\mathcal{F}_{\text{new}}$, performing the successive resultant eliminations, and the final back-substitution and gcd steps. Throughout we take the worst case in which all admissible monomials appear.

Generation of $\mathcal{F}_{\text{new}}$. Each F_j is a product of K linear forms, generated iteratively: after each multiplication by a linear form, Algorithm 3 reduces individual degree $< d$ in every variable. A reduced intermediate polynomial thus has at most $d^{n_{\text{solve}}}$ monomials, and multiplication by a linear form (at most $n_{\text{solve}} + 1$ terms) costs $\mathcal{O}(n_{\text{solve}} d^{n_{\text{solve}}})$. The ensuing reduction is cheaper than in Lemma 7: since the input has individual degree at most d rather than $2d-2$, reducing in u variables makes $d+1$ recursive calls instead of $2d-1$, giving

$$C_{\text{gen}}(0,d) = 0, \qquad C_{\text{gen}}(u,d) = (d+1)C_{\text{gen}}(u-1,d) + d^u = \mathcal{O}\big((d+1)^{u+1}\big).$$

With $u = n_{\text{solve}}$, one multiplication–reduction step costs

$$\mathcal{O}\left(n_{\text{solve}} d^{n_{\text{solve}}} + (d+1)^{n_{\text{solve}}+1}\right).$$

Each F_j needs K such steps over W polynomials, so

$$T_{\text{gen}} = \mathcal{O}\big(KW\left(n_{\text{solve}} d^{n_{\text{solve}}} + (d+1)^{n_{\text{solve}}+1}\right)\big) = \mathcal{O}(KW \cdot (d+1)^{n_{\text{solve}}+1}).$$

Successive resultant eliminations. Let $H_{n_{\text{solve}},j} = F_j$ for $j = 1, 2, \ldots, W$. The elimination process then proceeds iteratively as follows:

$$\begin{cases} H_{i-1,1} = \text{Res}_{s_i}\big(H_{i,1},\ s_i^d - h_i(s_i)\big), \\ H_{i-1,2} = \text{Res}_{s_i}\big(H_{i,2},\ s_i^d - h_i(s_i)\big), \\ \qquad \vdots \\ H_{i-1,W} = \text{Res}_{s_i}\big(H_{i,W},\ s_i^d - h_i(s_i)\big), \end{cases} \tag{16}$$

where $i = n_{\text{solve}}, n_{\text{solve}} - 1, \ldots, 2$. Algorithm 3 keeps every individual degree $< d$, so no intermediate polynomial exceeds degree $2d-2$. For $s_{n_{\text{solve}}}$, each resultant is a $(2d-1)\times(2d-1)$ Sylvester determinant, whose first d rows contain multivariate polynomial entries and whose last $d-1$ rows contain only scalar entries. Treating the polynomial entries as formal variables, the determinant expansion has degree

d, can be precomputed in $\mathcal{O}((2d-1)!(d-1))$, and contains at most $\binom{2d-1}{d} < 2^{2d-1}$ monomials. Thus, evaluating one resultant requires at most $2^{2d-1}(d-1)$ multiplication-reduction steps. By Lemma 1, multiplying two polynomials in $n_{\text{solve}}-1$ variables of individual degree $< d$ costs $\mathcal{O}(v \log v \log\log v)$ with $v = (2d-1)^{n_{\text{solve}}-1}$, and by Lemma 7 the reduction costs $\mathcal{O}((2d-1)^{n_{\text{solve}}})$. Thus eliminating $s_{n_{\text{solve}}}$ costs

$$\mathcal{O}\Big((2d{-}1)!(d{-}1){+}W2^{2d-1}(d{-}1)\big(v\log v\log\log v{+}(2d{-}1)^{n_{\text{solve}}}\big)\Big) \triangleq C(d,W,n_{\text{solve}}).$$

The remaining eliminations cost $C(d,W,n_{\text{solve}}-1),\ldots,C(d,W,2)$; as the dominant term of $C(d,W,r)$ grows exponentially in r, all are dominated by $C(d,W,n_{\text{solve}})$.[8]

Back-substitution and gcd. The elimination yields W univariate polynomials in s_1, whose gcd gives s_1; substituting back into the bivariate polynomials in (s_1,s_2) and taking gcds recovers s_2, and so on through $s_{n_{\text{solve}}}$. Spurious candidates may arise, but their presence does not affect the asymptotic bound. For a worst-case estimate, we assume that each of the first $n_{\text{solve}}-1$ variables $s_1,\ldots,s_{n_{\text{solve}}-1}$ ranges over all d possible values; substituting these assignments into F_j $(j=1,\ldots,W)$ and taking gcds in $s_{n_{\text{solve}}}$ then recovers all solutions of F_{new} without spurious ones.

Let $S = [\![-T,T]\!]$ with $|S| = d$, and write each reduced expanded polynomial from the generation phase as $\overline{F}_j = \sum_{k=0}^{d-1} f_{j,k}(s_2,\ldots,s_{n_{\text{solve}}})\, s_1^k$, where each coefficient $f_{j,k}$ has at most $d^{n_{\text{solve}}-1}$ monomials. Substituting a fixed $\alpha \in S$ for s_1 costs $O(d^{n_{\text{solve}}})$, so all d values over the W polynomials cost $O(Wd^{n_{\text{solve}}+1})$. At each subsequent variable the number of polynomials grows by a factor of d while the monomial count per polynomial drops by the same factor; since their product $Wd^{n_{\text{solve}}+1}$ remains constant, the cost per step stays $O(Wd^{n_{\text{solve}}+1})$. Summing over the $n_{\text{solve}}-1$ substitution steps and the final univariate gcd step, which is of the same order, gives

$$O\big(W\, n_{\text{solve}}\, d^{n_{\text{solve}}+1}\big),$$

which is negligible relative to $C(d,W,n_{\text{solve}})$.

Combining the three estimates yields the claimed bound. □

Although the Sylvester determinant considered in the proof of Theorem 2 admits a reduction to a smaller size, following Proposition 9 of [9], this reduction does not affect the dominant asymptotic complexity.

For

$$\operatorname{Res}_{s_i}(H_{i,j}, g_i), \qquad g_i = s_i^d - h_i(s_i),$$

the Sylvester determinant of size $(2d-1)\times(2d-1)$ can be reduced to a $d\times d$ determinant by using the same block-elimination idea as in Proposition 9 of [10]. Indeed, write

$$H_{i,j}(s_i) = a_{d-1}s_i^{d-1} + \cdots + a_1 s_i + a_0.$$

[8] In our small-scale experiments, we observed that early eliminations can be faster due to the sparsity of the corresponding polynomials. However, for the purpose of complexity analysis, tracking the exact number of terms is difficult; therefore, we adopt the worst-case bound in which all admissible monomials appear.

Since g_i is monic of degree d, the Sylvester matrix can be written in block form as

$$\mathrm{Syl}(H_{i,j}, g_i) = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{L} & \mathbf{R} \end{pmatrix},$$

where $\mathbf{L}$ is unit triangular, and the entries of both $\mathbf{L}$ and $\mathbf{R}$ are constants. Applying determinant-preserving column operations gives

$$\det \mathrm{Syl}(H_{i,j}, g_i) = \det(\mathbf{B} - \mathbf{A}\mathbf{L}^{-1}\mathbf{R}),$$

where $\mathbf{B} - \mathbf{A}\mathbf{L}^{-1}\mathbf{R}$ is a $d \times d$ matrix.

This reduction, however, does not change the dominant term in Theorem 2. In the proof, the resultant is pre-expanded as a polynomial in $a_0, \ldots, a_{d-1}$ and then evaluated after collecting like terms. The resulting expanded polynomial is independent of whether it is obtained from the original Sylvester determinant or from the reduced $d \times d$ determinant. Thus, the $d \times d$ determinant representation may reduce the determinant precomputation cost, but it does not change the main asymptotic complexity.

The time complexity above is expressed in terms of the number of field operations. To obtain a bit complexity, one can equate one field operation to $\lceil \log q \rceil^2$ bit operations.

We compare Algorithm 1 with the "lowest achievable complexity estimate" from [41] using CRYSTALS-KYBER as a concrete example. We first derive the time complexity of our algorithm and then describe the parameter selection that minimizes it.

The overall complexity combines the costs of the two stages. The first stage, from Lemma 5, costs $\mathcal{O}(2^{0.292\beta})$, where β is the blocksize. For the second stage, we start from Eq. (15) of Theorem 2. Since the secret follows a centered binomial distribution $\mathcal{B}_\eta$ ($\eta = 3$ for KYBER-512, $\eta = 2$ for KYBER-768/1024), the parameter T and thus $d = 2T + 1$ are small, making $(2d-1)! \cdot (d-1)$ negligible. We refine the estimate by treating each distinct polynomial appearing in the resultant determinant as a separate variable, expanding the determinant in practice, counting the resulting monomials, and thereby obtaining the exact number of multiplications: 2832 for KYBER-512 and 152 for KYBER-768/1024. The combined time complexity can be estimated as

$$\mathcal{O}\big(2^{0.292\beta} + \lceil \log q \rceil^2 \left(T_{\mathrm{gen}} + W w_i \left(v \log v \log\log v + (2d-1)^{\,n_{\mathrm{solve}} - \gamma}\right)\right)\big),$$

where $d = 2T + 1, v = (2d-1)^{\,n_{\mathrm{solve}} - \gamma - 1}$, and $T_{\mathrm{gen}} = \mathcal{O}\big(KW(d+1)^{\,n_{\mathrm{solve}} - \gamma + 1}\big)$. Here $w_1 = 2832$, $w_2 = 152$, and γ denotes the number of variables eliminated via Remark 1.

We fix the overall success probability at least $p_{\mathrm{true}} = 0.80$. To minimize this complexity, we perform a parameter search under the following procedure. The block size parameter β (where $\beta = \beta_{\mathrm{bkz}} = \beta_{\mathrm{sieve}}$) is chosen according to the core-SVP methodology in [40, Table 4], ensuring it remains strictly below the corresponding primal-attack BKZ blocksize of 403, 625, and 877 for KYBER-512/768/1024, respectively. We then search for the parameter tuple

$(n_{\text{lat}}, K, W, \gamma)$ while satisfying Eqs. (12), (13) and the sample-pool bound $K \cdot W < N/q^{\gamma}$.

The resulting time complexity estimates for CRYSTALS-KYBER are listed in Table 2.

Table 2. Complexity comparison between Algorithm 1 and the "lowest achievable estimates" from [41] for CRYSTALS-KYBER. The parameter d_{reg} is the lowest achievable degree of regularity, computed via Eq. (4). All complexities are expressed in bits ($\log_2$).

Scheme	**Complexity**		d_{reg}	**Key Parameters**						
	[41]	Ours		n_{lat}	β_{bkz}	β_{sieve}	γ	K	W	p_{true}
KYBER-512	1584	**724**	306	336	395	395	1	$2^{16.5}$	$2^{48.75}$	0.84
KYBER-768	1588	**967**	232	493	600	600	2	$2^{17.25}$	$2^{76.5}$	0.82
KYBER-1024	2014	**1316**	287	646	850	850	4	$2^{17.75}$	2^{105}	0.80

4 Further Discussion on the Resultant-Based Method

Gröbner bases constitute a general algebraic solving tool, first applied to the Arora-Ge system by Albrecht et al. [2]. That early work, however, relied on the semi-regularity assumption. More recently, Steiner [41] provided a proven method for solving the Arora-Ge system using Gröbner bases. As their analysis of KYBER-768 showed, the proven bound obtained in this way is relatively loose and leaves a significant gap compared to the "lowest achievable estimate" reported in [41]. To partially bridge this gap, we introduce a resultant-based method. In this section, we demonstrate that this resultant method not only synergizes with the dual lattice reduction stage but also stands as an independent algebraic solver of broader interest.

We focus on LWE with a small secret, meaning each coordinate of the secret vector is chosen from a small, fixed interval. Without loss of generality, we assume each secret coordinate is sampled from the integer interval $[\![-T, T]\!]$ and the error is chosen from a larger interval $[\![-T', T']\!]$.

Theorem 3 provides a provable complexity estimate for solving the Arora-Ge system via the resultant-based method. Its proof follows by specializing Theorem 2 to the Arora-Ge system. In the small-secret setting, this bound yields an **exponential speed-up** over the proven estimate in [41]. Hence, the resultant-based method provides a practical and competitive alternative to Gröbner-basis techniques for the Arora-Ge system, thereby broadening the algebraic cryptanalytic toolkit for LWE.

Theorem 3. *Let q be a prime, $m = n \geq 2$ be LWE parameters, and assume that the secret $\mathbf{s}$ and the error $\mathbf{e}$ have coordinates chosen from interval $[\![-T, T]\!]$ and*

$[\![-T', T']\!]$, *respectively. Let* $\mathcal{F}_{\mathrm{AG}}$ *be the corresponding Arora-Ge system. Denote* $d = 2T + 1$. *Then the time complexity, measured in* $\mathbb{Z}_q$*-operations, of solving* $\mathcal{F}_{\mathrm{AG}}$ *by the resultant-based method is upper bounded by*

$$\mathcal{O}\Big(T_{\mathrm{gen}} + (2d-1)!(d-1) + m2^{2d-1}(d-1)\big(v \log v \log\log v + (2d-1)^n\big)\Big), \tag{17}$$

where $v = (2d-1)^{n-1}$ *and* $T_{\mathrm{gen}} = \mathcal{O}\big(m(2T'+1)(d+1)^{n+1}\big)$.

Proof. We specialize the general resultant-based complexity estimate to the Arora-Ge system $\mathcal{F}_{\mathrm{AG}}$. Since each polynomial in $\mathcal{F}_{\mathrm{AG}}$ is the product of $2T'+1$ linear forms and the reduction system has degree $d = 2T+1$, we apply Algorithm 3 after each multiplication by a linear form. Hence the generation cost is

$$T_{\mathrm{gen}} = \mathcal{O}\big(m(2T'+1)(d+1)^{n+1}\big)\,.$$

We next estimate the cost of the successive resultant eliminations. We use the same elimination scheme as in Eq. (16). The worst-case estimate is governed by the cost of the first elimination step. The analysis consists of two components: the cost of a single multiplication-reduction step and the total number of such steps. Eliminating the first variable involves at most $m2^{2d-1}(d-1)$ multiplication-reduction steps. For each such step, the multiplication cost is bounded by

$$\mathcal{O}(v \log v \log\log v),$$

where $v = (2d-1)^{n-1}$, while the reduction cost is bounded by

$$\mathcal{O}\big((2d-1)^n\big).$$

Hence, each multiplication-reduction step costs

$$\mathcal{O}(v \log v \log\log v + (2d-1)^n)\,.$$

In addition, the pre-expanded resultant contributes $\mathcal{O}\big((2d-1)!(d-1)\big)$. Combining the generation cost and the elimination cost gives

$$\mathcal{O}\Big(T_{\mathrm{gen}} + (2d-1)!(d-1) + m2^{2d-1}(d-1)\big(v \log v \log\log v + (2d-1)^n\big)\Big),$$

where

$$T_{\mathrm{gen}} = \mathcal{O}\big(m(2T'+1)\left(nd^n + (d+1)^{n+1}\right)\big)\,, \qquad v = (2d-1)^{n-1}.$$

As in the proof of Theorem 2, the costs of back-substitution and gcd computation are negligible. This proves the claimed bound. □

Under the small-secret setting, we further simplify Eq. (17) to facilitate a clearer complexity comparison. Since T is small, the term $(2d-1)!(d-1)$ is negligible. Moreover, using $2T'+1 < q$, we obtain the cleaner estimate

$$\mathcal{O}\Big(m \cdot 2^{\log q + (n+1)\log(d+1)} + m \cdot n \log n \cdot 2^{\,2d-1+\log(d-1)+n\log(2d-1)}\Big). \tag{18}$$

Since $d = 2T + 1 \geq 3$, we have $2d - 1 > d + 1$. Therefore, the term $m \cdot 2^{\log q + (n+1)\log(d+1)}$ is omitted in the subsequent analysis. Theorem 22 of [41] gives the proven complexity for the same small secret setting,

$$\mathcal{O}(m \cdot (d-1) \cdot n^2 \cdot 2^{\omega \cdot 2^{\frac{1}{\ln(2)}} \cdot (d-1)^{1-\frac{1}{\ln(4)}} \cdot n^{2-\frac{1}{\ln(4)}}}). \tag{19}$$

The dominant term in Eq. (18) scales as 2^n, while the corresponding term in Eq. (19) scales as $2^{n^{2-1/\ln 4}}$. Since $2-1/\ln 4 > 1$, this alone implies an exponential speed-up of our bound over the proven estimate of [41]. Moreover, comparing the coefficients, one can easily verify that $\omega \cdot 2^{1/\ln 2} \cdot (d-1)^{1-1/\ln 4} > \log(2d-1)$ holds for all integers $d \geq 2$, which further substantiates the significant advantage of our method.

As a concrete example, for CRYSTALS-KYBER, since both $\mathbf{s}$ and $\mathbf{e}$ are sampled from the same distribution $\mathcal{B}_\eta$, we have $T' = T$, and hence the term T_{gen} is omitted. Then, we use a more precise complexity formula instead of the asymptotic form Eq. (18):

$$\mathcal{O}\left(\lceil \log q \rceil^2 \cdot m \cdot w_i \cdot \left(v \log v \log\log v + (2d-1)^n\right)\right), \tag{20}$$

where $d = 2T+1$, $v = (2d-1)^{n-1}$, and $w_1 = 2832$ for KYBER-512 and $w_2 = 152$ for KYBER-768/1024.

Table 3 presents the complexity estimates obtained via Eq. (20), alongside the corresponding proven estimates from [41].

Table 3. Complexity of solving the Arora-Ge system using resultant-based method for CRYSTALS-KYBER. Both the proven and optimistic estimates from [41] are provable and are included for comparison. All costs are measured in bits ($\log_2$).

Scheme	**Our Method**	**Proven estimates [41]**	**Optimistic estimates [41]**
KYBER-512	**1933**	4249	4479
KYBER-768	**2471**	5554	4717
KYBER-1024	**3283**	7402	6030

5 Augmenting Algorithm 1 with Hints

At CRYPTO 2025, Cao et al. [17] introduced a general framework that converts arbitrary numbers of perfect, modular, and approximate hints into a lattice problem, together with a complexity estimator for solving LWE with such auxiliary information. The three types of hints are formally defined as

$$\begin{aligned}
&\text{Perfect hint: } \langle \mathbf{s}, \mathbf{v} \rangle = l \in \mathbb{Z}_q, \\
&\text{Modular hint: } \langle \mathbf{s}, \mathbf{v} \rangle \equiv l \mod k, \\
&\text{Approximate hint: } \langle \mathbf{s}, \mathbf{v} \rangle + e_{\text{hint}} = l \in \mathbb{Z}_q.
\end{aligned}$$

In this section, we systematically examine how side information—both individual hints and their combinations—affects the performance of Algorithm 1. We first analyze the impact of each hint type on the algorithm's complexity, and then explore the effect of combining multiple hints, which is of particular importance in practice.

Perfect hints. Adding a perfect hint is straightforward: it introduces a linear equation that directly eliminates one variable from the polynomial system.

Modular hints. A modular hint of the form $\langle \mathbf{s}, \mathbf{v} \rangle - l \in \Omega \subseteq \mathbb{Z}_q$ yields a polynomial whose roots lie in Ω. For $\Omega = \{0, 1\}$ this gives $\big(\langle \mathbf{s}, \mathbf{v} \rangle - l\big)\big(\langle \mathbf{s}, \mathbf{v} \rangle - l - 1\big) = 0$. Adding such constraints to $\mathcal{F}_{\text{new}}$ further shrinks the candidate set, enabling larger n_{lat} and better complexity. A full treatment of modular hints is quite complicated; we instead focus on the special case $s_i \equiv 0 \mod t$ for integer $t < q$.

Approximate hints. One important class of approximate hints consists of Hamming-weight hints, which reveal the Hamming weight of secret coordinates. For example, given $s_1 \in \{-2, -1, 0, 1, 2\}$ and the knowledge that $H(s_1) = 1$, the candidate set is reduced to $\{1, 2\}$. Such a reduction decreases the degree of the corresponding variable in $\mathcal{P}$, directly benefiting the resultant-based method.

The following examples illustrate how hints, individually or in combination, directly benefit Algorithm 1.

Example 1 (Hamming hints). Suppose the secret coordinates lie in $[\![-2, 2]\!]$. If $H(s_i) = 1$ (resp. 2), then $s_i \in \{1, 2\}$ (resp. $\{-1, -2\}$). The degree-d polynomial of $\mathcal{P}$ defined in Eq. (14) can thus be replaced by the quadratic $s_j^2 - 3s_j + 2$ or $s_j^2 + 3s_j + 2$; the degree of each variable in $\mathcal{F}_{\text{new}}$ is then bounded by 2.

Example 2 (Perfect + Hamming hints). Assume we have n_1 perfect hints and that the Hamming weight of the secret is known to be 1. The perfect hints eliminate n_1 variables, and the remaining analysis follows Example 1.

Example 3 (Modular + Hamming hints). Consider m LWE samples $(\mathbf{a}_i, \mathbf{b}_i)$ with a secret $\mathbf{s}$ whose coordinates lie in $[\![-5, 5]\!]$. Suppose we are given both Hamming-weight hints (indicating $H(s_i) = 1$) and modular hints requiring $s_i = 0 \mod 2$. Together, these constraints reduce the number of possible values for each coordinate from 11 to 2, and the subsequent analysis proceeds analogously to Example 1.

6 Conclusions

This work introduces a new two-stage algebraic algorithm for LWE that strategically integrates dual lattice reduction into algebraic analysis. We also demonstrate that the classical Arora-Ge polynomial arises as a special case of our more general construction, thereby revealing a unifying algebraic perspective.

The proposed algorithm consists of two distinct stages. First, a dual lattice reduction stage transforms the original LWE samples into a set of dimension-reduced samples with amplified error. Second, instead of applying Gröbner basis methods to the Arora-Ge system, we construct a new polynomial system tailored to the distribution of the amplified error and solve it via a resultant-based method. This two-stage design is pivotal, as our polynomial construction exhibits improved compatibility with the preceding reduction stage, allowing us to effectively balance the trade-off between dimension reduction and error amplification. Concretely, for the NIST-selected algorithm CRYSTALS-KYBER, our algorithm delivers complexity estimates that are better than the results of solving the Arora-Ge system with Gröbner bases reported in [41].

Independently, this work establishes the resultant-based method as a new, competitive algebraic solver in its own right. When applied directly to the original Arora-Ge system (without dual lattice reduction) in the small secret LWE setting, it delivers a provable bound that exhibits an exponential speed-up over the proven estimates in [41]. We further illustrate how various forms of side information—including perfect, modular, and approximate hints—can be naturally incorporated into the polynomial system to accelerate our two-stage algorithm.

Our work explores whether techniques from dual attacks can meaningfully enhance the algebraic analysis of LWE, thereby narrowing the gap between these two cryptanalytic approaches. The primary goal is not to challenge the established security estimates of standardized LWE-based schemes, but to investigate potential synergies between different analytical paradigms. Although the complexity estimates we obtain for KYBER-512/768/1024 remain substantially higher than those of the best primal or dual attacks, they represent a significant improvement over previous algebraic results. This progress confirms the viability of integrating lattice reduction into the algebraic cryptanalysis framework.

During the course of this research, we identified several promising avenues for advancing the algebraic analysis of LWE:

Open Question 1: Fast Solvability Testing via Resultant. Empirical results in Section 3.2.2 indicate that the system $\mathcal{F}_{\text{new}}$ is very likely to have no solution if it does not contain the true secret. This raises a natural question: can the resultant be leveraged to develop a fast, dedicated test for the solvability of such systems? If feasible, one could set the probability that a system contains the true secret to a very low value, repeat the construction many times, and solve only those systems that pass the solvability test. This approach could lead to a substantial reduction in the overall computational cost.

Open Question 2: Leveraging Additional Samples. Our analysis assumes the standard setting $m = n$. Two natural extensions arise: whether and how extra samples can improve efficiency, and how many samples are needed for the algebraic approach to meet the NIST security levels for CRYSTALS-KYBER. Both questions are of direct relevance from a designer's worst-case perspective.

In summary, this paper helps bridge the gap where algebraic analysis lagged far behind dual attacks for LWE, offering improved complexity estimates and presenting a refined algebraic solver. We hope it stimulates further work on the algebraic analysis of LWE.

References

1. Albrecht, M.R.: On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 103–129. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_4
2. Albrecht, M.R., Cid, C., Faugère, J.C., Fitzpatrick, R., Perret, L.: Algebraic algorithms for LWE problems. ACM Commun. Comput. Algebra **49**(2), 62 (2015). https://doi.org/10.1145/2815111.2815158
3. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015). https://doi.org/10.1515/jmc-2015-0016
4. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.P.: A new algorithm for solving the rank syndrome decoding problem. In: 2018 IEEE International Symposium on Information Theory (ISIT). pp. 2421–2425. IEEE (2018)
5. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22006-7_34
6. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Tillich, J.P.: Revisiting algebraic attacks on MinRank and on the rank decoding problem. Designs, Codes and Cryptography **91**(11), 3671–3707 (2023). https://doi.org/10.1007/s10623-023-01265-x
7. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Improvements of algebraic attacks for solving the rank decoding and minrank problems. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. pp. 507–536. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_17
8. Bardet, M., Faugère, J.C., Salvy, B.: On the complexity of gröbner basis computation of semi-regular overdetermined algebraic equations. In: Proceedings of the International Conference on Polynomial System Solving. pp. 71–74 (2004)
9. Bariant, A., Boeuf, A., Briaud, P., Hostettler, M., Øygarden, M., Raddum, H.: Improved resultant attack against Arithmetization-Oriented primitives. In: Tauman Kalai, Y., Kamara, S.F. (eds.) CRYPTO 2025. LNCS, vol. 16004, pp. 335–367. Springer, Cham (2025). https://doi.org/10.1007/978-3-032-01901-1_11
10. Bariant, A., Boeuf, A., Briaud, P., Hostettler, M., Øygarden, M., Raddum, H.: Improved resultant attack against Arithmetization-Oriented primitives. Cryptology ePrint Archive, Paper 2025/259 (2025), https://eprint.iacr.org/2025/259
11. Bariant, A., Bouvier, C., Leurent, G., Perrin, L.: Algebraic attacks against some Arithmetization-Oriented primitives. IACR Transactions on Symmetric Cryptology **2022**(3), 73–101 (2022). https://doi.org/10.46586/tosc.v2022.i3.73-101
12. Beullens, W.: Breaking rainbow takes a weekend on a laptop. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. pp. 464–479. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15979-4_16

13. Briaud, P., Bros, M., Perlner, R., Smith-Tone, D.: Practical attack on all parameters of the DME signature scheme. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024. LNCS, vol. 14657, pp. 3–29. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58754-2_1
14. Briaud, P., Bros, M., Perlner, R., Smith-Tone, D.: Practical attack on all parameters of the HPPC signature scheme. In: SAC 2025. pp. 519–538. Springer-Verlag, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-3-032-10536-3_19
15. Briaud, P., Øygarden, M.: A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14008, pp. 391–422. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_14
16. Caminata, A., Gorla, E.: Solving multivariate polynomial systems and an invariant from commutative algebra. In: Bajard, J.C., Topuzoğlu, A. (eds.) WAIFI 2020: Arithmetic of Finite Fields. LNCS, vol. 12542, pp. 3–36. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68869-1_1
17. Cao, J., Jiang, H., Cheng, Q.: Refined attack on LWE with hints: Constructing lattice via gaussian elimination. In: Tauman Kalai, Y., Kamara, S.F. (eds.) CRYPTO 2025. LNCS, vol. 16000, pp. 385–416. Springer, Cham (2025). https://doi.org/10.1007/978-3-032-01855-7_13
18. Carrier, K., Meyer-Hilfiger, C., Shen, Y., Tillich, J.P.: Assessing the impact of a variant of MATZOV's dual attack on Kyber. In: Tauman Kalai, Y., Kamara, S.F. (eds.) CRYPTO 2025. LNCS, vol. 16000, pp. 444–476. Springer, Cham (2025). https://doi.org/10.1007/978-3-032-01855-7_15
19. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_1
20. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, 2 edn. (1997). https://doi.org/10.1007/978-1-4757-2693-0
21. Cueto Noval, M., Merz, S.P., Stählin, P., Ünal, A.: On the soundness of algebraic attacks against code-based assumptions. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025. LNCS, vol. 15606, pp. 385–415. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-91095-1_14
22. Ducas, L., Pulles, L.N.: Does the dual-sieve attack on learning with errors even work? In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023. LNCS, vol. 14083, pp. 37–69. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-38548-3_2
23. Ducas, L., Pulles, L.N.: Accurate score prediction for dual-sieve attacks. Journal of Cryptology **39**(1), 8 (2026). https://doi.org/10.1007/s00145-025-09560-7
24. Espitau, T., Joux, A., Kharchenko, N.: On a dual/hybrid approach to small secret LWE. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) INDOCRYPT 2020. LNCS, vol. 12578, pp. 440–462. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-65277-7_20
25. Esser, A., Verbel, J., Villanueva-Polanco, R.: Rank syndrome decoding estimator - an asymptotic and concrete analysis. Cryptology ePrint Archive, Paper 2026/082 (2026), https://eprint.iacr.org/2026/082
26. Fröberg, R.: An inequality for Hilbert series of graded algebras. Mathematica Scandinavica **56**(2), 117–144 (1985), https://www.jstor.org/stable/24491560
27. Gao, S.: Counting zeros over finite fields using Gröbner bases. Ph.D. thesis, Carnegie Mellon University (2009)

28. Guo, Q., Johansson, T.: Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13093, pp. 33–62. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92068-5_2
29. Lyubashevsky, V., et al.: CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology (2022), https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
30. MATZOV: Report on the security of LWE: Improved dual lattice attack (2022), https://zenodo.org/record/6412487
31. Micciancio, D., Regev, O.: Lattice-based Cryptography, pp. 147–191. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5
32. Moenck, R.T.: Practical fast polynomial multiplication. In: Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation. pp. 136–148. SYMSAC '76, Association for Computing Machinery, New York, NY, USA (1976). https://doi.org/10.1145/800205.806332
33. Nguyen, P.Q.: Hermite's constant and lattice algorithms, pp. 19–69. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-02295-1_2
34. Pouly, A., Shen, Y.: Provable dual attacks on learning with errors. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024. LNCS, vol. 14657, pp. 256–285. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58754-2_10
35. Qu, H., Xu, G.: On the provable dual attack for LWE by modulus switching. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025. LNCS, vol. 16247, pp. 34–64. Springer, Singapore (2026). https://doi.org/10.1007/978-981-95-5099-9_2
36. Ran, L.: Wedges, oil, and vinegar. In: Daemen, J., Thomé, E. (eds.) EUROCRYPT 2026. LNCS, vol. 16544, pp. 363–388. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-25327-9_13
37. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing. pp. 84–93. ACM Press, Baltimore, MA, USA (2005). https://doi.org/10.1145/1060590.1060603
38. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming **66**(1), 181–199 (1994). https://doi.org/10.1007/BF01581144
39. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 145–156. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/3-540-36494-3_14
40. Schwabe, P., et al.: CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology (2022), https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
41. Steiner, M.J.: The complexity of algebraic algorithms for LWE. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024. LNCS, vol. 14653, pp. 375–403. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58734-4_13
42. Sun, C., Tibouchi, M., Abe, M.: Revisiting the hardness of binary error LWE. In: Liu, J.K., Cui, H. (eds.) Information Security and Privacy. LNCS, vol. 12248, pp. 425–444. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55304-3_22
43. Yang, H.S., Zheng, Q.X., Yang, J.: Algebraic cryptanalysis of AO primitives based on polynomial decomposition. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025. LNCS, vol. 16245, pp. 383–410. Springer, Singapore (2026). https://doi.org/10.1007/978-981-95-5018-0_13

44. Yang, H.S., Zheng, Q.X., Yang, J., Liu, Q.F., Tang, D.: A new security evaluation method based on resultant for Arithmetic-Oriented algorithms. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024. LNCS, vol. 15490, pp. 457–489. Springer, Singapore (2025). https://doi.org/10.1007/978-981-96-0941-3_15
45. Yu, Y., Ducas, L.: Second order statistical behavior of LLL and BKZ. In: Adams, C., Camenisch, J. (eds.) Selected Areas in Cryptography – SAC 2017. LNCS, vol. 10719, pp. 3–22. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72565-9_1