Algebraic Analysis of Homomorphic Trace Evaluation and Its Applications

Han Xia

Xi'an Jiaotong-Liverpool University, Suzhou, China

Abstract. Field trace evaluation has emerged as a powerful tool in fully homomorphic encryption, with broad applications ranging from bootstrapping algorithms to privacy-preserving protocols. Recent advances have significantly reduced its noise growth by combining tower-based evaluation strategies with rescaling operations. However, existing analyses rely on uniform noise bounds that fail to capture the actual noise behavior across different coefficients, leading to substantial gaps between theoretical estimates and empirical observations.

In this work, we present a refined algebraic analysis of trace evaluation over power-of-two cyclotomics that uncovers structured cancellation effects among noise coefficients induced by subsequent linear operators, in particular the trace mappings of subextensions. We show that, except for the constant term, the variance of each output noise coefficient depends on the 2-adic valuation of its index, yielding bounds that improve upon prior uniform estimates by a factor of $O(\log n)$ both for non-constant coefficients and after a subsequent plaintext-ciphertext multiplication, where n is the ring degree. We further extend our analysis to two typical algorithmic applications of trace evaluation. For ciphertext packing, we derive a non-recursive formulation that admits a cleaner structure and slightly tighter noise estimates. For coefficient extraction, our coefficient-wise analysis improves upon prior uniform variance bounds by factors ranging from $\Theta(n)$ to $\Theta(n^2)$ for non-constant coefficients and by a factor of $O(n)$ after post-multiplication. Experimental results confirm that the observed noise variances follow the coefficient-wise pattern predicted by our analysis and demonstrate pronounced improvements over existing estimates, providing effective guidance for parameter selection and system configuration in practice.

Keywords: Homomorphic Encryption · Field Trace Evaluation · Noise Analysis

1 Introduction

Since the groundbreaking work of Gentry [Gen09], fully homomorphic encryption (FHE) has become a powerful cryptographic tool that allows arbitrary computation on encrypted data without decryption. To avoid the less standard assumptions underlying Gentry's original construction, Brakerski and Vaikuntanathan [BV11a] introduced a more efficient FHE scheme based on the Learning With Errors (LWE) problem [Reg05], which has been further extended in subsequent works [BV11b, BGV12] through optimizations including the use of algebraically structured LWE such as Ring-LWE [SSTX09, LPR10] and General-LWE [BGV12]. Since then, algebraically structured LWE has become the most popular foundation for constructing efficient FHE schemes [FV12, DM15, CGGI16, CKKS17].

In algebraically structured LWE, the underlying basic object is typically taken to be the ring of integers R of a cyclotomic field K, which has been extensively studied in algebraic number theory [Lan90, Was97]. One notable improvement from using cyclotomic rings is

E-mail: hx@xiahan.net (Han Xia)

This work is licensed under a "CC BY 4.0" license.
Date of this document: 2026-08-04.

the ability to support packed plaintexts, where multiple messages can be encoded into a single plaintext in R and processed simultaneously. This can be achieved by leveraging the Chinese Remainder Theorem (CRT) decomposition of R/tR into a direct product of smaller fields or rings where t is the plaintext modulus [SV14, BGV12, GHS12b, CIV18], or by utilizing the canonical embedding of K into $\mathbb{C}^n$ [CKKS17], where n is the degree of the field extension $K/\mathbb{Q}$. Both approaches are typical ways to enable SIMD-style operations on messages encoded in the corresponding slots, thereby significantly enhancing the throughput of the underlying FHE schemes.

Automorphism Evaluation. Beyond basic homomorphic operations like addition and multiplication, another crucial operation enabled by the algebraic structure of cyclotomic rings is the evaluation of automorphisms in the Galois group of $K/\mathbb{Q}$. These automorphisms can be homomorphically evaluated on ciphertexts [BGV12, GHS12b], allowing for advanced functionalities such as homomorphic slot-wise rotations in packed plaintexts [BGV12, SV14, HS14, CKKS17], which are essential for evaluating linear transformations required in bootstrapping [GHS12a, HS15, CHK+18, HS18, BMTH21, MHWW24]. Additionally, in FHEW/TFHE-type schemes [DM15, CGGI20], automorphisms can be used to multiply the exponent of a primitive q-th root of unity by units modulo q, an operation underlying many efficient bootstrapping algorithms [BDF18, LMK+23, XZD+23, GPvL23, DKMS24, WWL+24, LLW+24, XLW24, BJ25, PMH+25, GP25, ZWC25]. A distinct recent direction realizes bootstrapping directly over the automorphism group [HMS25].

Trace Evaluation. The feasibility of automorphism evaluation naturally extends to the evaluation of the trace mapping from K to its subfield K', which is a K'-linear operator defined as the sum of the Galois conjugates over K'. This technique was first used to switch the underlying algebraic context in BGV-type bootstrapping [GHS12a, GHPS13, AP13] and was later leveraged to develop efficient algorithms for coefficient extraction [ACLS18] and ciphertext packing [CDKS21]. These techniques have since been further developed and used as building blocks in a broad range of cryptographic constructions and applications [GH19, CCR19, IY21, ALP+21, YXS+21, MCR21, MW22, RSG+22, KLD+23, LHZ+23, RCG+23, KDE+24, BZZ+24, ADDG24, MW24, BMW24, HYT+24, DFH+24, ACJ24, XW24a, XW24b, LHG+25, IB25, BZS+25, CR25, LWWC25, CGL+25, BI26, KS26, HZD+26, MPSY26, DXQ+26, LLT+26]. More recently, trace evaluation has emerged as a useful building block for bootstrapping leveled FHE schemes [OPP23, Gee24, GV26]. It has also found use in a variety of FHEW/TFHE-type bootstrapping algorithms, including batch bootstrapping [LW23a, LW23b, LWZ+26], functional bootstrapping [XLW24, XW25, LSL+25], and circuit bootstrapping [WWL+24, WHS+25, LY26].

Efficient Algorithms. In the FHE literature, Alperin-Sheriff and Peikert [AP13] first suggested exploiting the tower structure of field extensions together with the transitivity to evaluate the trace mapping more efficiently. An explicit algorithm for the case of power-of-two cyclotomic fields was later provided in [CDKS21]. For a power-of-two cyclotomic field $K = \mathbb{Q}(\zeta_{2n})$ where ζ_{2n} is a primitive $2n$-th root of unity, the CDKS algorithm evaluates the trace mapping along the tower $K/\mathbb{Q}(\zeta_n)/\mathbb{Q}(\zeta_{n/2})/\cdots/\mathbb{Q}$. Since the relative trace of each degree-two extension in the tower requires only a single non-trivial automorphism evaluation, the total cost of evaluating the trace $\mathsf{Tr}_{K/\mathbb{Q}}$ is $\log n$ automorphism evaluations. Recall that for any $a \in K$, the trace satisfies $\mathsf{Tr}_{K/\mathbb{Q}}(a) = n \cdot a_0$ where a_0 is the constant coefficient of a. Consequently, after trace evaluation, both the plaintext and the noise are amplified by a factor of n, a phenomenon commonly referred to as the *phase amplification*, which can be undesirable in certain applications. The original work [CDKS21] proposed to address this issue by multiplying the ciphertext by $n^{-1} \in \mathbb{Z}_q^*$ prior to trace evaluation, where q is the ciphertext modulus. However, this approach does not support the case where $n \notin \mathbb{Z}_q^*$, which is precisely the setting adopted by high-performance FFT-based libraries

such as TFHE-rs [Zam22], where q is chosen to be a word-size power of two. To overcome this limitation, the work [WHS+25] proposed applying a modulus switching that rescales the ciphertext by a factor of $1/n$, followed by a modulus raising operation to recover the original modulus. In this way, the factor n introduced by trace evaluation and the factor $1/n$ introduced by modulus switching cancel out, thereby avoiding phase amplification. Very recently, Lee and Yoon [LY26] further optimized the approach of [WHS+25] by splitting the rescaling factor $1/n$ into $\log n$ successive rescalings by $1/2$, with one rescaling performed at each level of a modified tower. This strategy significantly reduces the additive noise growth from $O(n^3)$ to $O(n \log n)$. Additionally, efficient trace evaluation for non-power-of-two cyclotomic extensions has also been studied [ZLW25, XLW24, OPP25, CKL26].

This Work. In this work, we focus on the analysis of trace evaluation over power-of-two cyclotomics and its applications. Although the work of Lee and Yoon [LY26] significantly reduces the additive noise growth, their analysis still provides a uniform upper bound and does not distinguish among different coefficients of the noise term. As a result, these estimates fail to accurately capture the actual noise behavior across coefficients, especially for non-constant terms. This loss of tightness becomes particularly evident in applications that rely on trace evaluation, yet the mechanisms behind it are not fully understood. For instance, as reported in [MW22], the experimentally observed noise magnitude exhibits a substantial gap compared to the theoretical estimates. We show that such untightness arises in part from an algebraic effect not captured by prior uniform analyses:

> *Noise terms introduced during homomorphic evaluation are also acted upon by subsequent algebraic operators, leading to structured coefficient cancellation.*

As a minimal illustrative example, consider the successive homomorphic evaluation of three linear operators τ_1, τ_2, and τ_3 on a ring-based ciphertext $\boldsymbol{c}$ with initial noise e:

$$\boldsymbol{c} \xrightarrow{\mathsf{Hom.Eval\ of\ }\tau_1} \boldsymbol{c}_1 \xrightarrow{\mathsf{Hom.Eval\ of\ }\tau_2} \boldsymbol{c}_2 \xrightarrow{\mathsf{Hom.Eval\ of\ }\tau_3} \boldsymbol{c}_3$$

Suppose that the evaluation of τ_1, τ_2, and τ_3 introduces additive noise terms e_1, e_2, and e_3, respectively. Prior analyses (e.g., [CDKS21, WHS+25, LY26]) use scalar recurrences to reduce the cumulative action of subsequent operators on each newly introduced noise term to scalar amplification; our analysis instead retains their exact algebraic action. This leads to the following contrast for the estimated noise in the final ciphertext $\boldsymbol{c}_3$:

- *Prior Estimate:* $\tau_3 \circ \tau_2 \circ \tau_1(e) + k_1 e_1 + k_2 e_2 + e_3$ for some amplification factors k_1, k_2.
- *Actual Behavior Tracked in Our Analysis:* $\tau_3 \circ \tau_2 \circ \tau_1(e) + \tau_3 \circ \tau_2(e_1) + \tau_3(e_2) + e_3$.

In homomorphic trace evaluation, the compositions of the relevant linear operators yield the trace mappings of subextensions, leading to widespread cancellation of non-constant coefficients. Motivated by this observation, we develop a coefficient-wise noise analysis that explicitly tracks the effect of algebraic operators on each coefficient, leading to tighter coefficient-wise noise estimates for trace evaluation and its applications.

Our Results. By accounting for the effect of algebraic operators on overall noise growth, we derive tighter noise estimates for trace evaluation and its applications to ciphertext packing and coefficient extraction. Our main findings are summarized below.

- *Trace Evaluation:* We show that the noise variances of all non-constant coefficients depend on the 2-adic valuations of their indices, yielding an improvement over prior uniform estimates [LY26] by a factor of $O(\log n)$ both for the non-constant output noise coefficients and after a subsequent plaintext-ciphertext multiplication.
- *Ciphertext Packing:* By reanalyzing the algebraic structure of the algorithm, we propose a non-recursive version of the packing procedure in [LY26], which improves

upon [CDKS21]. This formulation provides a more precise treatment of the overall noise, yielding a slightly tighter variance bound than [LY26]. Although this variance tightening may be minor in practice, our non-recursive approach offers a cleaner formulation that may simplify implementation and facilitate further system-level optimization.

- *Coefficient Extraction:* Similar to trace evaluation, we prove that the noise variances of all non-constant coefficients in each output ciphertext depend on the 2-adic valuation of their indices and decrease as the valuations increase. Our coefficient-wise variance bounds for non-constant coefficients improve upon prior uniform estimates [MW22, BMW24] by factors ranging from $\Theta(n)$ to $\Theta(n^2)$. Consequently, after plaintext-ciphertext multiplication, our full variance bound improves upon the corresponding bound derived from prior uniform estimates by a factor of $O(n)$.

Finally, we provide experimental validation for trace evaluation and coefficient extraction, where the analytical improvements are pronounced. The observed noise variances are consistent with our theoretical predictions, supporting both the improvements over prior work and the specific coefficient-wise noise patterns captured by our analysis.

Applications. We briefly mention two representative applications for which our results can be directly applied within existing frameworks developed recently:

- *Circuit Bootstrapping:* The tighter bounds may improve decryption failure estimates, enable more favorable parameter choices, or certify a larger remaining post-bootstrapping noise margin for subsequent homomorphic evaluations [WWL+24, WHS+25, LY26].
- *Private Information Retrieval:* In FHE-based PIR protocols that employ coefficient extraction to expand compressed queries [ACLS18, ALP+21, MCR21, MW22, BMW24], the tighter analysis can support more flexible configurations of the underlying cryptosystem and may thereby improve key performance metrics.

Beyond these two potential use cases, our results may also be relevant to other applications built upon trace evaluation, ciphertext packing, or coefficient extraction. More broadly, our coefficient-wise algebraic analysis shows that tracking how algebraic operators affect noise throughout homomorphic evaluation can yield tighter bounds. This perspective may inspire improved design and analysis of other FHE techniques.

Roadmap. The remainder of the paper is organized as follows. Section 2 provides the necessary background on cyclotomics and homomorphic encryption. We revisit existing trace evaluation algorithms in Section 3 and develop our refined algebraic analysis in Section 4. Section 5 extends this algebraic methodology to the analysis of ciphertext packing and coefficient extraction, while Section 6 reports our experimental validation. Finally, Section 7 concludes the paper with remarks on limitations and future directions.

2 Preliminaries

Notation. We denote by $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ the sets of integers, rational numbers, real numbers, and complex numbers, respectively. For a positive integer q, we use $\mathbb{Z}_q$ to represent the quotient ring $\mathbb{Z}/q\mathbb{Z}$, with two typical complete systems of representatives $[q] := \{0, 1, \ldots, q-1\}$ and $[q]_\pm := \mathbb{Z} \cap (-q/2, q/2]$. For $x \in \mathbb{Z}$, we use $[x]_q$ to denote modulo q reduction of x into the set $[q]$. We write $\mathbb{Z}_q^*$ for the group of units of $\mathbb{Z}_q$, whose elements are represented by those $a \in [q]$ with $\gcd(a, q) = 1$. For $x \in \mathbb{R}$, $\lfloor x \rfloor$, $\lceil x \rceil$, and $\lfloor x \rceil := \lfloor x + 1/2 \rfloor$ denote floor, ceiling, and nearest-integer rounding, respectively. We write $(v_1, \ldots, v_n)$ for a row vector and $(v_1, \ldots, v_n)^\top$ for the corresponding column vector, and use $\langle \cdot, \cdot \rangle$ to denote the vector inner product. We use $a \leftarrow \mathbb{A}$ to denote that a is sampled uniformly from the finite set $\mathbb{A}$, and $a \leftarrow \chi$ to denote sampling from the distribution χ. Unless otherwise specified, the logarithm $\log(\cdot)$ is taken to base 2.

2.1 Cyclotomics

In this work, we focus on cyclotomic extensions with power-of-two indices.

Definition. Let ζ_{2n} be a primitive $2n$-th root of unity where n is a power of two. The cyclotomics with index $2n$ are defined as follows:

- *Cyclotomic field:* $K := \mathbb{Q}(\zeta_{2n}) \cong \mathbb{Q}[X]/(X^n+1)$ where X^n+1 is the minimal polynomial of ζ_{2n} over $\mathbb{Q}$ called the $2n$-th cyclotomic polynomial.
- *Cyclotomic ring:* $R := \mathbb{Z}[\zeta_{2n}] \cong \mathbb{Z}[X]/(X^n+1)$, which is the ring of integers $\mathcal{O}_K$ of K. For a positive integer q, we denote by R_q the quotient ring $R/qR \cong \mathbb{Z}_q[X]/(X^n+1)$.

The ring R is a free $\mathbb{Z}$-module of rank n, so it has $\mathbb{Z}$-bases of the form $\{b_i\}_{i\in[n]}$ where $b_i \in R$; these are also $\mathbb{Q}$-bases of K called integral bases. One such basis is the power basis $\{1, \zeta_{2n}, \ldots, \zeta_{2n}^{n-1}\}$. For any $x \in K$, we can write $x = \sum_{i=0}^{n-1} x_i \cdot \zeta_{2n}^i$ with $x_i \in \mathbb{Q}$. We refer to $\{x_i\}_{i\in[n]}$ as the coefficients of x (with respect to the power basis). The ℓ_p-norm of x is defined as $\|x\|_p := \left(\sum_{i=0}^{n-1} |x_i|^p\right)^{1/p}$ for a positive integer $p < \infty$ and $\|x\|_\infty := \max_{i\in[n]} |x_i|$ for $p = \infty$. We denote the coefficient-wise rounding of x by $\lfloor x \rceil := \sum_{i=0}^{n-1} \lfloor x_i \rceil \cdot \zeta_{2n}^i$.

Field Trace. Let K be a cyclotomic extension of F with $K = F(\zeta_d)$ and $[K : F] = \varphi(d)$. Then K/F is Galois with Galois group $\mathsf{Gal}(K/F) \cong \mathbb{Z}_d^*$, which consists of all automorphisms of K that fix F, i.e., the automorphisms given by $\sigma_j : \ \zeta_d \mapsto \zeta_d^j$ for all $j \in \mathbb{Z}_d^*$. The field trace with respect to K/F is an F-linear map from K onto F defined as $\mathsf{Tr}_{K/F}(x) = \sum_{\sigma\in\mathsf{Gal}(K/F)} \sigma(x)$ for $x \in K$, which exhibits the following properties:

- *Integrality:* For any $x \in \mathcal{O}_K$, $\mathsf{Tr}_{K/F}(x) \in \mathcal{O}_F$.
- *Linearity:* For any $x, y \in K$ and $\alpha, \beta \in F$, $\mathsf{Tr}_{K/F}(\alpha x + \beta y) = \alpha\mathsf{Tr}_{K/F}(x) + \beta\mathsf{Tr}_{K/F}(y)$.
- *Transitivity:* For a tower of cyclotomic extensions $K/E/F$, $\mathsf{Tr}_{K/F} = \mathsf{Tr}_{E/F} \circ \mathsf{Tr}_{K/E}$.

For computing the field trace with respect to cyclotomic extensions with power-of-two indices, we have the following useful lemma.

Lemma 1. *Let n, m be powers of two such that $m \mid n$. Let $K = \mathbb{Q}(\zeta_{2n})$ and $L = \mathbb{Q}(\zeta_{2m})$ be cyclotomic fields. For any element $a \in K$ expressed in the power basis as $a = \sum_{i=0}^{n/m-1} a^{(i)}\zeta_{2n}^i$ where $a^{(i)} \in L$, the relative trace satisfies $\mathsf{Tr}_{K/L}(a) = \frac{n}{m}a^{(0)}$.*

Proof. Let $d = [K : L] = n/m$. Since K/L is a Galois extension, the relative trace is given by $\mathsf{Tr}_{K/L}(x) = \sum_{\sigma\in\mathsf{Gal}(K/L)} \sigma(x)$. Every $\mathbb{Q}$-automorphism of K is of the form $\sigma_r : \zeta_{2n} \mapsto \zeta_{2n}^r$ for some $r \in \mathbb{Z}_{2n}^*$. Since $L = \mathbb{Q}(\zeta_{2m})$ with $\zeta_{2m} = \zeta_{2n}^d$, the automorphism σ_r fixes L if and only if $\sigma_r(\zeta_{2m}) = \zeta_{2m}$, equivalently, $\zeta_{2n}^{d(r-1)} = 1$, or $r \equiv 1 \pmod{2m}$. Therefore, the Galois group $\mathsf{Gal}(K/L)$ consists of automorphisms

$$\sigma_j : \ \zeta_{2n} \longmapsto \zeta_{2n}^{1+2mj}, \qquad j = 0, \ldots, d-1.$$

Consequently, for $1 \le i < d$, we have

$$\mathsf{Tr}_{K/L}(\zeta_{2n}^i) = \sum_{j\in[d]} \zeta_{2n}^{i(1+2mj)} = \zeta_{2n}^i \sum_{j\in[d]} \left(\zeta_{2n}^{2m}\right)^{ij} = \zeta_{2n}^i \sum_{j\in[d]} \left(\zeta_d^i\right)^j.$$

Since $1 \le i < d$, we have $\zeta_d^i \neq 1$, and thus the above geometric series vanishes, yielding $\mathsf{Tr}_{K/L}(\zeta_{2n}^i) = 0$. For $i = 0$, we clearly have $\mathsf{Tr}_{K/L}(1) = [K : L] = d$. By the L-linearity of the trace map, it follows that

$$\mathsf{Tr}_{K/L}(a) = \sum_{i\in[d]} a^{(i)} \cdot \mathsf{Tr}_{K/L}(\zeta_{2n}^i) = d \cdot a^{(0)} + 0 = \frac{n}{m}\, a^{(0)}. \qquad \square$$

2.2 Basic Encryption Scheme

Now, we review the basics of the encryption scheme used in this work. We begin with the Ring Learning with Errors (RLWE) problem [SSTX09, LPR10], which is the underlying problem known to be at least as hard as certain computational problems on ideal lattices.

Definition 1 (Primal-RLWE)**.** For a security parameter λ, let $n := n(\lambda)$ be the degree of the ring, let $q = q(\lambda) \geq 2$ be an integer modulus, and let $\chi = \chi(\lambda)$ be an error distribution over R. For $s \in R_q$, the decision version of (primal-)RLWE is to distinguish between the following two sets of samples: (1) RLWE samples $(a, b) \in R_q^2$ where $a \leftarrow R_q$ and $b = a \cdot s + e \pmod q$ for some $e \leftarrow \chi$, and (2) uniform samples $(a, b) \leftarrow R_q^2$.

RLWE-Based Encryption Scheme. A typical RLWE-based symmetric-key encryption scheme consists of the following algorithms:

- $\mathsf{ParamGen}(1^\lambda)$: The parameter generation algorithm takes as input the security parameter λ and outputs a parameter tuple $\mathtt{params} := (n, q, \chi_s, \chi_e)$ where n is the ring degree, q is the ciphertext modulus, χ_s is the secret key distribution and χ_e is the noise distribution (typically a discrete Gaussian distribution).
- $\mathsf{KeyGen}(\mathtt{params})$: The key generation algorithm takes as input the parameter tuple $\mathtt{params}$ and outputs a secret key $\mathtt{sk} := (-s, 1) \in R_q^2$ where $s \leftarrow \chi_s$.
- $\mathsf{Enc}(\mathtt{sk}, m, p)$: The encryption algorithm takes as input the secret key $\mathtt{sk}$, a message $m \in R_p$ and a plaintext modulus p, and outputs a ciphertext $\boldsymbol{c} := (a, b) \in R_q^2$ where $a \leftarrow R_q$ and $b = a \cdot s + e + \Delta \cdot m \pmod q$ for some $e \leftarrow \chi_e$ and $\Delta := \lfloor q/p \rceil$.
- $\mathsf{Dec}(\mathtt{sk}, \boldsymbol{c}, p)$: The decryption algorithm takes as input the secret key $\mathtt{sk}$, a ciphertext $\boldsymbol{c} \in R_q^2$, and a plaintext modulus p, then computes the *phase* $\varphi = \langle \boldsymbol{c}, \mathtt{sk} \rangle \bmod q$ and outputs $m' = \lfloor (p/q) \cdot \varphi \rceil \bmod p$.

For convenience, we denote the set of all valid RLWE encryptions of the message m under the secret key $\mathtt{sk}$ and scaling factor Δ as $\mathsf{RLWE}_{\mathtt{sk}}(\Delta \cdot m)$. When the scaling factor Δ is implicit in m or absent, we denote such RLWE encryptions or encodings as $\mathsf{RLWE}_{\mathtt{sk}}(m)$.

Definition 2 (Noise of RLWE Ciphertexts)**.** For a ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$, the noise term of $\boldsymbol{c}$ is defined as $\mathtt{Err}(\boldsymbol{c}) := \langle \boldsymbol{c}, \mathtt{sk} \rangle - m \pmod q$ with coefficients in $[q]_\pm$.

2.3 Probabilistic Analysis

Considering that the noise in a freshly generated RLWE ciphertext is typically sampled from a discrete Gaussian distribution, variance provides a convenient and conventional measure of noise growth. In particular, if X and Y are independent random variables with variances V_X and V_Y, respectively, then, for any $a, b \in \mathbb{R}$, $\mathsf{Var}(aX + bY) = a^2 V_X + b^2 V_Y$. Based on this fact, we will model the coefficients of the noise terms in RLWE ciphertexts as random variables and use the following definitions and lemmas in our noise analysis.

Definition 3 (Variance of Coefficients)**.** For a ring element $a = \sum_{i=0}^{n-1} a_i \cdot \zeta_{2n}^i \in R$, we denote the variance of its i-th coefficient a_i as $\mathsf{Var}_i(a)$. In particular, when all coefficients of a are assumed to have the same variance, we simply denote it as $\mathsf{Var}(a)$.

Lemma 2. *Let $a, b \in R$ be two ring elements with coefficients a_i and b_i independent for all $i \in [n]$. Then, $\mathsf{Var}_i(a + b) = \mathsf{Var}_i(a) + \mathsf{Var}_i(b)$ for all $i \in [n]$.*

Proof. Let $a = \sum_{i \in [n]} a_i \cdot \zeta_{2n}^i$ and $b = \sum_{i \in [n]} b_i \cdot \zeta_{2n}^i$, then $a + b = \sum_{i \in [n]} (a_i + b_i) \cdot \zeta_{2n}^i$. The result follows from the independence of a_i and b_i for all $i \in [n]$. □

For ring multiplication, we consider the case where one of the ring elements has bounded coefficients, as formalized in the following lemma.

Lemma 3. *Let $a, b \in R$, and suppose the coefficients of a are pairwise independent. Then*

- *If $\mathsf{Var}_i(a) = \mathsf{Var}(a)$ for all $i \in [n]$, then $\mathsf{Var}_i(a \cdot b) = \|b\|_2^2 \cdot \mathsf{Var}(a)$ for all $i \in [n]$.*
- *If $\|b\|_\infty \le B$, then $\mathsf{Var}_i(a \cdot b) \le B^2 \cdot \sum_{j\in[n]} \mathsf{Var}_j(a)$ for all $i \in [n]$.*

Proof. Let $a = \sum_{i\in[n]} a_i \cdot \zeta_{2n}^i$ and $b = \sum_{i\in[n]} b_i \cdot \zeta_{2n}^i$, then we can write $a \cdot b = \sum_{i\in[n]} c_i \cdot \zeta_{2n}^i$ where $c_i = \sum_{j\in[n]} (-1)^{e_{i,j}} a_j b_{[i-j]_n}$ for some $e_{i,j} \in \{0,1\}$. Then $\mathsf{Var}_i(a \cdot b) = \mathsf{Var}(c_i)$, and

- If $\mathsf{Var}_i(a) = \mathsf{Var}(a)$ for all $i \in [n]$, then the fact $\{[i-j]_n\}_{j\in[n]} = [n]$ implies

$$\mathsf{Var}_i(a \cdot b) = \sum_{j\in[n]} \left((-1)^{e_{i,j}} b_{[i-j]_n}\right)^2 \cdot \mathsf{Var}(a_j) = \sum_{j\in[n]} b_j^2 \cdot \mathsf{Var}(a) = \|b\|_2^2 \cdot \mathsf{Var}(a).$$

- If $\|b\|_\infty \le B$, then $\left|(-1)^{e_{i,j}} b_{[i-j]_n}\right| \le B$ for all $i, j \in [n]$, and hence

$$\mathsf{Var}_i(a \cdot b) \le B^2 \cdot \sum_{j\in[n]} \mathsf{Var}(a_j) = B^2 \cdot \sum_{j\in[n]} \mathsf{Var}_j(a). \qquad \square$$

2.4 Homomorphic Operations

We now review the basic homomorphic operations that will be used in this work.

Linear Homomorphisms of RLWE. RLWE-based ciphertexts support basic linear homomorphisms, namely addition and scalar multiplication (with small-norm ring elements). The following lemma summarizes this property and the associated noise analysis.

Lemma 4. *Let $\boldsymbol{c}_i \in \mathsf{RLWE}_{\mathtt{sk}}(m_i)$ with independent $\mathtt{Err}(\boldsymbol{c}_i)$ for $i \in \{0,1\}$. Then we have*

- *Homomorphic addition: Let $\boldsymbol{c} = \boldsymbol{c}_0 + \boldsymbol{c}_1$. Then $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m_0+m_1)$ and $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c})) = \mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}_0)) + \mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}_1))$ for all $i \in [n]$.*
- *Scalar multiplication: Let $t \in R$ have small norm and set $\boldsymbol{c}' = t \cdot \boldsymbol{c}_0$. Suppose the coefficients of $\mathtt{Err}(\boldsymbol{c}_0)$ are pairwise independent, then $\boldsymbol{c}' \in \mathsf{RLWE}_{\mathtt{sk}}(t \cdot m_0)$ and $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}')) \le \|t\|_\infty^2 \cdot \sum_{j\in[n]} \mathsf{Var}_j(\mathtt{Err}(\boldsymbol{c}_0))$ for all $i \in [n]$.*

Proof. For homomorphic addition, we have

$$\langle \boldsymbol{c}, \mathtt{sk} \rangle = \langle \boldsymbol{c}_0 + \boldsymbol{c}_1, \mathtt{sk} \rangle = \langle \boldsymbol{c}_0, \mathtt{sk} \rangle + \langle \boldsymbol{c}_1, \mathtt{sk} \rangle = \mathtt{Err}(\boldsymbol{c}_0) + \mathtt{Err}(\boldsymbol{c}_1) + (m_0 + m_1),$$

which implies $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m_0 + m_1)$ and $\mathtt{Err}(\boldsymbol{c}) = \mathtt{Err}(\boldsymbol{c}_0) + \mathtt{Err}(\boldsymbol{c}_1)$. By Lemma 2, we have $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c})) = \mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}_0)) + \mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}_1))$ for all $i \in [n]$. For scalar multiplication,

$$\langle \boldsymbol{c}', \mathtt{sk} \rangle = \langle t \cdot \boldsymbol{c}_0, \mathtt{sk} \rangle = t \cdot \langle \boldsymbol{c}_0, \mathtt{sk} \rangle = t \cdot \mathtt{Err}(\boldsymbol{c}_0) + t \cdot m_0,$$

which implies $\boldsymbol{c}' \in \mathsf{RLWE}_{\mathtt{sk}}(t \cdot m_0)$ and $\mathtt{Err}(\boldsymbol{c}') = t \cdot \mathtt{Err}(\boldsymbol{c}_0)$. By Lemma 3, we have $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}')) \le \|t\|_\infty^2 \cdot \sum_{j\in[n]} \mathsf{Var}_j(\mathtt{Err}(\boldsymbol{c}_0))$ for all $i \in [n]$. $\square$

Since RLWE-based schemes are defined over cyclotomics, we can also evaluate the automorphisms in the Galois group $\mathsf{Gal}(K/\mathbb{Q})$ on ciphertexts when provided with some auxiliary key material. To facilitate this, we first review the structure of these keys.

RLev Ciphertexts. The RLev ciphertexts [CLOT21] are also referred to as RLWE′ ciphertexts in the literature [MP21, KDE+24]. Intrinsically, a RLev ciphertext consists of a collection of RLWE encodings of the same message with different scaling factors. Formally, fix an integer base $\beta > 1$ and define $d = \lceil \log_\beta q \rceil$, then the set of all valid RLev encodings of a message $m \in R$ under the secret key $\mathtt{sk}$ is defined as

$$\mathsf{RLev}_{\mathtt{sk}}(m) := \left\{\mathsf{RLWE}_{\mathtt{sk}}(m \cdot \beta^i)\right\}_{i\in[d]} \subseteq R_q^{d\times 2},$$

where we use vertical concatenation of row vectors. The noise of RLev ciphertexts is defined as the concatenation of the noise from each RLWE encoding component:

Definition 4 (Noise of RLev Ciphertexts)**.** For a ciphertext $\mathbf{C} := (\boldsymbol{c}_0^\top | \boldsymbol{c}_1^\top | \cdots | \boldsymbol{c}_{d-1}^\top)^\top \in \mathsf{RLev}_{\mathtt{sk}}(m)$, the noise of $\mathbf{C}$ is defined as $\mathtt{Err}(\mathbf{C}) := (\mathtt{Err}(\boldsymbol{c}_0), \mathtt{Err}(\boldsymbol{c}_1), \ldots, \mathtt{Err}(\boldsymbol{c}_{d-1}))$.

Plaintext-Ciphertext Multiplication for RLev. We can multiply a RLev ciphertext $\mathbf{C} \in \mathsf{RLev}_{\mathtt{sk}}(m)$ with an arbitrary plaintext $t \in R_q$ as follows:

- *Deterministic gadget decomposition:* For $t \in R_q$, compute the signed base-β expansion of t as $t = \sum_{i\in[d]} t_i \beta^i$ where $t_i \in R$ and $\|t_i\|_\infty \le \beta/2$. In other words, define the gadget vector $\mathbf{g} := (1, \beta, \ldots, \beta^{d-1}) \in \mathbb{Z}^d$, then the gadget decomposition of t is defined as $\mathbf{g}^{-1}(t) := (t_0, \ldots, t_{d-1}) \in R^d$, which satisfies $\langle \mathbf{g}, \mathbf{g}^{-1}(t) \rangle = t \pmod q$.
- *Matrix-vector multiplication:* The multiplication of t with $\mathbf{C}$, denoted as $t \boxdot \mathbf{C}$, is defined as the matrix-vector multiplication $t \boxdot \mathbf{C} := \mathbf{g}^{-1}(t) \cdot \mathbf{C}$.

The following lemma demonstrates the correctness and noise analysis of this operation.

Lemma 5. *Let $\mathbf{C} \in \mathsf{RLev}_{\mathtt{sk}}(m)$, $t \in R_q$, and $\boldsymbol{c} \leftarrow t \boxdot \mathbf{C}$, then we have $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(t \cdot m)$. Suppose all coefficients of the components in $\mathtt{Err}(\mathbf{C})$ are pairwise independent and have common variance $\mathsf{Var}(\mathtt{Err}(\mathbf{C}))$, then $\mathsf{Var}(\mathtt{Err}(\boldsymbol{c})) \le nd\beta^2 \mathsf{Var}(\mathtt{Err}(\mathbf{C}))/4$.*

Proof. Denote $\mathbf{C} := (\boldsymbol{c}_0^\top | \boldsymbol{c}_1^\top | \cdots | \boldsymbol{c}_{d-1}^\top)^\top$ where $\boldsymbol{c}_i \in \mathsf{RLWE}_{\mathtt{sk}}(m \cdot \beta^i)$ for $i \in [d]$, then by definition, we have $\boldsymbol{c} := t \boxdot \mathbf{C} = \mathbf{g}^{-1}(t) \cdot \mathbf{C} = \sum_{i=0}^{d-1} t_i \cdot \boldsymbol{c}_i$ and $\langle \boldsymbol{c}_i, \mathtt{sk} \rangle = m \cdot \beta^i + \mathtt{Err}(\boldsymbol{c}_i)$ for $i \in [d]$. Therefore, we have

$$\langle \boldsymbol{c}, \mathtt{sk} \rangle = \sum_{i\in[d]} t_i \cdot \langle \boldsymbol{c}_i, \mathtt{sk} \rangle = \sum_{i\in[d]} t_i \cdot (m \cdot \beta^i + \mathtt{Err}(\boldsymbol{c}_i)) = t \cdot m + \sum_{i\in[d]} t_i \cdot \mathtt{Err}(\boldsymbol{c}_i),$$

which implies $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(t \cdot m)$ and $\mathtt{Err}(\boldsymbol{c}) = \sum_{i\in[d]} t_i \cdot \mathtt{Err}(\boldsymbol{c}_i)$. Applying Lemma 3 to each product $t_i \cdot \mathtt{Err}(\boldsymbol{c}_i)$ gives a coefficient-wise variance bound of $n\beta^2 \mathsf{Var}(\mathtt{Err}(\mathbf{C}))/4$, since $\|t_i\|_\infty^2 \le \beta^2/4$ and the n coefficients of $\mathtt{Err}(\boldsymbol{c}_i)$ each have variance $\mathsf{Var}(\mathtt{Err}(\mathbf{C}))$; summing these bounds over $i \in [d]$ yields the claimed bound. □

Automorphism Evaluation for RLWE. The evaluation of automorphisms involves two procedures, carried out during the key generation and evaluation phases, respectively:

- *Automorphism key generation:* On input the RLWE secret key $\mathtt{sk} = (-s, 1) \in R_q^2$, an automorphism $\sigma \in \mathsf{Gal}(K/\mathbb{Q})$, the automorphism key generation algorithm $\mathsf{AutoKeyGen}(\mathtt{sk}, \sigma)$ outputs the automorphism key $\mathtt{AK}_\sigma \in \mathsf{RLev}_{\mathtt{sk}}(-\sigma(s))$.
- *Automorphism evaluation:* On input a RLWE ciphertext $\boldsymbol{c} := (a, b)$ and an automorphism key $\mathtt{AK}_\sigma$, the automorphism evaluation algorithm $\mathsf{EvalAuto}(\boldsymbol{c}, \mathtt{AK}_\sigma)$ outputs $\boldsymbol{c}' \leftarrow (0, \sigma(b)) + \sigma(a) \boxdot \mathtt{AK}_\sigma$.

The following lemma demonstrates the correctness and noise analysis of this operation.

Lemma 6. *Let $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$, $\mathtt{AK}_\sigma \leftarrow \mathsf{AutoKeyGen}(\mathtt{sk}, \sigma)$, and $\boldsymbol{c}' \leftarrow \mathsf{EvalAuto}(\boldsymbol{c}, \mathtt{AK}_\sigma)$. Then $\boldsymbol{c}' \in \mathsf{RLWE}_{\mathtt{sk}}(\sigma(m))$, and $\mathtt{Err}(\boldsymbol{c}') = \sigma(\mathtt{Err}(\boldsymbol{c})) + e_{\mathtt{auto}}$ where $e_{\mathtt{auto}}$ is the additive noise introduced by $\mathsf{EvalAuto}$. Suppose that the coefficients of the components in $\mathtt{Err}(\mathtt{AK}_\sigma)$ are pairwise independent and have common variance $\mathsf{Var}(\mathtt{Err}(\mathtt{AK}_\sigma))$. For every $i \in [n]$, further assume that the i-th coefficient of $e_{\mathtt{auto}}$ is independent of the i-th coefficient of $\sigma(\mathtt{Err}(\boldsymbol{c}))$. Then, for each $i \in [n]$, we have $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}')) = \mathsf{Var}_i(\sigma(\mathtt{Err}(\boldsymbol{c}))) + V_{\mathtt{auto}}$, where $V_{\mathtt{auto}}$ is the variance of the additive noise $e_{\mathtt{auto}}$ bounded by $V_{\mathtt{auto}} \le nd\beta^2 \mathsf{Var}(\mathtt{Err}(\mathtt{AK}_\sigma))/4$.*

Proof. Denote $\boldsymbol{c} := (a, b) \in R_q^2$, then we have $\mathtt{Err}(\boldsymbol{c}) = \langle \boldsymbol{c}, \mathtt{sk} \rangle - m = b - s \cdot a - m$. Applying the automorphism σ, we have $\sigma(\mathtt{Err}(\boldsymbol{c})) = \sigma(b) - \sigma(s) \cdot \sigma(a) - \sigma(m)$. By definition, we

have $\mathtt{AK}_\sigma \in \mathsf{RLev}_{\mathtt{sk}}(-\sigma(s))$, which implies $\boldsymbol{c}_{\mathtt{mul}} := \sigma(a) \boxdot \mathtt{AK}_\sigma \in \mathsf{RLWE}_{\mathtt{sk}}(-\sigma(s)\cdot\sigma(a))$ by Lemma 5. Then, $\langle \boldsymbol{c}_{\mathtt{mul}}, \mathtt{sk}\rangle = -\sigma(s)\cdot\sigma(a) + \mathtt{Err}(\boldsymbol{c}_{\mathtt{mul}})$, and thus we have

$$\begin{aligned}\langle \boldsymbol{c}', \mathtt{sk}\rangle &= \langle (0,\sigma(b)) + \boldsymbol{c}_{\mathtt{mul}}, \mathtt{sk}\rangle \\ &= \langle (0,\sigma(b)), \mathtt{sk}\rangle + \langle \boldsymbol{c}_{\mathtt{mul}}, \mathtt{sk}\rangle \\ &= \sigma(b) - \sigma(s)\cdot\sigma(a) + \mathtt{Err}(\boldsymbol{c}_{\mathtt{mul}}) \\ &= \sigma(m) + \sigma(\mathtt{Err}(\boldsymbol{c})) + \mathtt{Err}(\boldsymbol{c}_{\mathtt{mul}}),\end{aligned}$$

which implies $\boldsymbol{c}' \in \mathsf{RLWE}_{\mathtt{sk}}(\sigma(m))$ and $\mathtt{Err}(\boldsymbol{c}') = \sigma(\mathtt{Err}(\boldsymbol{c})) + \mathtt{Err}(\boldsymbol{c}_{\mathtt{mul}})$. Therefore, we have $e_{\mathtt{auto}} = \mathtt{Err}(\boldsymbol{c}_{\mathtt{mul}})$ and $\mathsf{Var}(e_{\mathtt{auto}}) \le nd\beta^2\mathsf{Var}(\mathtt{Err}(\mathtt{AK}_\sigma))/4$ by Lemma 5, and the final result follows from the independence assumption and Lemma 2. □

Remark 1. To simplify notation, we write $\mathsf{EvalAuto}(\boldsymbol{c}, i)$ for $\mathsf{EvalAuto}(\boldsymbol{c}, \mathtt{AK}_\sigma)$, where σ is the automorphism defined by $\sigma : \zeta_{2n} \mapsto \zeta_{2n}^i$ for some $i \in \mathbb{Z}_{2n}^*$.

Modulus Switching. The modulus switching technique enables the conversion of a RLWE ciphertext from a modulus q to another modulus q', a method initially introduced in leveled FHE for noise management [BV11a, BGV12]. The functionality and noise analysis of modulus switching are summarized in the following lemma.

Lemma 7. *Let $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m) \subseteq R_q^2$ and q' be another modulus, then the modulus switching algorithm $\mathsf{ModSwitch}_{q\to q'}(\boldsymbol{c})$ outputs $\boldsymbol{c}' = \lfloor (q'/q)\boldsymbol{c} \rceil \bmod q' \in \mathsf{RLWE}_{\mathtt{sk}}((q'/q)m) \subseteq R_{q'}^2$. The noise of $\boldsymbol{c}'$ satisfies $\mathtt{Err}(\boldsymbol{c}') = (q'/q)\cdot\mathtt{Err}(\boldsymbol{c}) + e_{\mathtt{ms}}$ where $e_{\mathtt{ms}} = \langle \boldsymbol{c}' - (q'/q)\boldsymbol{c}, \mathtt{sk}\rangle$ is the additive noise introduced by the modulus switching operation. Assume heuristically that all the coefficients across the two entries of $\boldsymbol{c}' - (q'/q)\boldsymbol{c}$ are pairwise independent and independent of $\mathtt{Err}(\boldsymbol{c})$, and that they are uniformly distributed over the set of rounding errors induced by nearest-integer rounding for the ratio q'/q. Then $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}')) = (q'/q)^2\cdot\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c})) + V_{\mathtt{ms}}$ for all $i \in [n]$ where $V_{\mathtt{ms}}$ is the variance of $e_{\mathtt{ms}}$ satisfying $V_{\mathtt{ms}} < \left(1 + \|s\|_2^2\right)/12$.*

Proof. By definition, we have $\langle \boldsymbol{c}, \mathtt{sk}\rangle = m + \mathtt{Err}(\boldsymbol{c})$. Write $\boldsymbol{c}' - (q'/q)\boldsymbol{c} = (e_0, e_1) \in K^2$ where e_0, e_1 are the rounding errors. Under the assumed model, the coefficients of e_0 and e_1 are uniformly distributed over the same equally spaced grid of rounding errors in $(-1/2, 1/2]$, and hence $\mathsf{Var}_i(e_0) = \mathsf{Var}_i(e_1) < 1/12$ for all $i \in [n]$. Thus, we have

$$\langle \boldsymbol{c}', \mathtt{sk}\rangle = \left\langle \frac{q'}{q}\boldsymbol{c} + (e_0, e_1),\ \mathtt{sk} \right\rangle = \frac{q'}{q}m + \frac{q'}{q}\mathtt{Err}(\boldsymbol{c}) + e_1 - s\cdot e_0 \mod q',$$

which implies $\boldsymbol{c}' \in \mathsf{RLWE}_{\mathtt{sk}}((q'/q)m)$ and $\mathtt{Err}(\boldsymbol{c}') = (q'/q)\cdot\mathtt{Err}(\boldsymbol{c}) + e_{\mathtt{ms}}$ where $e_{\mathtt{ms}} = e_1 - s\cdot e_0$. Thus, we have $\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c}')) = (q'/q)^2\cdot\mathsf{Var}_i(\mathtt{Err}(\boldsymbol{c})) + \mathsf{Var}_i(e_{\mathtt{ms}})$ for all $i \in [n]$. By our independence assumption of e_0 and e_1 and Lemma 3, we have

$$\mathsf{Var}_i(e_{\mathtt{ms}}) = \mathsf{Var}_i(e_1) + \mathsf{Var}_i(s\cdot e_0) < \frac{1}{12} + \|s\|_2^2\cdot\frac{1}{12} = \frac{1+\|s\|_2^2}{12}.$$

The final result follows from the independence assumption and Lemma 2. □

Modulus Raising. The modulus raising technique reinterprets a RLWE ciphertext defined over a modulus q as a ciphertext over a larger modulus $Q > q$. This operation is widely used as the first step in CKKS bootstrapping [CHK+18]. Concretely, given a RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_{\mathtt{sk}}(m) \subseteq R_q^2$, the modulus raising algorithm $\mathsf{ModRaise}_{q\to Q}(\boldsymbol{c})$ outputs $\boldsymbol{c}' = \boldsymbol{c} \bmod Q$. Since $Q > q$, this procedure involves no actual operation on the coefficients of $\boldsymbol{c}$. However, when working modulo Q, additional higher-order terms naturally arise in the phase but carry no semantic information. Specifically, since the phase of $\boldsymbol{c}$ satisfies $\langle \boldsymbol{c}, \mathtt{sk}\rangle = m + \mathtt{Err}(\boldsymbol{c}) \bmod q$, it follows that $\langle \boldsymbol{c}, \mathtt{sk}\rangle = qI + m + \mathtt{Err}(\boldsymbol{c}) \in R$ for some $I \in R$. Therefore, after modulus raising, the phase of $\boldsymbol{c}'$ satisfies $\langle \boldsymbol{c}', \mathtt{sk}\rangle = qI + m + \mathtt{Err}(\boldsymbol{c}) \bmod Q$, where the term qI is newly introduced due to the modulus raising.

Algorithm 1 The CDKS Homomorphic Trace Evaluation $\mathsf{CDKS.EvalTr}_{K/\mathbb{Q}}$

Input: A ciphertext $\mathsf{ct} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$
Input: Automorphism keys for $\sigma_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+n/2^{k-1}}$ where $k = 1, \ldots, \log n$
Output: A ciphertext $\mathsf{ct}' \in \mathsf{RLWE}_{\mathtt{sk}}(\mathsf{Tr}_{K/\mathbb{Q}}(m))$
1: $\mathsf{ct}' \leftarrow \mathsf{ct}$
2: **for** $k = 1$ to $\log n$ **do**
3: $\quad \mathsf{ct}' \leftarrow \mathsf{ct}' + \mathsf{EvalAuto}(\mathsf{ct}', 2^{\log n - k + 1} + 1)$
4: **end for**
5: **return** ct'

3 Homomorphic Trace Evaluation

In this section, we revisit the existing efficient methods for homomorphic evaluation of the trace mapping over power-of-two cyclotomics, including the widely used CDKS method [CDKS21] and the recently optimized LY method [LY26] based on [WHS+25].

3.1 The CDKS Method

The CDKS method [CDKS21] takes advantage of the tower structure of power-of-two cyclotomic extensions and the transitivity of trace mapping. Let $K_r = \mathbb{Q}(\zeta_{2^r})$ be the 2^r-th cyclotomic field. Then the evaluation of $\mathsf{Tr}_{K_r/\mathbb{Q}}$ follows the chain:

$$K_r \xrightarrow{\mathsf{Tr}_{K_r/K_{r-1}}} K_{r-1} \xrightarrow{\mathsf{Tr}_{K_{r-1}/K_{r-2}}} K_{r-2} \longrightarrow \cdots \longrightarrow K_2 \xrightarrow{\mathsf{Tr}_{K_2/K_1}} K_1 = \mathbb{Q}.$$

Each intermediate trace mapping $\mathsf{Tr}_{K_i/K_{i-1}}$ requires evaluating the automorphisms in the Galois group $\mathsf{Gal}(K_i/K_{i-1})$, which is of order 2. Specifically, $\mathsf{Gal}(K_i/K_{i-1}) = \{\mathsf{id}, \rho_i\}$, where id is the identity map and ρ_i is defined by $\rho_i : \zeta_{2^i} \mapsto -\zeta_{2^i}$. Therefore, $\mathsf{Tr}_{K_r/\mathbb{Q}}$ only requires evaluating the automorphisms ρ_i for all $i = 2, \ldots, r$, which is $r-1$ in total. The CDKS method for homomorphic evaluation of $\mathsf{Tr}_{K/\mathbb{Q}}$ where $K = \mathbb{Q}(\zeta_{2n})$ is given in Algorithm 1, and its original noise analysis is summarized in Lemma 8.

Lemma 8 (Adapted from [CDKS21])**.** *Adopt the notations from Algorithm 1. The variance of the output noise is bounded by* $\mathsf{Var}(\mathtt{Err}(\mathsf{ct}')) \le \mathsf{Var}(\mathsf{Tr}_{K/\mathbb{Q}}(\mathtt{Err}(\mathsf{ct}))) + \frac{1}{3}(n^2-1) \cdot V_{\mathtt{auto}}$ *where* $V_{\mathtt{auto}}$ *is the additive noise variance introduced by* $\mathsf{EvalAuto}$ *as given in Lemma 6.*

Issues in the CDKS Method. Although the CDKS method is computationally efficient, it has the following obstacles in practice, as reported in recent works [WHS+25, LY26]:

- *Phase Amplification:* Suppose the input ciphertext ct has a phase $\varphi = m + e$, where $m \in R$ is the plaintext and $e \in R$ is the noise term. Then the phase of the output ciphertext ct' contains $\mathsf{Tr}_{K/\mathbb{Q}}(\varphi) = n \cdot (m_0 + e_0)$. Namely, the constant terms of m and e remain but are multiplied by n, which is undesirable in most trace-based applications. The work [CDKS21] suggests first scaling ct by $n^{-1} \in \mathbb{Z}_q^*$ before applying the trace evaluation, but this method is not compatible with FFT-based implementations[1] (over $\mathbb{C}$). For the word-size moduli used in FHEW/TFHE-type bootstrapping, FFT has been reported to be more efficient in practice than NTT [ABBB+22, Zam22, WHS+25, WLX+25].
- *Noise Accumulation:* For fixed gadget parameters and noise variance of automorphism keys, Lemma 6 gives $V_{\mathtt{auto}} \in O(n)$. Hence, Lemma 8 implies that trace evaluation adds $\frac{1}{3}(n^2-1)V_{\mathtt{auto}} \in O(n^3)$ to the noise variance. This accumulation becomes significant when n is large or when the noise in the input ciphertext is relatively small.

[1] For NTT-based implementations, q is a prime or a product of distinct odd primes, and hence $n \in \mathbb{Z}_q^*$. In contrast, for FFT-based implementations, q is typically a power of two, e.g., 2^{32} or 2^{64}, so $n \notin \mathbb{Z}_q^*$.

Algorithm 2 The LY Homomorphic (Scaled) Trace Evaluation $\mathsf{LY.EvalScaledTr}_{K/\mathbb{Q}}$

Input: A ciphertext $\mathsf{ct} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$ where $m = \sum_{i\in[n]} m_i \cdot \zeta_{2n}^i$

Input: Automorphism keys for $\hat{\sigma}_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+2^k}$ where $k = 1, \ldots, \log n$

Output: A ciphertext $\mathsf{ct}' \in \mathsf{RLWE}_{\mathtt{sk}}(m_0)$

1: $\mathsf{ct}' \leftarrow \mathsf{ct}$
2: **for** $k = 1$ to $\log n$ **do**
3: $\quad \widetilde{\mathsf{ct}} \leftarrow \mathsf{ModRaise}_{q/2\to q} \circ \mathsf{ModSwitch}_{q\to q/2}(\mathsf{ct}')$
4: $\quad \mathsf{ct}' \leftarrow \widetilde{\mathsf{ct}} + \mathsf{EvalAuto}(\widetilde{\mathsf{ct}}, 2^k + 1)$
5: **end for**
6: **return** ct'

3.2 The WHS$^+$-LY Method

The WHS$^+$ Method. To address the phase amplification issue in FFT-based implementations, the work [WHS+25] proposes using modulus switching and modulus raising as preprocessing steps before applying the CDKS method. Specifically, given a ciphertext $\mathsf{ct} \in \mathsf{RLWE}_{\mathtt{sk}}(m) \subseteq R_q^2$ where q is a power of two, the WHS$^+$ method proceeds as follows:

$$\mathsf{ct} \xrightarrow{\mathsf{ModSwitch}_{q\to q/n}} \mathsf{ct}_{\mathtt{ms}} \xrightarrow{\mathsf{ModRaise}_{q/n\to q}} \mathsf{ct}_{\mathtt{mr}} \xrightarrow{\mathsf{CDKS.EvalTr}_{K/\mathbb{Q}}} \mathsf{ct}_{\mathtt{tr}}.$$

To see why this method works, let us analyze the phase of the intermediate ciphertexts:

- The phase of the original ciphertext ct is $\varphi = m + e \pmod q$.
- The phase of $\mathsf{ct}_{\mathtt{ms}}$ is $\varphi_{\mathtt{ms}} = \frac{1}{n} \cdot \varphi + e_{\mathtt{ms}} = \frac{1}{n} \cdot (m + e) + e_{\mathtt{ms}} \pmod{q/n}$ by Lemma 7.
- The phase of $\mathsf{ct}_{\mathtt{mr}}$ is $\varphi_{\mathtt{mr}} = \frac{q}{n} \cdot I + \varphi_{\mathtt{ms}} = \frac{q}{n} \cdot I + \frac{1}{n} \cdot (m+e) + e_{\mathtt{ms}} \pmod q$ for some $I \in R$.
- The phase of $\mathsf{ct}_{\mathtt{tr}}$ is $\varphi_{\mathtt{tr}} = \mathsf{Tr}_{K/\mathbb{Q}}(\varphi_{\mathtt{mr}}) + e_{\mathtt{tr}} \pmod q$ where $e_{\mathtt{tr}}$ is the noise introduced by the CDKS trace evaluation as given in Lemma 8. Expanding $\varphi_{\mathtt{tr}}$ further yields:

$$\begin{aligned}\varphi_{\mathtt{tr}} = \mathsf{Tr}_{K/\mathbb{Q}}(\varphi_{\mathtt{mr}}) + e_{\mathtt{tr}} &= n \cdot \frac{q}{n} \cdot I_0 + n \cdot \frac{1}{n} \cdot (m_0 + e_0) + n \cdot e_{\mathtt{ms}}^{(0)} + e_{\mathtt{tr}} \pmod q \\ &= m_0 + e_0 + n \cdot e_{\mathtt{ms}}^{(0)} + e_{\mathtt{tr}} \pmod q\end{aligned}$$

  where I_0, m_0, e_0, and $e_{\mathtt{ms}}^{(0)}$ are the constant terms of I, m, e, and $e_{\mathtt{ms}}$ respectively. Therefore, the constant term of the original phase remains without amplification.

The Optimized LY Method. To further reduce the noise accumulation remaining in the WHS$^+$ method, the work [LY26] proposed integrating modulus switching and modulus raising into each iteration of the CDKS trace evaluation. Instead of performing modulus switching by a factor of $1/n$ all at once as in the WHS$^+$ method, the LY method performs modulus switching by a factor of $1/2$ in each iteration. However, to ensure correctness, the LY method needs to evaluate the automorphisms in the reverse order relative to the CDKS method during the iterations. Without phase amplification, the WHS$^+$-LY method is not precisely a trace evaluation but a scaled version with a scaling factor of $1/n$. The formal description of the LY method for homomorphic trace evaluation is provided in Algorithm 2, and its original noise analysis is summarized in Lemma 9.

Lemma 9 (Adapted from [LY26])**.** *Adopt the notations from Algorithm 2. The variance of the output noise is bounded by* $\mathsf{Var}(\mathtt{Err}(\mathsf{ct}')) \le \mathsf{Var}(\mathtt{Err}(\mathsf{ct})) + 4\log n \cdot V_{\mathtt{ms}} + \log n \cdot V_{\mathtt{auto}}$ *where* $V_{\mathtt{ms}}$ *and* $V_{\mathtt{auto}}$ *are the variances of the additive noise introduced by* ModSwitch *and* EvalAuto *as given in Lemma 7 and Lemma 6, respectively.*

4 Algebraic Analysis of Trace Evaluation

In this section, we present an algebraic analysis of homomorphic trace evaluation that clarifies its underlying mechanism and yields refined noise estimates. We begin by reanalyzing the LY method [LY26] in the same setting considered in the original work, where the ciphertext modulus q is a power of two, as in FFT-based implementations.

4.1 Reanalysis of the LY Method

First, we provide several insights into the correctness of the LY method from an algebraic perspective. To illustrate this, we need to establish the following two facts: (1) the overall sequence of automorphisms applied during the iterations ultimately yields a correct (scaled) trace evaluation, and (2) the additional terms introduced by modulus raising in the iterations always vanish after the final iteration of trace evaluation.

Ordering of Automorphisms. When exploiting the tower structure for trace evaluation as in the CDKS method (Algorithm 1), the automorphism applied at each iteration is $\sigma_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+n/2^{k-1}}$ for $k = 1, \ldots, \log n$. After each iteration, the plaintext is transformed by the $\mathbb{Q}$-linear operator τ_k, defined by $\tau_k(a) = a + \sigma_k(a)$ for $a \in K$. In fact, the operators τ_k commute with each other, as shown by the following lemma in a more general setting.

Lemma 10. *Let $\sigma, \sigma' \in \mathsf{Gal}(K/\mathbb{Q})$ be two automorphisms, and define the $\mathbb{Q}$-linear operators $\tau, \tau' : K \to K$ by $\tau(a) = a + \sigma(a)$ and $\tau'(a) = a + \sigma'(a)$ for any $a \in K$. Then, τ and τ' commute, i.e., $\tau \circ \tau' = \tau' \circ \tau$.*

Proof. For any $a \in K$, we have

$$\tau \circ \tau'(a) = \tau(a + \sigma'(a)) = a + \sigma'(a) + \sigma(a + \sigma'(a)) = a + \sigma'(a) + \sigma(a) + \sigma \circ \sigma'(a),$$
$$\tau' \circ \tau(a) = \tau'(a + \sigma(a)) = a + \sigma(a) + \sigma'(a + \sigma(a)) = a + \sigma(a) + \sigma'(a) + \sigma' \circ \sigma(a).$$

Since $\mathsf{Gal}(K/\mathbb{Q})$ is abelian, we have $\sigma \circ \sigma' = \sigma' \circ \sigma$. Thus, we conclude that $\tau \circ \tau'(a) = \tau' \circ \tau(a)$ for any $a \in K$, which implies the desired commutativity of the operators τ and τ'. □

Corollary 1. *Let $\{\sigma_i\}_{i \in I} \subseteq \mathsf{Gal}(K/\mathbb{Q})$ be a finite set of automorphisms, and for each $i \in I$ let $\tau_i : K \to K$ be the $\mathbb{Q}$-linear operator defined by $\tau_i(a) = a + \sigma_i(a)$. Then the composition of the operators $\{\tau_i\}_{i \in I}$ is independent of order.*

The above corollary shows that the ordering of the linear operators τ_i does not affect the resulting trace value. Therefore, in the LY method (Algorithm 2) where the operators are applied in reverse order relative to the CDKS method (i.e., applying $\hat{\tau}_k := \tau_{\log n - k + 1}$ in the k-th iteration), the final result still contains a correct (scaled) trace evaluation.

Vanishing of Additional Terms. Although the order of applying the linear operators is immaterial to the final trace value, the reverse order used in the LY method is crucial for ensuring that the additional terms introduced by modulus raising vanish after the final iteration. Specifically, in the k-th iteration of Algorithm 2, the modulus raising step $\mathsf{ModRaise}_{q/2 \to q}$ will introduce an additional term added to the phase, which is of the form $\frac{q}{2} I_k$ for some $I_k \in R$ since the input ciphertext modulus becomes $\frac{q}{2}$ after the preceding modulus switching. These terms are also transformed by $\{\hat{\tau}_i\}_{i=k,\ldots,\log n}$ in the current and subsequent iterations. The key is therefore to show that their final contribution vanishes. Interestingly, we can use the commutativity of the linear operators shown in Lemma 10 to prove that the composition of the subsequent operators is in fact a relative trace mapping from K to one of its cyclotomic subfields.

Lemma 11. *Let $\hat{\sigma}_k \in \mathsf{Gal}(K/\mathbb{Q})$ be the automorphism defined by $\hat{\sigma}_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+2^k}$ for $k = 1, \ldots, \log n$, and let $\hat{\tau}_k : K \to K$ be the $\mathbb{Q}$-linear operator defined by $\hat{\tau}_k(a) = a + \hat{\sigma}_k(a)$ for any $a \in K$. Then, for any $1 \le k \le \log n$, we have*

$$\hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_{k+1} \circ \hat{\tau}_k = \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}.$$

Proof. By construction, the operator $\hat{\tau}_k$ coincides with the trace map $\mathsf{Tr}_{\mathbb{Q}(\zeta_{2^{k+1}})/\mathbb{Q}(\zeta_{2^k})}$ when restricted to $\mathbb{Q}(\zeta_{2^{k+1}})$. Since the automorphisms $\{\hat{\sigma}_k\}_{k=1,\ldots,\log n}$ correspond, up to relabeling, to the same set of automorphisms as $\{\sigma_k\}_{k=1,\ldots,\log n}$, Lemma 10 and Corollary 1 imply that the operators $\hat{\tau}_k$ commute with each other. Therefore, we may reorder them as

$$\begin{aligned}
\hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_{k+1} \circ \hat{\tau}_k &= \hat{\tau}_k \circ \hat{\tau}_{k+1} \circ \cdots \circ \hat{\tau}_{\log n} \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{2^{k+1}})/\mathbb{Q}(\zeta_{2^k})} \circ \mathsf{Tr}_{\mathbb{Q}(\zeta_{2^{k+2}})/\mathbb{Q}(\zeta_{2^{k+1}})} \circ \cdots \circ \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^{\log n}})} \\
&= \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})},
\end{aligned}$$

where the last equality follows from the transitivity of the trace. □

Using the lemma above, we can now demonstrate how the additional terms introduced by modulus raising vanish after the final iteration.

Proposition 1. *Adopt the notations from Algorithm 2. For each $k = 1, \ldots, \log n$, let $\frac{q}{2} I_k \in R$ be the additional term introduced in the k-th iteration due to modulus raising. Then, all these additional terms vanish after the final iteration of Algorithm 2.*

Proof. In the k-th iteration of Algorithm 2, the newly introduced additional term $\frac{q}{2} I_k$ is transformed by the operator $\hat{\tau}_k$. In each subsequent iteration i for $i = k+1, \ldots, \log n$, the term is first scaled by $\frac{1}{2}$ due to the modulus switching, and then transformed by the operator $\hat{\tau}_i$. By the $\mathbb{Q}$-linearity of the operators, all the factors of $\frac{1}{2}$ and q can be combined together as $q \cdot 2^{-(\log n - k + 1)} = q \cdot 2^{k-1}/n$. Therefore, after the final iteration, the contribution of the additional term $\frac{q}{2} I_k$ to the final result is given by

$$\begin{aligned}
\frac{q \cdot 2^{k-1}}{n} \cdot \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_{k+1} \circ \hat{\tau}_k(I_k) &= \frac{q \cdot 2^{k-1}}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}(I_k) && \text{(by Lemma 11)} \\
&= \frac{q \cdot 2^{k-1}}{n} \cdot \frac{n}{2^{k-1}} \cdot I_k^{(0)} && \text{(by Lemma 1)} \\
&\equiv 0 \pmod{q},
\end{aligned}$$

where $I_k^{(0)} \in \mathbb{Z}[\zeta_{2^k}]$ denotes the constant term of I_k when expressed in the power basis of K over the subfield $\mathbb{Q}(\zeta_{2^k})$. This completes the proof. □

The preceding discussion establishes the correctness of the LY method from an algebraic perspective. In the next subsection, we will further leverage these insights to derive improved noise estimates for the LY method.

4.2 Improved Noise Estimation

Our improved noise estimation stems from the observation that noise introduced in each iteration of trace evaluation is further transformed by the subsequent linear operators characterized in the preceding subsection. Prior recurrence-based noise analyses [CDKS21, WHS+25, LY26] generally reduce the cumulative action of these transformations to scalar amplification, thereby obscuring their exact algebraic effects.

As established above, the relevant compositions of these operators are trace mappings from K to its subfields and therefore zero out certain coefficients. Therefore, we conduct a coefficient-wise analysis to track which coefficients survive these trace mappings and contribute to the final noise. Moreover, as the accumulated additive noise arises from modulus switching and automorphism evaluation, linearity enables us to analyze these two contributions separately.

Noise from Modulus Switching. In each iteration of Algorithm 2, modulus switching from q to $q/2$ introduces an additive noise term whose coefficients can be viewed as samples from a distribution with variance $V_{\mathtt{ms}}$ (see Lemma 7). The following lemma characterizes how these noise terms are transformed by the subsequent linear operators.

Lemma 12. *Assume that in Algorithm 2, the noise introduced by each* ModSwitch *operation has coefficient-wise variance $V_{\mathtt{ms}}$, and that the noise terms from different iterations are independent of each other. Denote by $E_{\mathtt{ms}}$ the total noise accumulated from modulus switching after all $\log n$ iterations. Then, for each coefficient index $i \in [n]$, we have $\mathsf{Var}_i(E_{\mathtt{ms}}) = 4\nu_2(i) \cdot V_{\mathtt{ms}}$, where $\nu_2(i)$ denotes the 2-adic valuation of i for $i > 0$ (i.e., the largest integer r such that 2^r divides i) and we set $\nu_2(0) := \log n$.*

Proof. Let E_k denote the noise term introduced by modulus switching in the k-th iteration for $k = 1, \ldots, \log n$. By the assumption, we have $\mathsf{Var}_i(E_k) = V_{\mathtt{ms}}$ or simply $\mathsf{Var}(E_k) = V_{\mathtt{ms}}$.

In the k-th iteration and subsequent iterations, the noise term E_k is transformed by the operators $\hat{\tau}_k, \ldots, \hat{\tau}_{\log n}$. By Lemma 11, the overall transformation is given by the trace mapping $\mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}$. Additionally, due to modulus switching in each subsequent iteration, the noise term is scaled by $1/2$ in each iteration. Since there are $(\log n - k)$ such subsequent iterations, the total scaling factor is $2^{-(\log n - k)} = 2^k/n$. Combining these effects, after all iterations, the contribution of E_k to the final noise term $E_{\mathtt{ms}}$ is given by $2^k/n \cdot \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}(E_k)$. Therefore, we have

$$E_{\mathtt{ms}} = \sum_{k=1}^{\log n} \frac{2^k}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}(E_k) = \sum_{k=1}^{\log n} \frac{2^k}{n} \cdot \frac{n}{2^{k-1}} \cdot E_k^{(0)} = \sum_{k=1}^{\log n} 2 \cdot E_k^{(0)},$$

where $E_k^{(0)} \in \mathbb{Q}(\zeta_{2^k})$ denotes the constant term of E_k when expressed in the power basis of K over the subfield $\mathbb{Q}(\zeta_{2^k})$.

Since $\mathbb{Q}(\zeta_{2^k}) = \mathbb{Q}\big(\zeta_{2n}^{2^{\log n - k + 1}}\big)$, when any element in $\mathbb{Q}(\zeta_{2^k})$ is represented in the power basis $\{1, \zeta_{2n}, \ldots, \zeta_{2n}^{n-1}\}$ of K, its coefficients can be nonzero only at indices $i = r \cdot 2^{\log n - k + 1}$ for $0 \le r < 2^{k-1}$. Equivalently, for $i \in [n]$, the contribution of $E_k^{(0)}$ appears in the i-th coefficient of $E_{\mathtt{ms}}$ if and only if $2^{\log n - k + 1}$ divides i.

For $i = 0$, all the contributions $2E_k^{(0)}$ appear in the constant coefficient of $E_{\mathtt{ms}}$ with respect to the power basis of K over $\mathbb{Q}$, so

$$\mathsf{Var}_0(E_{\mathtt{ms}}) = \mathsf{Var}_0\left(\sum_{k=1}^{\log n} 2 \cdot E_k^{(0)}\right) = \sum_{k=1}^{\log n} 4 \cdot \mathsf{Var}\left(E_k\right) = \sum_{k=1}^{\log n} 4 \cdot V_{\mathtt{ms}} = 4 \log n \cdot V_{\mathtt{ms}}.$$

For $0 < i < n$, the condition that $2^{\log n - k + 1}$ divides i is equivalent to $\log n - k + 1 \le \nu_2(i)$, or $k \ge \log n - \nu_2(i) + 1$. As k ranges from 1 to $\log n$, the contributions appear for $k = \log n - \nu_2(i) + 1, \ldots, \log n$. Thus,

$$\mathsf{Var}_i(E_{\mathtt{ms}}) = \sum_{k=\log n - \nu_2(i) + 1}^{\log n} \mathsf{Var}\left(2 \cdot E_k\right) = \sum_{k=\log n - \nu_2(i) + 1}^{\log n} 4 \cdot V_{\mathtt{ms}} = 4\nu_2(i) \cdot V_{\mathtt{ms}}.$$

Combining with the case of $i = 0$ and the setting $\nu_2(0) := \log n$, we can conclude that for each coefficient index $i \in [n]$, we have $\mathsf{Var}_i(E_{\mathtt{ms}}) = 4\nu_2(i) \cdot V_{\mathtt{ms}}$. □

Noise from Automorphism Evaluation. In each iteration of Algorithm 2, the automorphism evaluation EvalAuto introduces an additive noise term whose coefficients can be modeled as samples from a distribution with variance $V_{\mathtt{auto}}$ as given in Lemma 6. Similar to the previous case, we analyze how these noise terms are transformed by the subsequent linear operators in the following lemma.

Lemma 13. *Suppose in all iterations of Algorithm 2, the noise introduced by* EvalAuto *has the same coefficient-wise variance $V_{\mathtt{auto}}$, and that the noise terms from different iterations are independent of each other. Denote by $E_{\mathtt{auto}}$ the total noise accumulated from automorphism evaluation after all $\log n$ iterations. Then, the constant coefficient satisfies $\mathsf{Var}_0(E_{\mathtt{auto}}) = \log n \cdot V_{\mathtt{auto}}$, whereas every nonconstant coefficient $i \in [n] \setminus \{0\}$ satisfies $\mathsf{Var}_i(E_{\mathtt{auto}}) = (\nu_2(i)+1) \cdot V_{\mathtt{auto}}$, where $\nu_2(i)$ denotes the 2-adic valuation of i.*

Proof. Let E_k denote the noise term introduced by automorphism evaluation in the k-th iteration for $k = 1, \ldots, \log n$. By the assumption, the coefficients of E_k are samples from a distribution with variance $V_{\mathtt{auto}}$, which implies $\mathsf{Var}_i(E_k) = V_{\mathtt{auto}}$ or simply $\mathsf{Var}(E_k) = V_{\mathtt{auto}}$.

For $k < \log n$, after the k-th iteration, the noise term E_k is transformed by the subsequent operators $\hat{\tau}_{k+1}, \ldots, \hat{\tau}_{\log n}$ in the following iterations. By Lemma 11, the overall transformation is given by the trace mapping $\mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^{k+1}})}$. For $k = \log n$, we interpret this trace map as the identity map on $K = \mathbb{Q}(\zeta_{2n})$. Combining with the scaling effect due to modulus switching in each subsequent iteration, after all iterations, we have

$$E_{\mathtt{auto}} = \sum_{k=1}^{\log n} \frac{2^k}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^{k+1}})}(E_k) = \sum_{k=1}^{\log n} \frac{2^k}{n} \cdot \frac{n}{2^k} \cdot E_k^{(0)} = \sum_{k=1}^{\log n} E_k^{(0)},$$

where $E_k^{(0)} \in \mathbb{Q}(\zeta_{2^{k+1}})$ denotes the constant term of E_k when expressed in the power basis of K over the subfield $\mathbb{Q}(\zeta_{2^{k+1}})$; in particular, $E_{\log n}^{(0)} = E_{\log n}$.

Since $\mathbb{Q}(\zeta_{2^{k+1}}) = \mathbb{Q}\big(\zeta_{2n}^{2^{\log n - k}}\big)$, when any element in $\mathbb{Q}(\zeta_{2^{k+1}})$ is represented in the power basis $\{1, \zeta_{2n}, \ldots, \zeta_{2n}^{n-1}\}$ of K, its coefficients can be nonzero only at indices $i = r \cdot 2^{\log n - k}$ for $0 \le r < 2^k$. Equivalently, for $i \in [n]$, the contribution of $E_k^{(0)}$ appears in the i-th coefficient of $E_{\mathtt{auto}}$ if and only if $2^{\log n - k}$ divides i.

For $i = 0$, all the contributions $E_k^{(0)}$ appear in the constant coefficient of $E_{\mathtt{auto}}$ with respect to the power basis of K over $\mathbb{Q}$. Hence, by independence across iterations,

$$\mathsf{Var}_0(E_{\mathtt{auto}}) = \mathsf{Var}_0\left(\sum_{k=1}^{\log n} E_k^{(0)}\right) = \sum_{k=1}^{\log n} \mathsf{Var}\left(E_k\right) = \sum_{k=1}^{\log n} V_{\mathtt{auto}} = \log n \cdot V_{\mathtt{auto}}.$$

For $0 < i < n$, the condition that $2^{\log n - k}$ divides i is equivalent to $\log n - k \le \nu_2(i)$ or $k \ge \log n - \nu_2(i)$. Considering that k ranges from 1 to $\log n$, we have

$$\mathsf{Var}_i(E_{\mathtt{auto}}) = \mathsf{Var}_i\left(\sum_{k=\log n - \nu_2(i)}^{\log n} E_k^{(0)}\right) = \sum_{k=\log n - \nu_2(i)}^{\log n} \mathsf{Var}(E_k) = (\nu_2(i)+1) \cdot V_{\mathtt{auto}}.$$

Combining with the case $i = 0$, we have the desired result. □

Overall Noise Estimate. By combining the results from Lemmas 12 and 13, we can derive the overall noise estimate for the LY method as follows. Note that the original noise term of the input ciphertext is also transformed by the scaled trace mapping.

Theorem 1. *Adopt the notations from Algorithm 2. Suppose that the noise introduced by each* ModSwitch *has common coefficient variance $V_{\mathtt{ms}}$ and that the noise introduced by each* EvalAuto *has common coefficient variance $V_{\mathtt{auto}}$. Assume heuristically that the constant term of $\mathtt{Err}(\mathsf{ct})$ and the noise contributions arising from all distinct* ModSwitch *and* EvalAuto *calls are mutually independent. Then, for $i \in [n]$, we have*

$$\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct}')) = \begin{cases} \mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + \log n \cdot V_{\mathtt{auto}} + 4 \log n \cdot V_{\mathtt{ms}}, & \text{if } i = 0, \\ (\nu_2(i)+1) \cdot V_{\mathtt{auto}} + 4\nu_2(i) \cdot V_{\mathtt{ms}}, & \text{if } i > 0, \end{cases}$$

where $\nu_2(i)$ denotes the 2-adic valuation of i for $i > 0$.

Proof. By Lemma 11, the original noise term $\mathtt{Err}(\mathsf{ct})$ in the input ciphertext is transformed by the scaled trace mapping $\frac{1}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}}(\cdot)$. Then, the linearity of the trace implies that

$$\mathtt{Err}(\mathsf{ct}') = \frac{1}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}}(\mathtt{Err}(\mathsf{ct})) + E_{\mathtt{auto}} + E_{\mathtt{ms}} = \mathtt{Err}(\mathsf{ct})^{(0)} + E_{\mathtt{auto}} + E_{\mathtt{ms}},$$

where $\mathtt{Err}(\mathsf{ct})^{(0)}$ denotes the constant term of $\mathtt{Err}(\mathsf{ct})$ when expressed in the power basis of K over $\mathbb{Q}$ and $E_{\mathtt{auto}}, E_{\mathtt{ms}}$ are the total noise terms accumulated from automorphism evaluation and modulus switching, respectively. Since these three terms are assumed to be independent, for each coefficient index $i \in [n]$, we have

$$\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct}')) = \begin{cases} \mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + \mathsf{Var}_0(E_{\mathtt{auto}}) + \mathsf{Var}_0(E_{\mathtt{ms}}), & \text{if } i = 0, \\ \mathsf{Var}_i(E_{\mathtt{auto}}) + \mathsf{Var}_i(E_{\mathtt{ms}}), & \text{if } i > 0. \end{cases}$$

The desired result then follows by a direct combination of Lemmas 12 and 13. □

Under $V_{\mathtt{ms}} = O(V_{\mathtt{auto}})$, the ratio of the uniform operation noise estimate in [LY26] to our $\nu_2(i)$-dependent bounds is $O(\log n)$ for non-constant coefficients.

4.3 Post-Trace Noise Analysis

Our new analysis provides a more precise coefficient-wise noise estimate for the LY method. In particular, it reveals that the noise variances for different coefficients vary according to the 2-adic valuations of their indices. When trace evaluation is followed by further operations, this coefficient-wise information can yield tighter noise bounds. A common example is multiplication of the output ciphertext by a bounded-norm ring element, as considered in the following corollary.

Corollary 2. *Adopt the settings from Theorem 1. Let* $\mathsf{ct}' \leftarrow \mathsf{LY.EvalScaledTr}_{K/\mathbb{Q}}(\mathsf{ct})$ *(Algorithm 2) be the output ciphertext after trace evaluation on an input ciphertext* ct*, and* $\mathsf{ct}_{\mathtt{pmul}} = b \cdot \mathsf{ct}'$ *be the ciphertext obtained by multiplying* ct' *by a* B*-bounded ring element* $b \in R$*, i.e.,* $\|b\|_\infty \le B$*. Assume heuristically that the coefficients of* $\mathtt{Err}(\mathsf{ct}')$ *are pairwise independent. Then, for each coefficient index* $j \in [n]$*, we have*

$$\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_{\mathtt{pmul}})) \le B^2 \cdot (\mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + 2(n-1) \cdot V_{\mathtt{auto}} + 4(n-1) \cdot V_{\mathtt{ms}}).$$

Proof. Combining Theorem 1, Lemma 3, and the independence heuristic, we have

$$\begin{aligned} &\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_{\mathtt{pmul}})) \le B^2 \cdot \sum_{i \in [n]} \mathsf{Var}_i(\mathtt{Err}(\mathsf{ct}')) \\ &= B^2 \cdot \left(\mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + \log n \cdot (V_{\mathtt{auto}} + 4V_{\mathtt{ms}}) + \sum_{i=1}^{n-1} \left((\nu_2(i) + 1) \cdot V_{\mathtt{auto}} + 4\nu_2(i) \cdot V_{\mathtt{ms}} \right) \right) \\ &= B^2 \cdot (\mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + 2(n-1) \cdot V_{\mathtt{auto}} + 4(n-1) \cdot V_{\mathtt{ms}}) \end{aligned}$$

where the last equality follows from the fact that $\sum_{i=1}^{n-1} \nu_2(i) = n - 1 - \log n$. □

Improvements over Previous Analyses. Assume, as in Corollary 2, that the coefficients of the noise after trace evaluation are pairwise independent. For a fair comparison, we retain the exact contribution from the constant term of the input noise and apply Lemma 3 to the uniform estimate of operation noise in [LY26]. Then, for every $j \in [n]$,

$$\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_{\mathtt{pmul}})) \le B^2 \cdot (\mathsf{Var}_0(\mathtt{Err}(\mathsf{ct})) + n \log n \cdot V_{\mathtt{auto}} + 4n \log n \cdot V_{\mathtt{ms}}).$$

The ratio of this bound to ours in Corollary 2 is maximized when the constant term of input noise is negligible compared to the additive noise growth. In this case, it lies between $n \log n/(2(n-1))$ and $n \log n/(n-1)$, depending on the relative contributions of $V_{\mathtt{auto}}$ and $V_{\mathtt{ms}}$. Hence, our bound is tighter by a factor of $O(\log n)$.

5 Algebraic Analyses of Algorithmic Applications

In this section, we explore how our new algebraic insights can be applied to two representative algorithmic applications of homomorphic trace evaluation: (1) ciphertext packing, as introduced in [CDKS21] and recently improved in [LY26] with reduced noise growth; and (2) coefficient extraction, as introduced in [ACLS18] and subsequently developed in [GH19, CCR19, KLD+23, BMW24]. We provide refined algebraic analyses and tighter noise estimates for both applications when the ciphertext modulus q is a power of two, as in FFT-based implementations. Note that the analysis also applies to NTT-based implementations with $\gcd(n, q) = 1$, where the modulus switching/raising steps can be replaced by exact multiplications by the appropriate inverses modulo q.

5.1 Ciphertext Packing

The CDKS packing method [CDKS21] enables packing multiple LWE ciphertexts into a single RLWE ciphertext, with messages encoded in the coefficients with respect to the power basis of K over its subfields. Here we focus on the full packing case, namely:

- The input includes n LWE ciphertexts encrypting $m_0, \ldots, m_{n-1} \in \mathbb{Z}$ under the same secret key; the output is a single RLWE ciphertext that encrypts $m = \sum_{i=0}^{n-1} m_i \cdot \zeta_{2n}^i$.
- The input LWE ciphertexts are assumed to be of the form $(\boldsymbol{a}, b) \in \mathbb{Z}_q^{n+1}$, with modulus q and dimension n matching the modulus and ring degree of the RLWE scheme. The underlying secret key $\boldsymbol{s} \in \mathbb{Z}_q^n$ is assumed to coincide, up to some signed permutation, with the coefficient vector of the RLWE secret key. These compatibility requirements are generally manageable: when parameters differ, standard key and modulus switching can align the ciphertexts, at the cost of additional computation and noise growth.

Algebraic Reanalysis. CDKS packing starts from a natural coefficient embedding (see Appendix A), under which each of these LWE ciphertexts is interpreted as an RLWE ciphertext encrypting a ring element $M_i \in R$. The constant coefficient of M_i is m_i, while its remaining coefficients carry no semantic information. Specifically, for $i \in [n]$, the original n input LWE ciphertexts can be directly viewed as n RLWE encodings of

$$M_i := m_i + \sum_{j=1}^{n-1} c_{i,j} \cdot \zeta_{2n}^j \quad \text{for some } c_{i,j} \in \mathbb{Z}.$$

Then, by the property of the trace mapping, packing the m_i's into the coefficients can be achieved by homomorphically evaluating

$$\frac{1}{n} \cdot \sum_{i \in [n]} \mathsf{Tr}_{K/\mathbb{Q}}(M_i) \cdot \zeta_{2n}^i = \sum_{i \in [n]} m_i \cdot \zeta_{2n}^i = m. \tag{1}$$

If each trace evaluation is performed independently, the total number of automorphism evaluations would be $n \log n$. Instead, by employing a divide-and-conquer strategy, CDKS packing can be accomplished with only $n-1$ automorphism evaluations [CDKS21]. However, the recursive formulation makes both the algorithm and its analysis less transparent. Building on the algebraic insights from Section 4, we instead provide a top-down analysis that leads to a non-recursive variant of CDKS packing.

Lemma 14. *Let $\hat{\tau}_k$ be the $\mathbb{Q}$-linear operator defined as $\hat{\tau}_k(a) = a + \hat{\sigma}_k(a)$ for $a \in K$ where $\hat{\sigma}_k \in \mathsf{Gal}(K/\mathbb{Q})$ is defined by $\zeta_{2n} \mapsto \zeta_{2n}^{1+2^k}$ for $k = 1, \ldots, \log n$. Then, for any $a \in K$, we have $\hat{\tau}_k(\zeta_{2n}^{2^\ell} \cdot a) = \zeta_{2n}^{2^\ell} \cdot \hat{\tau}_k(a)$ for all $\ell \geq \log n - k + 1$.*

Proof. Since $\ell \geq \log n - k + 1$, we have $(\zeta_{2n}^{2^\ell})^{2^k} = (\zeta_{2n}^{2n})^{2^{\ell+k-\log n-1}} = 1$. Thus,

$$\hat{\tau}_k(\zeta_{2n}^{2^\ell} \cdot a) = \zeta_{2n}^{2^\ell} \cdot a + \zeta_{2n}^{2^\ell(1+2^k)} \cdot \hat{\sigma}_k(a) = \zeta_{2n}^{2^\ell} \cdot a + \zeta_{2n}^{2^\ell} \cdot \hat{\sigma}_k(a) = \zeta_{2n}^{2^\ell} \cdot \hat{\tau}_k(a). \qquad \square$$

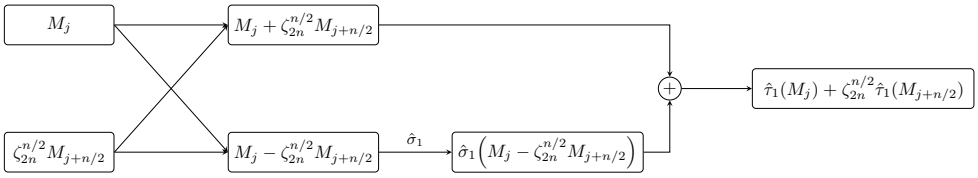

Figure 1: One butterfly-style folding step at the first level of the non-recursive packing procedure. Two neighboring inputs are first combined into a sum and a difference; then a single automorphism evaluation on the difference suffices to obtain the desired folded output. The same pattern repeats at later levels with $\zeta_{2n}^{n/2^k}$ and $\hat{\sigma}_k$.

For an index $i \in [n]$, write the binary expansion of i as $i = \sum_{\ell \in [\log n]} i\langle \ell \rangle \cdot 2^\ell$, where $i\langle \ell \rangle \in \{0, 1\}$ is the ℓ-th least significant bit of i. Then, by Lemma 11 and Lemma 14, we can rewrite the formula that needs to be homomorphically evaluated in Equation (1) (without the scaling factor $1/n$) as

$$\begin{aligned}
&\sum_{i \in [n]} \zeta_{2n}^{i} \cdot \mathsf{Tr}_{K/\mathbb{Q}}(M_i) \\
&= \sum_{i \in [n]} \zeta_{2n}^{\sum_{\ell \in [\log n]} i\langle \ell \rangle \cdot 2^\ell} \cdot \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_2 \circ \hat{\tau}_1(M_i) \\
&= \sum_{i \in [n]} \left(\prod_{\ell \in [\log n]} \zeta_{2n}^{i\langle \ell \rangle \cdot 2^\ell} \right) \cdot \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_2 \circ \hat{\tau}_1(M_i) \\
&= \sum_{i \in [n]} \zeta_{2n}^{i\langle 0 \rangle} \cdot \hat{\tau}_{\log n} \left(\zeta_{2n}^{i\langle 1 \rangle \cdot 2} \cdot \hat{\tau}_{\log n - 1} \left(\cdots \hat{\tau}_2 \left(\zeta_{2n}^{i\langle \log n - 1 \rangle \cdot \frac{n}{2}} \cdot \hat{\tau}_1(M_i) \right) \cdots \right) \right). \qquad (2)
\end{aligned}$$

This formula can be evaluated in a tree-like manner using the linearity of $\hat{\tau}_k$'s. For example, when fixing $i\langle 0 \rangle, \ldots, i\langle \log n - 2 \rangle$, the innermost summation that can be combined is

$$\hat{\tau}_1(M_j) + \zeta_{2n}^{\frac{n}{2}} \cdot \hat{\tau}_1\left(M_{j+\frac{n}{2}}\right) \quad \text{for} \quad j = \sum_{k=0}^{\log n - 2} i\langle k \rangle \cdot 2^k.$$

The total number of $\hat{\tau}_1$ evaluations is n, which implies n evaluations of $\hat{\sigma}_1$. To further reduce this, we can rewrite the above summation as

$$\begin{aligned}
\hat{\tau}_1(M_j) + \zeta_{2n}^{\frac{n}{2}} \cdot \hat{\tau}_1\left(M_{j+\frac{n}{2}}\right) &= M_j + \hat{\sigma}_1(M_j) + \zeta_{2n}^{\frac{n}{2}} \cdot \left(M_{j+\frac{n}{2}} + \hat{\sigma}_1\left(M_{j+\frac{n}{2}}\right)\right) \\
&= \left(M_j + \zeta_{2n}^{\frac{n}{2}} \cdot M_{j+\frac{n}{2}}\right) + \left(\hat{\sigma}_1(M_j) + \zeta_{2n}^{\frac{n}{2}} \cdot \hat{\sigma}_1\left(M_{j+\frac{n}{2}}\right)\right) \\
&= \left(M_j + \zeta_{2n}^{\frac{n}{2}} \cdot M_{j+\frac{n}{2}}\right) + \hat{\sigma}_1\left(M_j - \zeta_{2n}^{\frac{n}{2}} \cdot M_{j+\frac{n}{2}}\right)
\end{aligned}$$

where the last equality follows from the fact that $\hat{\sigma}_1(\zeta_{2n}^{\frac{n}{2}}) = \zeta_{2n}^{\frac{n}{2} \cdot 3} = \zeta_{2n}^{n} \cdot \zeta_{2n}^{\frac{n}{2}} = -\zeta_{2n}^{\frac{n}{2}}$. Thus, by first computing $M_j + \zeta_{2n}^{\frac{n}{2}} \cdot M_{j+\frac{n}{2}}$ and $M_j - \zeta_{2n}^{\frac{n}{2}} \cdot M_{j+\frac{n}{2}}$ for each j, we can evaluate all the above summations with only $n/2$ evaluations of $\hat{\sigma}_1$. Figure 1 illustrates this local butterfly structure at the first level.

Applying the same idea to subsequent levels allows the entire packing procedure to be performed with only $\sum_{k=1}^{\log n} n/2^k = n - 1$ automorphism evaluations, matching the complexity of the original CDKS packing [CDKS21] but with an iterative structure. We extend this idea with the optimizations from [LY26] to obtain a non-recursive variant[2]

[2]Note that closely related iterative (non-recursive) butterfly realizations and structured specializations of the original CDKS packing approach have appeared in the literature in a variety of contexts and under varying terminology; see, e.g., [LHG+25, LSL+25, IB25, BI26].

Algorithm 3 A Non-Recursive Variant of CDKS-LY Packing (In-Place Version)

Input: RLWE ciphertexts $\mathsf{ct}_i \in \mathsf{RLWE}_{\mathtt{sk}}(M_i)$ for $i \in [n]$ where the constant term of M_i with respect to the power basis of K over $\mathbb{Q}$ is $m_i \in \mathbb{Z}$

Input: Automorphism keys for $\hat{\sigma}_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+2^k}$ where $k = 1, \ldots, \log n$

Output: A RLWE ciphertext $\mathsf{ct}' \in \mathsf{RLWE}_{\mathtt{sk}}(m)$ where $m = \sum_{i=0}^{n-1} m_i \cdot \zeta_{2n}^i$

1: **for** $k = 1$ to $\log n$ **do**
2: **for** $j = 0$ to $n/2^k - 1$ **do**
3: $\widetilde{\mathsf{ct}} \leftarrow \mathsf{ct}_j - \zeta_{2n}^{n/2^k} \cdot \mathsf{ct}_{j+n/2^k}$
4: $\widetilde{\mathsf{ct}} \leftarrow \mathsf{ModRaise}_{q/2 \to q} \circ \mathsf{ModSwitch}_{q \to q/2}(\widetilde{\mathsf{ct}})$
5: $\mathsf{ct}_j \leftarrow \widetilde{\mathsf{ct}} + \zeta_{2n}^{n/2^k} \cdot \mathsf{ct}_{j+n/2^k} + \mathsf{EvalAuto}(\widetilde{\mathsf{ct}}, 2^k + 1)$
6: **end for**
7: **end for**
8: **return** $\mathsf{ct}' \leftarrow \mathsf{ct}_0$

of CDKS-LY packing as shown in Algorithm 3. The corresponding coefficient-wise noise analysis is provided in Theorem 2.

Theorem 2. *Adopt the notations of Algorithm 3 and Theorem 1, and let* ct_i *on the right-hand side denote the* i*-th input ciphertext. For each* $i \in [n]$*, assume heuristically that the constant term of* $\mathtt{Err}(\mathsf{ct}_i)$ *and the noise contributions from all distinct* ModSwitch *and* EvalAuto *calls are mutually independent. Then, the algorithm is correct, and its output ciphertext* ct' *satisfies the following equality for every* $i \in [n]$*:*

$$\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct}')) = \mathsf{Var}_0\left(\mathtt{Err}(\mathsf{ct}_i)\right) + \log n \cdot V_{\mathtt{auto}} + 4\omega(i) \cdot V_{\mathtt{ms}}$$

where $\omega(i)$ *is the number of zero bits in the binary representation of* i *with bit-length* $\log n$*.*

Proof. The correctness follows from the fact that the algorithm evaluates the formula in Equation (2) with a scaling factor of $1/n$ due to the modulus switching at each level. Therefore, the phase of the output ciphertext contains

$$\frac{1}{n} \cdot \sum_{i \in [n]} \mathsf{Tr}_{K/\mathbb{Q}}(M_i + \mathtt{Err}(\mathsf{ct}_i)) \cdot \zeta_{2n}^i = \sum_{i=0}^{n-1} m_i \cdot \zeta_{2n}^i + \sum_{i=0}^{n-1} e_i \cdot \zeta_{2n}^i \tag{3}$$

where e_i is the constant term of $\mathtt{Err}(\mathsf{ct}_i)$ with respect to the power basis of K over $\mathbb{Q}$. Another essential class consists of the terms introduced by $\mathsf{ModRaise}_{q/2 \to q}$. We write the term introduced in the j-th ciphertext at level k as $\frac{q}{2} I_{k,j}$ for $k = 1, \ldots, \log n$ and $j \in [n/2^k]$. According to Equation (2), the contribution of the additional terms introduced in the k-th level to the final phase is

$$\begin{aligned}
\frac{2^k}{n} \cdot \sum_{j \in [n/2^k]} \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_k \left(\frac{q}{2} \cdot I_{k,j}\right) \cdot \zeta_{2n}^j &= \frac{2^k}{n} \cdot \sum_{j \in [n/2^k]} \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})} \left(\frac{q}{2} \cdot I_{k,j}\right) \cdot \zeta_{2n}^j \\
&= \frac{2^k}{n} \cdot \sum_{j \in [n/2^k]} \frac{n}{2^{k-1}} \cdot \frac{q}{2} \cdot I_{k,j}^{(0)} \cdot \zeta_{2n}^j \\
&= q \cdot \sum_{j \in [n/2^k]} I_{k,j}^{(0)} \cdot \zeta_{2n}^j \\
&\equiv 0 \pmod{q}
\end{aligned}$$

where $I_{k,j}^{(0)} \in \mathbb{Z}[\zeta_{2^k}]$ is the constant term of $I_{k,j}$ with respect to the power basis of K over $\mathbb{Q}(\zeta_{2^k})$. Thus, these additional terms vanish in the final phase.

For the noise analysis, we show how the noise introduced by modulus switching and automorphism evaluation contributes to the final noise. Let $E_{\mathtt{ms}}$ denote the noise introduced by modulus switching in the final phase and $E_{k,j}$ denote the noise introduced by modulus switching in the j-th output ciphertext of the k-th level for $k = 1, \ldots, \log n$ and $j \in [n/2^k]$. By a similar argument as above, the aggregate contribution of level-k terms to $E_{\mathtt{ms}}$ is

$$\begin{aligned}\frac{2^k}{n} \sum_{j \in [n/2^k]} \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_k(E_{k,j}) \cdot \zeta_{2n}^j &= \frac{2^k}{n} \sum_{j \in [n/2^k]} \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^k})}(E_{k,j}) \cdot \zeta_{2n}^j \\ &= \sum_{j \in [n/2^k]} 2E_{k,j}^{(0)} \cdot \zeta_{2n}^j\end{aligned}$$

where $E_{k,j}^{(0)}$ is the constant term of $E_{k,j}$ with respect to the power basis of K over $\mathbb{Q}(\zeta_{2^k})$. For a fixed coefficient index $i \in [n]$ for $E_{\mathtt{ms}}$, the contribution from level k appears only if the $(\log n - k)$-th bit of i equals zero, in which case it contributes one coefficient of the term $2E_{k,[i]_{n/2^k}}^{(0)}$. Hence, the i-th coefficient of $E_{\mathtt{ms}}$ is the sum of the contributions from all levels k such that $i\langle \log n - k \rangle = 0$. As k ranges from 1 to $\log n$, the number of such levels exactly corresponds to the number of zero bits in the binary representation of i, which is denoted as $\omega(i)$. Therefore, we have

$$\mathsf{Var}_i(E_{\mathtt{ms}}) = \sum_{k \,:\, i\langle \log n - k \rangle = 0} \mathsf{Var}\left(2 \cdot E_{k,[i]_{n/2^k}}^{(0)}\right) = 4 \cdot \omega(i) \cdot V_{\mathtt{ms}}. \tag{4}$$

Similarly, let $E_{\mathtt{auto}}$ denote the noise introduced by automorphism evaluation in the final phase and $\tilde{E}_{k,j}$ denote the noise introduced by automorphism evaluation in the j-th ciphertext of the k-th level for $k = 1, \ldots, \log n$ and $j \in [n/2^k]$. By a similar argument, for $k = 1, \ldots, \log n - 1$, the aggregate contribution of level-k terms $\tilde{E}_{k,j}$ to $E_{\mathtt{auto}}$ is

$$\begin{aligned}\frac{2^k}{n} \sum_{j \in [n/2^k]} \hat{\tau}_{\log n} \circ \cdots \circ \hat{\tau}_{k+1}\left(\tilde{E}_{k,j}\right) \cdot \zeta_{2n}^j &= \frac{2^k}{n} \sum_{j \in [n/2^k]} \mathsf{Tr}_{K/\mathbb{Q}(\zeta_{2^{k+1}})}\left(\tilde{E}_{k,j}\right) \cdot \zeta_{2n}^j \\ &= \sum_{j \in [n/2^k]} \tilde{E}_{k,j}^{(0)} \cdot \zeta_{2n}^j\end{aligned}$$

where $\tilde{E}_{k,j}^{(0)}$ is the constant term of $\tilde{E}_{k,j}$ with respect to the power basis of K over $\mathbb{Q}(\zeta_{2^{k+1}})$. In this case, $\{\zeta_{2n}^j\}_{j \in [n/2^k]}$ is the power basis of K over $\mathbb{Q}(\zeta_{2^{k+1}})$, so the i-th coefficient of $E_{\mathtt{auto}}$ receives contributions from exactly one coefficient of $\tilde{E}_{k,[i]_{n/2^k}}^{(0)}$ for each $k = 1, \ldots, \log n - 1$. Therefore, for each coefficient index $i \in [n]$, we have

$$\mathsf{Var}_i(E_{\mathtt{auto}}) = \sum_{k=1}^{\log n - 1} \mathsf{Var}\left(\tilde{E}_{k,[i]_{n/2^k}}\right) + \mathsf{Var}\left(\tilde{E}_{\log n}\right) = \log n \cdot V_{\mathtt{auto}}. \tag{5}$$

Combining the fact that $\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct}')) = \mathsf{Var}_0\left(\mathtt{Err}(\mathsf{ct}_i)\right) + \mathsf{Var}_i(E_{\mathtt{auto}}) + \mathsf{Var}_i(E_{\mathtt{ms}})$ and the results in Equations (3), (4) and (5), we complete the proof. □

5.2 Coefficient Extraction

Coefficient extraction, which can be regarded as dual to packing, aims to homomorphically extract the coefficients with respect to the power basis of K over its subfields. In the full extraction setting, given a ciphertext $\mathsf{ct} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$ with $m = \sum_{i \in [n]} m_i \cdot \zeta_{2n}^i$ for $m_i \in \mathbb{Z}$, the coefficient extraction algorithm outputs ciphertexts $\mathsf{ct}_i \in \mathsf{RLWE}_{\mathtt{sk}}(m_i)$ for $i \in [n]$. Note that this task is distinct from the sample extraction technique used in FHEW/TFHE bootstrapping [DM15, CGGI20], where the output collapses to a plain-LWE form rather than the RLWE form required here.

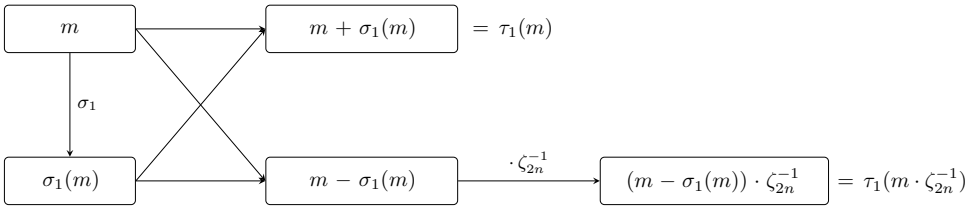

Figure 2: One butterfly-style splitting step at the first level of the coefficient extraction procedure. The input m first branches into m and its automorphism image $\sigma_1(m)$. These two values are then combined into a sum and a difference, finally yielding $\tau_1(m)$ and $\tau_1(m \cdot \zeta_{2n}^{-1})$, respectively. The same pattern repeats at later levels with $\sigma_{\ell+1}$ and $\zeta_{2n}^{-2^\ell}$.

Algebraic Reanalysis. Since the trace mapping zeroes out the non-constant terms, coefficient extraction can be naturally achieved by rotating each coefficient to the constant term and then applying the trace mapping. Namely, we need to homomorphically evaluate

$$\frac{1}{n} \cdot \mathsf{Tr}_{K/\mathbb{Q}}\left(m \cdot \zeta_{2n}^{-i}\right) = m_i \quad \text{for all} \quad i = 0, \ldots, n-1. \tag{6}$$

A direct implementation requires n trace evaluations, which amounts to $n \log n$ automorphism evaluations in total. However, by using tree-based evaluation patterns as proposed in [ACLS18], the number of automorphism evaluations can be reduced to $O(n)$. To see this intuitively, we can use the following lemma, which is analogous to Lemma 14.

Lemma 15. *Let τ_k be the $\mathbb{Q}$-linear operator defined as $\tau_k(a) = a + \sigma_k(a)$ for $a \in K$ where $\sigma_k \in \mathsf{Gal}(K/\mathbb{Q})$ is determined by $\zeta_{2n} \mapsto \zeta_{2n}^{1+n/2^{k-1}}$ for $k = 1, \ldots, \log n$. Then, for any $a \in K$, we have $\tau_k(\zeta_{2n}^{-2^\ell} \cdot a) = \zeta_{2n}^{-2^\ell} \cdot \tau_k(a)$ for all $\ell \geq k$.*

Proof. Since $\ell \geq k$, we have $\zeta_{2n}^{-2^\ell \cdot n/2^{k-1}} = (\zeta_{2n}^{2n})^{-2^{\ell-k}} = 1$. Thus,

$$\tau_k(\zeta_{2n}^{-2^\ell} \cdot a) = \zeta_{2n}^{-2^\ell} \cdot a + \zeta_{2n}^{-2^\ell(1+n/2^{k-1})} \cdot \sigma_k(a) = \zeta_{2n}^{-2^\ell} \cdot a + \zeta_{2n}^{-2^\ell} \cdot \sigma_k(a) = \zeta_{2n}^{-2^\ell} \cdot \tau_k(a). \quad \square$$

Combining this lemma with the binary expansion of i, written as $i = \sum_{\ell \in [\log n]} i\langle \ell \rangle \cdot 2^\ell$ where $i\langle \ell \rangle \in \{0, 1\}$, we can rewrite the formula in Equation (6) (omitting $1/n$) as

$$\begin{aligned} \mathsf{Tr}_{K/\mathbb{Q}}\left(m \cdot \zeta_{2n}^{-i}\right) =& \tau_{\log n} \circ \cdots \circ \tau_2 \circ \tau_1 \left(m \cdot \zeta_{2n}^{-\sum_{\ell \in [\log n]} i\langle \ell \rangle \cdot 2^\ell}\right) \\ =& \tau_{\log n}\left(\tau_{\log n - 1}\left(\cdots \tau_2\left(\tau_1\left(m \cdot \zeta_{2n}^{-i\langle 0 \rangle}\right) \cdot \zeta_{2n}^{-i\langle 1 \rangle \cdot 2}\right) \cdots\right) \cdot \zeta_{2n}^{-i\langle \log n - 1 \rangle \cdot \frac{n}{2}}\right). \end{aligned} \tag{7}$$

To evaluate this formula, we can also employ a tree-like pattern by reusing the same intermediate results when the indices share common least significant bits. For example, all indices with $i\langle 0 \rangle = 0$ (even indices) share the same innermost evaluation $\tau_1(m)$, while those with $i\langle 0 \rangle = 1$ (odd indices) share $\tau_1(m \cdot \zeta_{2n}^{-1})$. More interestingly, these two operations can be combined by noticing that $\tau_1(m) = m + \sigma_1(m)$ and

$$\tau_1(m \cdot \zeta_{2n}^{-1}) = m \cdot \zeta_{2n}^{-1} + \sigma_1(m \cdot \zeta_{2n}^{-1}) = m \cdot \zeta_{2n}^{-1} + \sigma_1(m) \cdot \zeta_{2n}^{-(1+n)} = (m - \sigma_1(m)) \cdot \zeta_{2n}^{-1}.$$

Thus, the automorphism evaluation $\sigma_1(m)$ only needs to be performed once to obtain both $\tau_1(m)$ and $\tau_1(m \cdot \zeta_{2n}^{-1})$. Figure 2 illustrates this local butterfly structure at the first level.

By applying the same idea to subsequent levels, the entire coefficient extraction procedure can be performed with only $\sum_{\ell \in [\log n]} 2^\ell = n - 1$ automorphism evaluations [KLD+23, BMW24]. Unlike the reverse order used in the LY method, the order $\tau_1, \ldots, \tau_{\log n}$ used here does not make the composition of the operators remaining after

Algorithm 4 Coefficient Extraction

Input: A RLWE ciphertext $\mathsf{ct} \in \mathsf{RLWE}_{\mathtt{sk}}(m)$ where $m = \sum_{i\in[n]} m_i \cdot \zeta_{2n}^i$ for $m_i \in \mathbb{Z}$

Input: Automorphism keys for $\sigma_k : \zeta_{2n} \mapsto \zeta_{2n}^{1+n/2^{k-1}}$ where $k = 1, \ldots, \log n$

Output: RLWE ciphertexts $\mathsf{ct}_i \in \mathsf{RLWE}_{\mathtt{sk}}(m_i)$ for $i \in [n]$

1: $\mathsf{ct}_0 \leftarrow \mathsf{ModRaise}_{q/n\to q} \circ \mathsf{ModSwitch}_{q\to q/n}(\mathsf{ct})$
2: **for** $\ell = 0$ to $\log n - 1$ **do**
3: **for** $j = 0$ to $2^\ell - 1$ **do**
4: $\widetilde{\mathsf{ct}} \leftarrow \mathsf{EvalAuto}(\mathsf{ct}_j, 1 + n/2^\ell)$
5: $\mathsf{ct}_{j+2^\ell} \leftarrow (\mathsf{ct}_j - \widetilde{\mathsf{ct}}) \cdot \zeta_{2n}^{-2^\ell}$
6: $\mathsf{ct}_j \leftarrow \mathsf{ct}_j + \widetilde{\mathsf{ct}}$
7: **end for**
8: **end for**
9: **return** $\{\mathsf{ct}_i\}_{i\in[n]}$

each level a relative trace that merely zeroes out or scales coefficients. Consequently, modulus switching and raising cannot be inserted at every level to control noise growth. Nevertheless, the WHS$^+$ method (as revisited in Section 3.2) can still be applied, and our new algebraic analysis yields tighter coefficient-wise noise bounds in this case. The detailed algorithm and analysis are given in Algorithm 4 and Theorem 3, respectively.

Theorem 3. *Adopt the notations in Algorithm 4. Suppose the noise terms introduced by* $\mathsf{ModSwitch}$ *and* $\mathsf{EvalAuto}$ *are independent of* $\mathtt{Err}(\mathsf{ct})$ *and follow the same assumptions as in Theorem 1 with variance* $V_{\mathtt{ms}}$ *and* $V_{\mathtt{auto}}$*, respectively. Assume heuristically that all coefficients of the noise introduced by* $\mathsf{EvalAuto}$ *are mutually independent. Then, the algorithm is correct, and for each* $i, j \in [n]$*, the coefficient-wise variance of* $\mathtt{Err}(\mathsf{ct}_i)$ *satisfies*

$$\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_i)) = \begin{cases} \mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + \frac{1}{3}(n^2-1)\cdot V_{\mathtt{auto}} + n^2 \cdot V_{\mathtt{ms}} & \text{if } j = 0 \\ \left(\frac{n}{2^{\nu_2(j)}} - 1\right)\cdot V_{\mathtt{auto}} & \text{if } j > 0, \end{cases}$$

where $\nu_2(j)$ *is the 2-adic valuation of* j *for* $j > 0$*.*

Proof. The correctness follows from the fact that the algorithm exactly evaluates the formula in Equation (6) using the pattern in Equation (7) with a scaling factor of $1/n$ due to the modulus switching in *line* 1. Let $e = \mathtt{Err}(\mathsf{ct})$ be the input noise, $e_{\mathtt{ms}}$ the noise introduced by $\mathsf{ModSwitch}_{q\to q/n}$, and $\frac{q}{n}I$ with $I \in R$ the additional term introduced by $\mathsf{ModRaise}_{q/n\to q}$. After line 1, the phase of ct_0 has the form $\frac{1}{n}(m+e) + e_{\mathtt{ms}} + \frac{q}{n}I$. Its contribution to the i-th output ciphertext ct_i is therefore

$$\begin{aligned} \mathsf{Tr}_{K/\mathbb{Q}}\left(\left(\frac{m}{n} + \frac{e}{n} + e_{\mathtt{ms}} + \frac{q}{n}I\right)\cdot\zeta_{2n}^{-i}\right) &= n\cdot\left(\frac{m_i}{n} + \frac{e_i}{n} + e_{\mathtt{ms}}^{(i)} + \frac{q}{n}I_i\right) \\ &= m_i + e_i + n\cdot e_{\mathtt{ms}}^{(i)} \pmod q \end{aligned} \tag{8}$$

where m_i, e_i, $e_{\mathtt{ms}}^{(i)}$, and I_i are the i-th coefficients of m, e, $e_{\mathtt{ms}}$, and I with respect to the power basis of K over $\mathbb{Q}$, respectively. The remaining task is to analyze how the noise terms introduced by automorphism evaluations contribute to the noise terms of each output ciphertext. Let E_i denote the accumulated noise introduced by automorphism evaluations in the i-th output ciphertext ct_i for $i \in [n]$, and write $i = \sum_{\ell\in[\log n]} i\langle\ell\rangle \cdot 2^\ell$ as the binary expansion of i. By Equation (7) and *line* 5, we have

$$\begin{aligned} E_i = \sum_{k=1}^{\log n-1} \tau_{\log n} \circ \cdots \circ \tau_{k+1}\left((-1)^{\left\lfloor \frac{i}{2^{k-1}}\right\rfloor}\cdot \tilde{E}_{k,\,[i]_{2^{k-1}}}\cdot \zeta_{2n}^{-\sum_{\ell=k-1}^{\log n-1} i\langle\ell\rangle\cdot 2^\ell}\right) \\ + (-1)^{\left\lfloor \frac{i}{n/2}\right\rfloor}\cdot \tilde{E}_{\log n,\,[i]_{n/2}}\cdot \zeta_{2n}^{-i\langle\log n-1\rangle\cdot\frac{n}{2}} \end{aligned} \tag{9}$$

where $\tilde{E}_{k,j}$ denotes the noise introduced by the automorphism evaluation in inner-loop iteration j of outer-loop iteration $k-1$.

For ease of analysis, define $T_k = \tau_{\log n} \circ \cdots \circ \tau_{k+1}$ and let $T_{\log n} = \mathsf{id}$ be the identity operator. In fact, $T_k|_{\mathbb{Q}(\zeta_{2n/2^k})} = \mathsf{Tr}_{\mathbb{Q}(\zeta_{2n/2^k})/\mathbb{Q}}$, or equivalently, T_k is a particular $\mathbb{Q}$-linear lift of the trace mapping $\mathsf{Tr}_{\mathbb{Q}(\zeta_{2n/2^k})/\mathbb{Q}}$ to K. We model the variables $\tilde{E}_{k,j}$ as independent across different levels k, with all coefficients having common variance $V_{\mathtt{auto}}$. Since multiplication by a root of unity only permutes coefficients up to sign, it preserves their common variance. Thus, Equation (9) gives $\mathsf{Var}_j(E_i) = \mathsf{Var}_j(E'_i)$ for all $j \in [n]$, where

$$E'_i := \sum_{k=1}^{\log n - 1} \tau_{\log n} \circ \cdots \circ \tau_{k+1}\left(\tilde{E}_k\right) + \tilde{E}_{\log n} = \sum_{k=1}^{\log n} T_k(\tilde{E}_k)$$

where $\tilde{E}_k \in R$ and $\mathsf{Var}(\tilde{E}_k) = V_{\mathtt{auto}}$ for all $k = 1, \ldots, \log n$. Expanding T_k gives $T_k = \sum_{s \in \mathcal{S}_k} \rho_s$ where $\rho_s \in \mathsf{Gal}(K/\mathbb{Q})$ is determined by $\zeta_{2n} \mapsto \zeta_{2n}^s$ for $s \in \mathcal{S}_k$ and

$$\mathcal{S}_k := \left\{ \left[\prod_{t=k}^{\log n - 1} \left(1 + \frac{n}{2^t}\right)^{\epsilon_t} \right]_{2n} \;\middle|\; \epsilon_t \in \{0,1\} \right\} \quad \text{with} \quad \mathcal{S}_{\log n} = \{1\}.$$

Moreover, $|\mathcal{S}_k| = n/2^k$ and $\mathcal{S}_k \bmod 2n/2^k = \mathbb{Z}^*_{2n/2^k}$ (see Appendix B), which means that the elements in $\mathcal{S}_k$ are distinct modulo $2n/2^k$.

Now we analyze the term $T_k(\tilde{E}_k) = \sum_{s \in \mathcal{S}_k} \rho_s(\tilde{E}_k)$. Since ρ_s permutes (up to a sign) the coefficients of $\tilde{E}_k$, the u-th coefficient of $\tilde{E}_k$ will contribute to the $[s \cdot u]_n$-th coefficient of $\rho_s(\tilde{E}_k)$ for each $s \in \mathcal{S}_k$ and $u \in [n]$. For $j \in [n]$, we consider the following two cases for the j-th coefficient of $T_k(\tilde{E}_k)$:

- *Case 1:* If $2^k \mid j$, suppose the u-th coefficient of $\tilde{E}_k$ contributes to the j-th coefficient of $\rho_s(\tilde{E}_k)$ for some $s \in \mathcal{S}_k$. Then, we have $[s \cdot u]_n = j$, so $2^k \mid [s \cdot u]_n$. Since $2^k \mid n$ and $\gcd(2^k, s) = 1$, we have $2^k \mid u$. Hence, the j-th coefficient of $T_k(\tilde{E}_k)$ can only receive contributions from the coefficients of $\tilde{E}_k$ with indices divisible by 2^k, which collectively form the component lying in the subfield $\mathbb{Q}(\zeta_{2n/2^k})$. As $T_k|_{\mathbb{Q}(\zeta_{2n/2^k})} = \mathsf{Tr}_{\mathbb{Q}(\zeta_{2n/2^k})/\mathbb{Q}}$, only the constant term (i.e., $j = 0$) of $T_k(\tilde{E}_k)$ receives a non-zero contribution of the form $\frac{n}{2^k} \cdot \tilde{E}_k^{(0)}$ where $\tilde{E}_k^{(0)}$ is the constant term of $\tilde{E}_k$ with respect to the power basis of K over $\mathbb{Q}$. Therefore, the constant coefficient of $T_k(\tilde{E}_k)$ has variance $(n/2^k)^2 \cdot V_{\mathtt{auto}}$.
- *Case 2:* If $2^k \nmid j$, we use the basic fact that the j-th coefficient of $T_k(\tilde{E}_k)$ receives contributions from the $[s^{-1}j]_n$-th coefficient of $\tilde{E}_k$ for each $s \in \mathcal{S}_k$. We will show that $|\{[s^{-1}j]_n\}_{s \in \mathcal{S}_k}| = |\mathcal{S}_k| = n/2^k$. Define the map $\phi_j : \mathcal{S}_k \longrightarrow [n]$ as $\phi_j(s) = [s^{-1}j]_n$. It suffices to show ϕ_j is injective. Suppose $\phi_j(s_1) = \phi_j(s_2)$ for some $s_1, s_2 \in \mathcal{S}_k$. Then $[s_1^{-1}j]_n = [s_2^{-1}j]_n$. Write $j = 2^v \cdot j'$ where $v := \nu_2(j) < k$ and j' odd. Then, we have $[s_1^{-1}j']_{n/2^v} = [s_2^{-1}j']_{n/2^v}$, which implies that $[s_1]_{n/2^v} = [s_2]_{n/2^v}$ since $\gcd(j', n/2^v) = 1$. Since $v < k$, we have $2n/2^k \mid n/2^v$, so $[s_1]_{2n/2^k} = [s_2]_{2n/2^k}$. Since the elements of $\mathcal{S}_k$ are distinct modulo $2n/2^k$, we have $s_1 = s_2$. Therefore, the j-th coefficient of $T_k(\tilde{E}_k)$ is a linear combination of $n/2^k$ distinct coefficients of $\tilde{E}_k$ with weights in $\{\pm 1\}$ and hence has variance $(n/2^k) \cdot V_{\mathtt{auto}}$.

Based on the above analysis, we can derive $\mathsf{Var}_j(E_i)$ from $\mathsf{Var}_j(E'_i)$ for each $j \in [n]$. For $j = 0$, the condition $2^k \mid j$ holds for every k, so all levels contribute to the constant coefficient of E'_i. Thus, we have

$$\mathsf{Var}_0(E_i) = \mathsf{Var}_0(E'_i) = \sum_{k=1}^{\log n} \left(\frac{n}{2^k}\right)^2 \cdot V_{\mathtt{auto}} = \frac{1}{3}(n^2 - 1) \cdot V_{\mathtt{auto}}.$$

Hence, $\mathsf{Var}_0(\mathtt{Err}(\mathsf{ct}_i)) = \mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + \frac{1}{3}(n^2-1) \cdot V_{\mathtt{auto}} + n^2 \cdot V_{\mathtt{ms}}$ by Equation (8).

For $j > 0$, the j-th coefficient of E_i receives contributions only from levels k such that $2^k \nmid j$, which is equivalent to $k > \nu_2(j)$. Therefore, we have

$$\mathsf{Var}_j(E_i) = \mathsf{Var}_j(E_i') = \sum_{k=\nu_2(j)+1}^{\log n} \frac{n}{2^k} \cdot V_{\mathtt{auto}} = \left(\frac{n}{2^{\nu_2(j)}} - 1\right) \cdot V_{\mathtt{auto}}.$$

Since $\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_i)) = \mathsf{Var}_j(E_i)$ for $j > 0$, we complete the proof. □

Post-Extraction Noise Analysis. As in many applications of coefficient extraction, such as query expansion in private information retrieval protocols [ACLS18, ALP+21, MCR21, MW22, BMW24], the extracted ciphertexts are typically subjected to further homomorphic operations, including plaintext-ciphertext multiplication. Accordingly, we analyze the noise growth resulting from coefficient extraction followed by multiplication by a ring element of bounded norm in the following corollary.

Corollary 3. *Adopt the settings in Theorem 3. Let $b \in R$ be a B-bounded ring element, i.e., $\|b\|_\infty \le B$. For each $i \in [n]$, let $\mathsf{ct}_i \in \mathsf{RLWE}_{\mathtt{sk}}(m_i)$ be the ciphertext obtained by coefficient extraction (Algorithm 4), and set $\mathsf{ct}_i' = b \cdot \mathsf{ct}_i$. Additionally, assume heuristically that the coefficients of $\mathtt{Err}(\mathsf{ct}_i)$ are pairwise independent for every $i \in [n]$. Then, for all $i, j \in [n]$, the j-th coefficient of $\mathtt{Err}(\mathsf{ct}_i')$ satisfies*

$$\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_i')) \le B^2 \cdot \left(\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + n(n-1) \cdot V_{\mathtt{auto}} + n^2 \cdot V_{\mathtt{ms}}\right).$$

Proof. By Theorem 3, Lemma 3, and the independence heuristic, we have

$$\begin{aligned}
&\mathsf{Var}_j(\mathtt{Err}(\mathsf{ct}_i')) \le B^2 \cdot \sum_{\ell \in [n]} \mathsf{Var}_\ell(\mathtt{Err}(\mathsf{ct}_i)) \\
&= B^2 \cdot \left(\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + \frac{1}{3}(n^2-1) \cdot V_{\mathtt{auto}} + n^2 \cdot V_{\mathtt{ms}} + \sum_{\ell=1}^{n-1} \left(\frac{n}{2^{\nu_2(\ell)}} - 1\right) \cdot V_{\mathtt{auto}}\right) \\
&= B^2 \cdot \left(\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + n(n-1) \cdot V_{\mathtt{auto}} + n^2 \cdot V_{\mathtt{ms}}\right)
\end{aligned}$$

where the last equality follows from the fact that

$$\begin{aligned}
\sum_{\ell=1}^{n-1} \left(\frac{n}{2^{\nu_2(\ell)}} - 1\right) &= \sum_{k=0}^{\log n - 1} \sum_{\substack{1 \le j \le n-1 \\ \nu_2(j) = k}} \frac{n}{2^k} - (n-1) = \sum_{k=0}^{\log n - 1} \frac{n}{2^k} \cdot 2^{\log n - k - 1} - (n-1) \\
&= n \sum_{k=0}^{\log n - 1} 2^{\log n - 1 - 2k} - (n-1) = \frac{(2n-1)(n-1)}{3}.
\end{aligned}$$

□

Asymptotic Improvements. Theorem 3 shows that, in each output ciphertext ct_i, input noise contributes only to the constant coefficient, which inherits the i-th coefficient of $\mathtt{Err}(\mathsf{ct})$. For $j > 0$, the j-th noise coefficient has variance $\left(n/2^{\nu_2(j)} - 1\right) V_{\mathtt{auto}}$ and contains only noise from automorphism evaluations. By contrast, prior recurrence-based analyses [MW22, BMW24] use a coefficient-independent multiplier of $\Theta(n^2)$ for the $V_{\mathtt{auto}}$ contribution. The ratio of this prior multiplier to $n/2^{\nu_2(j)} - 1$ therefore ranges from $\Theta(n)$ to $\Theta(n^2)$ across the non-constant coefficients. For fixed B and $V_{\mathtt{auto}}, V_{\mathtt{ms}} \in \Theta(n)$, Corollary 3 shows that our post-extraction multiplication bound scales as $\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + \Theta(n^3)$. For a fair comparison under the same pairwise independence heuristic for the output noise coefficients, we retain the exact input-noise contribution and apply the prior uniform estimate only to operation noise. The bound derived from the prior estimate then scales as $\mathsf{Var}_i(\mathtt{Err}(\mathsf{ct})) + \Theta(n^4)$, and its ratio to our bound is $O(n)$.

6 Experimental Validation

In Sections 4 and 5, we proposed new analyses for trace evaluation and two typical algorithmic applications by explicitly tracking the effects of algebraic operators. First, we give a brief summary of the potential implications of our analyses for practical use cases:

- *Trace Evaluation:* By Theorem 1, the constant-term variance bound matches that of [LY26], whereas all other coefficients receive no input-noise contribution and depend on the 2-adic valuations of their indices, yielding bounds tighter than the uniform estimate by a factor of $O(\log n)$. Under the pairwise independence heuristic for the output noise coefficients, Corollary 2 further shows that our post-multiplication bound is tighter than that derived from the uniform estimate by a factor of $O(\log n)$.
- *Ciphertext Packing:* By Theorem 2, our coefficient-wise analysis gives a slightly smaller variance estimate than that of [LY26] due to a more precise analysis of the noise introduced by modulus switching. For typical parameter sets where the RLWE secret key has a small norm, $V_{\texttt{auto}}$ dominates $V_{\texttt{ms}}$, so the variance difference may be small in practice. Nevertheless, our non-recursive variant offers a cleaner, more implementation-friendly formulation that may facilitate further system-level optimizations.
- *Coefficient Extraction:* By Theorem 3, the noise variance of the constant term in each output is comparable to prior uniform estimates [MW22, BMW24]. However, our analysis shows that the variances of all other coefficients depend on the 2-adic valuations of their indices, yielding coefficient-wise bounds that are tighter by factors ranging from $\Theta(n)$ to $\Theta(n^2)$. Under the pairwise independence heuristic for the output noise coefficients, the comparison following Corollary 3 shows that the ratio of the corresponding uniform estimate bound to ours is $O(n)$.

Therefore, we expect our new analyses to provide more accurate noise estimates, which can lead to better parameter selection or noise budget management in practical implementations. To this end, we conduct experiments to validate the asymptotically tighter noise bounds derived for trace evaluation and coefficient extraction.

Experimental Setup. We implement the main algorithms described in this work (Algorithms 2, 3, and 4) in C++[3] using the FFTW3 library [FJ05]. Based on the preceding analysis, our empirical comparisons focus on trace evaluation (Algorithm 2) and coefficient extraction (Algorithm 4), where the predicted improvements are visible. All experiments are conducted on a GitHub Codespaces instance with 4 CPU cores and 16 GB of RAM, backed by an Intel(R) Xeon(R) Platinum 8370C processor running at 2.80 GHz. The instance runs Ubuntu 24.04.4 LTS[4], and all code is compiled with GCC 13.3.0.

6.1 Validation of Trace Evaluation

We first validate our new noise analysis for the LY trace evaluation method (Algorithm 2) and compare it with the original analysis from [LY26].

Parameters. We set the ciphertext modulus $q = 2^{32}$, the ring degree $n = 2^{11}$, use a uniform ternary distribution for the secret key, and adopt a discrete Gaussian noise distribution with standard deviation 3.2 for both the input ciphertext and the automorphism keys. The gadget decomposition base for automorphism evaluation is set to $\beta = 2^4$, and the coefficients of the plaintext for post-trace multiplication are sampled uniformly from $[-\beta, \beta]$. Here, we choose a relatively small β both to limit the numerical error introduced by the FFT and to mitigate the looseness of the worst-case bound in Lemma 3 and in all subsequent analyses that rely on it.

[3]The source code is available at https://github.com/hxialab/ErrTrace.
[4]For ease of reproduction, the code also runs on other platforms, e.g., macOS on Apple silicon.

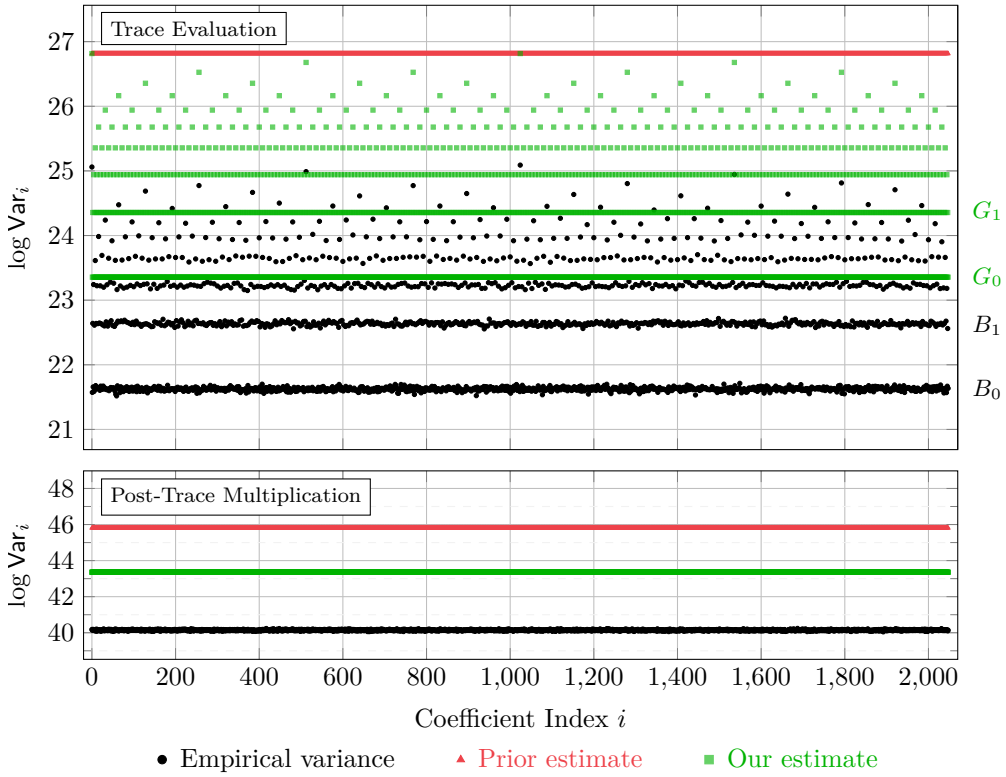

Figure 3: Comparison of the empirical and estimated variances for each output noise coefficient in trace evaluation (Algorithm 2) and post-trace multiplication. All the estimated values are obtained from the analysis in Section 4.

Experimental Results. We executed the LY trace evaluation 5,000 times and computed the empirical variance for each coefficient. The results are reported in Figure 3, where we plot our estimates together with the uniform coefficient estimate of [LY26]. As can be observed, our estimates match the experimental data closely for all coefficients. This is reflected by the fact that our estimated bounds (the green points) closely track the empirical variances (the black points) and appear as an almost uniform upward shift across indices. More specifically, we note the following observations:

- The line G_0 is approximately a 2-bit vertical upward shift of the region B_0, which corresponds to indices with 2-adic valuation 0 (i.e., the odd indices);
- Likewise, the line G_1 is nearly a 2-bit upward shift of B_1, corresponding to indices with 2-adic valuation 1. This pattern extends to indices with other 2-adic valuations.

In contrast, the previous uniform bound exceeds the empirical values by $4-6$ bits for coefficients whose indices have 2-adic valuation 0 or 1, which together account for 75% of all coefficient positions. For the post-trace multiplication, the small initial noise used in our experiment makes the operation noise introduced during trace evaluation the dominant contribution to the post-multiplication variance. In this regime, our post-multiplication bound is about 3 bits tighter than the comparison bound derived from the prior uniform estimate and about 3 bits above the empirical variance, placing it nearly midway between the two. Taken together, these comparisons highlight the loss caused by discarding coefficient-wise information about operation noise. However, our experiments do not cover cases where initial noise dominates the post-multiplication variance; the same contribution will then dominate both bounds, and the improvement will vanish.

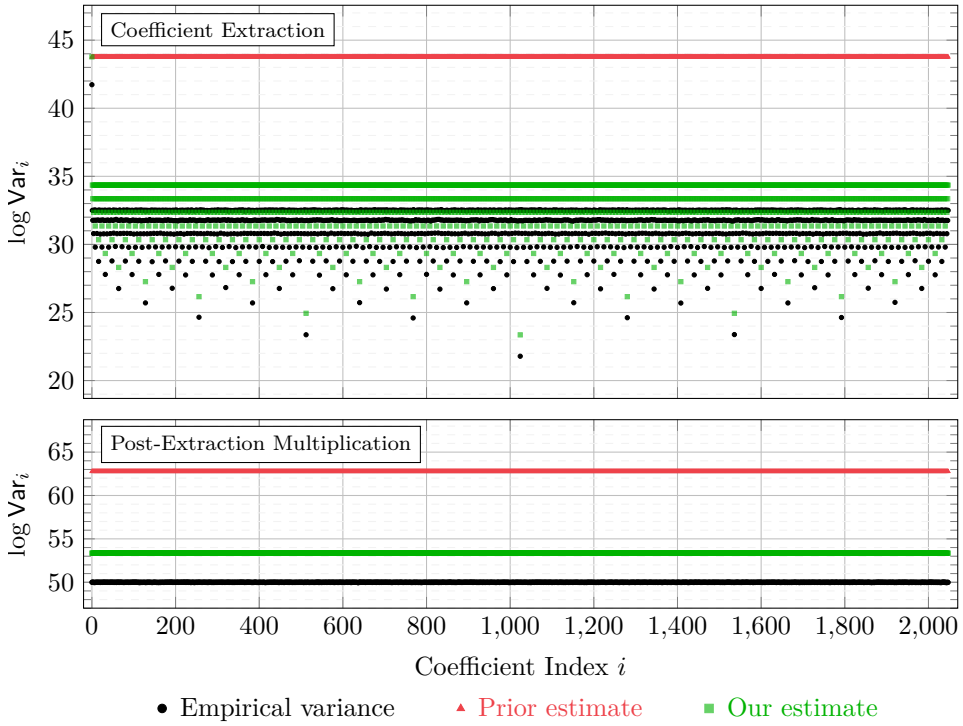

Figure 4: Comparison of the empirical and estimated variances for each output noise coefficient in coefficient extraction (Algorithm 4) and post-extraction multiplication. All the estimated values are obtained from the analysis in Section 5.2.

6.2 Validation of Coefficient Extraction

Next, we validate our new noise analysis for the coefficient extraction method (Algorithm 4) and compare it with the previous analyses from [MW22, BMW24]. We use the same parameters as in the experiments for trace evaluation. Unlike the NTT-based setting of the prior work, our FFT-based implementation introduces modulus-switching noise. In our typical parameter setting, the ternary RLWE secret key keeps $V_{\mathtt{ms}}$ small relative to $V_{\mathtt{auto}}$, so this additional contribution does not materially affect the conclusions. Namely, the same coefficient-wise pattern holds for NTT-based implementations.

Experimental Results. Since the input noise coefficients have common variance in our setup, Theorem 3 gives the same variance bound for each fixed coefficient index across all output ciphertexts. We therefore ran the algorithm 20 times and pooled the corresponding noise coefficients from the resulting 40,960 ciphertexts to compute empirical variances. As shown in Figure 4, the coefficient-wise variance profile is approximately the reverse of that observed for trace evaluation. For non-constant coefficients, this reversal arises because the extraction variance is proportional to $n/2^{\nu_2(i)} - 1$ and therefore decreases roughly geometrically with the 2-adic valuation (see Theorem 3), whereas the variance in trace evaluation grows affinely with the valuation (see Theorem 1).

For coefficient extraction, our variance bounds for non-constant coefficients are at least 8 bits tighter than the prior bounds, while exceeding the empirical values by only about 2 bits. For post-extraction multiplication, our bounds are around 9 bits tighter than the bound derived from the prior uniform estimate and remain only 3−4 bits above the observed variances. These results demonstrate that coefficient-wise noise tracking yields bounds that remain close to the empirical variances and are markedly tighter.

7 Concluding Remarks

Extensions. Our improvements arise from an algebraic, coefficient-wise analysis that preserves structure lost under uniform bounds. This analysis is not tied to the specific settings considered here and extends directly in the following directions.

- *Worst-Case Analysis.* The same coefficient-wise algebraic tracking remains applicable when worst-case noise bounds are used throughout in place of the average-case variance analysis. It can therefore likewise yield tighter bounds than those obtained by uniformly bounding all coefficients.
- *Intermediate Subfields.* This work focuses on full packing/extraction, where the trace is taken from K to $\mathbb{Q}$. More generally, the same coefficient-wise insights apply when the relative trace targets any intermediate cyclotomic subfield of K.
- *NTT Backend.* For NTT-based implementations with $\gcd(n, q) = 1$, each modulus-switching step followed by modulus raising can be replaced by multiplication by the appropriate inverse modulo q. Specifically, the per-level steps use $2^{-1} \in \mathbb{Z}_q^*$, whereas the one-shot preprocessing step uses $n^{-1} \in \mathbb{Z}_q^*$. These operations are exact and introduce no modulus-switching noise; hence, the same analyses apply with $V_{\mathtt{ms}} = 0$.

Limitations. The current estimates rely on the independence heuristic, which is not exact in general. In addition to dependencies induced by the deterministic structure of the algorithms, some noise terms share underlying random variables, although they are treated as independent in our variance estimates. In particular:

- The coefficients of each noise term introduced by an automorphism evaluation are heuristically treated as independent in the analysis of coefficient extraction (Theorem 3).
- The coefficients of the output noise from trace evaluation (Algorithm 2) and those from coefficient extraction (Algorithm 4) are heuristically treated as independent in the subsequent plaintext-multiplication analyses (Corollaries 2 and 3).

In both cases, the coefficients are derived, at least in part, from negacyclic convolutions involving the same underlying noise coefficients in the automorphism keys, and their dependence may therefore require a more refined probabilistic treatment. Nevertheless, for the parameter sets considered in our experiments, the proposed estimates remain above the empirical variances by a visible margin. For the post-multiplication results, however, this gap should not be interpreted as a guaranteed safety margin, since the analysis uses a worst-case bound for the plaintext, whereas the experiments use uniformly sampled bounded plaintexts and thus reflect typical rather than worst-case behavior.

Potential Refinements. Two main sources of discrepancy between the current estimates and the actual variances are the independence heuristic and the worst-case bounds for multiplication in Lemma 3. The latter are used, in particular, to estimate the variance $V_{\mathtt{auto}}$ of the noise introduced by automorphism evaluation in Lemma 6. The following directions may therefore further improve the accuracy of the estimates:

- The reliance on the independence heuristic may be reduced by tracking, within the present average-case framework, which contributions to the final noise depend on the same noise coefficients sampled during automorphism key generation and retaining the resulting covariance terms. This refinement could provide more accurate variance estimates at the cost of a more involved analysis.
- The worst-case bounds in Lemma 3 may be refined by more accurately characterizing the distribution of the gadget decomposition digits, for example by modeling them according to an appropriate discrete distribution. Alternatively, one may employ randomized gadget decomposition techniques [MP12, AP14, GMP19], for which the coefficients

after gadget decomposition can be modeled as subgaussian random variables, at the cost of requiring additional operations.

More broadly, our coefficient-wise algebraic framework is orthogonal to the particular probabilistic models and auxiliary bounds adopted here and can therefore accommodate future advances in noise analysis, potentially yielding more accurate estimates.

Acknowledgments

The author would like to thank the anonymous reviewers of IACR Communications in Cryptology (CiC) for their valuable feedback and suggestions, which helped improve the presentation and clarity of this paper. This work was supported by the XJTLU Research Development Fund under Grant RDF-25-02-040.

References

[ABBB+22] Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. OpenFHE: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC'22, pages 53–63, New York, NY, USA, 2022. ACM. doi:10.1145/3560827.3563379.

[ACJ24] Rashmi Agrawal, Anantha Chandrakasan, and Ajay Joshi. HEAP: A fully homomorphic encryption accelerator with parallelized bootstrapping. In *51st ACM/IEEE Annual International Symposium on Computer Architecture (ISCA 2024)*, pages 756–769, Buenos Aires, Argentina, June 29 – July 3, 2024. IEEE. doi:10.1109/ISCA59077.2024.00060.

[ACLS18] Sebastian Angel, Hao Chen, Kim Laine, and Srinath T. V. Setty. PIR with compressed queries and amortized query processing. In *2018 IEEE Symposium on Security and Privacy*, pages 962–979, San Francisco, CA, USA, May 21–23, 2018. IEEE Computer Society Press. doi:10.1109/SP.2018.00062.

[ADDG24] Martin R. Albrecht, Alex Davidson, Amit Deo, and Daniel Gardham. Crypto dark matter on the torus - oblivious PRFs from shallow PRFs and TFHE. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 447–476, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58751-1_16.

[ALP+21] Asra Ali, Tancrède Lepoint, Sarvar Patel, Mariana Raykova, Phillipp Schoppmann, Karn Seth, and Kevin Yeo. Communication-computation trade-offs in PIR. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021: 30th USENIX Security Symposium*, pages 1811–1828. USENIX Association, August 11–13, 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/ali.

[AP13] Jacob Alperin-Sheriff and Chris Peikert. Practical bootstrapping in quasilinear time. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*,

pages 1–20, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-40041-4_1.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 297–314, Santa Barbara, CA, USA, August 17–21, 2014. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-44371-2_17.

[BDF18] Guillaume Bonnoron, Léo Ducas, and Max Fillinger. Large FHE gates from tensored homomorphic accumulator. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 2018: 10th International Conference on Cryptology in Africa*, volume 10831 of *Lecture Notes in Computer Science*, pages 217–251, Marrakesh, Morocco, May 7–9, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-89339-6_13.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 309–325, Cambridge, MA, USA, January 8–10, 2012. Association for Computing Machinery. doi:10.1145/2090236.2090262.

[BI26] Jean-Philippe Bossuat and Malika Izabachène. Large domain homomorphic evaluation for BFV-like schemes via ring repacking. *Designs, Codes and Cryptography*, 94(2):46, 2026. doi:10.1007/s10623-025-01782-x.

[BJ25] Olivier Bernard and Marc Joye. Bootstrapping (T)FHE ciphertexts via automorphisms: Closing the gap between binary and Gaussian keys. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part VII*, volume 16251 of *Lecture Notes in Computer Science*, pages 3–35, Melbourne, VIC, Australia, December 8–12, 2025. Springer, Singapore, Singapore. doi:10.1007/978-981-95-5122-4_1.

[BMTH21] Jean-Philippe Bossuat, Christian Mouchet, Juan Ramón Troncoso-Pastoriza, and Jean-Pierre Hubaux. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 587–617, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-77870-5_21.

[BMW24] Alexander Burton, Samir Jordan Menon, and David J. Wu. Respire: High-rate PIR for databases with small records. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 1463–1477, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press. doi:10.1145/3658644.3690328.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, Palm Springs, CA, USA, October 22–25, 2011. IEEE Computer Society Press. doi:10.1109/FOCS.2011.12.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes*

in Computer Science, pages 505–524, Santa Barbara, CA, USA, August 14–18, 2011. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-22792-9_29.

[BZS+25] Song Bian, Zian Zhao, Ruiyu Shen, Zhou Zhang, Ran Mao, Dawei Li, Yizhong Liu, Masaki Waga, Kohei Suenaga, Zhenyu Guan, Jiafeng Hua, Yier Jin, and Jianwei Liu. CHLOE: Loop transformation over fully homomorphic encryption via multi-level vectorization and control-path reduction. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *2025 IEEE Symposium on Security and Privacy*, pages 2395–2413, San Francisco, CA, USA, May 12–15, 2025. IEEE Computer Society Press. doi:10.1109/SP61157.2025.00035.

[BZZ+24] Song Bian, Zian Zhao, Zhou Zhang, Ran Mao, Kohei Suenaga, Yier Jin, Zhenyu Guan, and Jianwei Liu. HEIR: A unified representation for cross-scheme compilation of fully homomorphic computation. In *ISOC Network and Distributed System Security Symposium – NDSS 2024*, San Diego, CA, USA, February 26 – March 1, 2024. The Internet Society. doi:10.14722/ndss.2024.23067.

[CCR19] Hao Chen, Ilaria Chillotti, and Ling Ren. Onion ring ORAM: Efficient constant bandwidth oblivious RAM from (leveled) TFHE. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 345–360, London, UK, November 11–15, 2019. ACM Press. doi:10.1145/3319535.3354226.

[CDKS21] Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. Efficient homomorphic conversion between (ring) LWE ciphertexts. In Kazue Sako and Nils Ole Tippenhauer, editors, *ACNS 2021: 19th International Conference on Applied Cryptography and Network Security, Part I*, volume 12726 of *Lecture Notes in Computer Science*, pages 460–479, Kamakura, Japan, June 21–24, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-78372-3_18.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33, Hanoi, Vietnam, December 4–8, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53887-6_1.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, January 2020. doi:10.1007/s00145-019-09319-x.

[CGL+25] Zhaohui Chen, Zhen Gu, Yanheng Lu, Xuanle Ren, Ruiguang Zhong, Wen-jie Lu, Jiansong Zhang, Yichi Zhang, Hanghang Wu, Xiaofu Zheng, Heng Liu, Tingqiang Chu, Cheng Hong, Changzheng Wei, Dimin Niu, and Yuan Xie. SAFE: A scalable homomorphic encryption accelerator for vertical federated learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 44(5):1662–1675, May 2025. doi:10.1109/TCAD.2024.3496836.

[CHK+18] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology –*

EUROCRYPT 2018, Part I, volume 10820 of *Lecture Notes in Computer Science*, pages 360–384, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78381-9_14.

[CIV18] Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Homomorphic SIM^2D operations: Single instruction much more data. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 338–359, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78381-9_13.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-70694-8_15.

[CKL26] Philippe Chartier, Michel Koskas, and Mohammed Lemou. Exploring general cyclotomic rings in torus-based fully homomorphic encryption. *IACR Communications in Cryptology*, 3(1), 2026. doi:10.62056/ab0ljb0kr.

[CLOT21] Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part III*, volume 13092 of *Lecture Notes in Computer Science*, pages 670–699, Singapore, December 6–10, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-92078-4_23.

[CR25] Yue Chen and Ling Ren. OnionPIRv2: Efficient single-server PIR. Cryptology ePrint Archive, Report 2025/1142, 2025. URL: https://eprint.iacr.org/2025/1142.

[DFH+24] Xianglong Deng, Shengyu Fan, Zhicheng Hu, Zhuoyu Tian, Zihao Yang, Jiangrui Yu, Dingyuan Cao, Dan Meng, Rui Hou, Meng Li, Qian Lou, and Mingzhe Zhang. Trinity: A general purpose FHE accelerator. In *2024 57th IEEE/ACM International Symposium on Microarchitecture*, pages 338–351, 2024. doi:10.1109/MICRO61859.2024.00033.

[DKMS24] Gabrielle De Micheli, Duhyeong Kim, Daniele Micciancio, and Adam Suhl. Faster amortized FHEW bootstrapping using ring automorphisms. In Qiang Tang and Vanessa Teague, editors, *PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part IV*, volume 14604 of *Lecture Notes in Computer Science*, pages 322–353, Sydney, NSW, Australia, April 15–17, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-57728-4_11.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640, Sofia, Bulgaria, April 26–30, 2015. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-46800-5_24.

[DXQ+26] Yiran Dai, Binwu Xiang, Lang Qin, Yi Deng, and Jiang Zhang. FlashPIR: Low-latency FHE-based single-server PIR with low client overhead. *Cybersecurity*, 9:171, 2026. doi:10.1186/s42400-026-00601-7.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. doi:10.1109/JPROC.2004.840301.

[FV12] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. URL: https://eprint.iacr.org/2012/144.

[Gee24] Robin Geelen. Revisiting the slot-to-coefficient transformation for BGV and BFV. *IACR Communications in Cryptology*, 1(3):37, 2024. doi:10.62056/a01zogy4e-.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press. doi:10.1145/1536414.1536440.

[GH19] Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 438–464, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-36033-7_17.

[GHPS13] Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Field switching in BGV-style homomorphic encryption. *Journal of Computer Security*, 21(5):663–684, 2013. doi:10.3233/JCS-130480.

[GHS12a] Craig Gentry, Shai Halevi, and Nigel P. Smart. Better bootstrapping in fully homomorphic encryption. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 1–16, Darmstadt, Germany, May 21–23, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-30057-8_1.

[GHS12b] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_28.

[GMP19] Nicholas Genise, Daniele Micciancio, and Yuriy Polyakov. Building an efficient lattice gadget toolkit: Subgaussian sampling and more. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 655–684, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-17656-3_23.

[GP25] Antonio Guimarães and Hilder V. L. Pereira. Fast amortized bootstrapping with small keys and polynomial noise overhead. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025: 32nd Conference on Computer and Communications Security*, pages 2967–2981, Taipei, Taiwan, October 13–17, 2025. ACM Press. doi:10.1145/3719027.3765181.

[GPvL23] Antonio Guimarães, Hilder V. L. Pereira, and Barry van Leeuwen. Amortized bootstrapping revisited: Simpler, asymptotically-faster, implemented. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 3–35, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8736-8_1.

[GV26] Robin Geelen and Frederik Vercauteren. Better GBFV bootstrapping and faster encrypted edit distance computation. *IACR Communications in Cryptology*, 2(4):35, 2026. doi:10.62056/a3n56chdj.

[HMS25] Intak Hwang, Seonhong Min, and Yongsoo Song. Carousel: Fully homomorphic encryption with bootstrapping over automorphism group. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part VII*, volume 16251 of *Lecture Notes in Computer Science*, pages 131–164, Melbourne, VIC, Australia, December 8–12, 2025. Springer, Singapore, Singapore. doi:10.1007/978-981-95-5122-4_5.

[HS14] Shai Halevi and Victor Shoup. Algorithms in HElib. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 554–571, Santa Barbara, CA, USA, August 17–21, 2014. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-44371-2_31.

[HS15] Shai Halevi and Victor Shoup. Bootstrapping for HElib. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 641–670, Sofia, Bulgaria, April 26–30, 2015. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-46800-5_25.

[HS18] Shai Halevi and Victor Shoup. Faster homomorphic linear transformations in HElib. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 93–120, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-96884-1_4.

[HYT+24] Jiaxing He, Kang Yang, Guofeng Tang, Zhangjie Huang, Li Lin, Changzheng Wei, Ying Yan, and Wei Wang. Rhombus: Fast homomorphic matrix-vector multiplication for secure two-party inference. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 2490–2504, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press. doi:10.1145/3658644.3690281.

[HZD+26] Chenyu Huang, Fan Zhang, Minxin Du, Sherman S. M. Chow, Huangxun Chen, Huaming Rao, Danqing Huang, Bo Qian, and Peng Chen. Practical anonymous two-party gradient boosting decision tree. In *2026 IEEE Symposium on Security and Privacy (SP)*, pages 2023–2041, 2026. doi:10.1109/SP63933.2026.00084.

[IB25] Malika Izabachène and Jean-Philippe Bossuat. TETRIS: Composing FHE techniques for private functional exploration over large datasets. *Proceedings on Privacy Enhancing Technologies*, 2025(2):23–38, April 2025. doi:10.56553/popets-2025-0047.

[IY21] Yu Ishimaki and Hayato Yamana. Faster homomorphic trace-type function evaluation. *IEEE Access*, 9:53061–53077, 2021. doi:10.1109/ACCESS.2021.3071264.

[KDE+24] Andrey Kim, Maxim Deryabin, Jieun Eom, Rakyong Choi, Yongwoo Lee, Whan Ghang, and Donghoon Yoo. General bootstrapping approach for RLWE-based homomorphic encryption. *IEEE Transactions on Computers*, 73(1):86–96, 2024. doi:10.1109/TC.2023.3318405.

[KLD+23] Andrey Kim, Yongwoo Lee, Maxim Deryabin, Jieun Eom, and Rakyong Choi. LFHE: Fully homomorphic encryption with bootstrapping key size less than a megabyte. Cryptology ePrint Archive, Report 2023/767, 2023. URL: https://eprint.iacr.org/2023/767.

[KS26] Jiayi Kang and Leonard Schild. Pirouette: Query efficient single-server PIR. *Proceedings on Privacy Enhancing Technologies*, 2026(2):215–241, 2026. doi:10.56553/popets-2026-0045.

[Lan90] Serge Lang. *Cyclotomic Fields I and II*, volume 121 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2 edition, 1990. Combined second edition. doi:10.1007/978-1-4612-0987-4.

[LHG+25] Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and Wenguang Chen. BumbleBee: Secure two-party inference framework for large transformers. In *ISOC Network and Distributed System Security Symposium – NDSS 2025*, San Diego, CA, USA, February 24–28, 2025. The Internet Society. doi:10.14722/ndss.2025.230057.

[LHZ+23] Wen-jie Lu, Zhicong Huang, Qizhi Zhang, Yuchen Wang, and Cheng Hong. Squirrel: A scalable secure two-party computation framework for training gradient boosting decision tree. In Joseph A. Calandrino and Carmela Troncoso, editors, *USENIX Security 2023: 32nd USENIX Security Symposium*, pages 6435–6451, Anaheim, CA, USA, August 9–11, 2023. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/lu.

[LLT+26] Haofei Liang, Zeyu Liu, Eran Tromer, Xiang Xie, and Yu Yu. InstantOMR: Oblivious message retrieval with low latency and optimal parallelizability. In *USENIX Security 2026: 35th USENIX Security Symposium*, Baltimore, MD, USA, August 12–14, 2026. USENIX Association. To appear. URL: https://www.usenix.org/conference/usenixsecurity26/presentation/liang.

[LLW+24] Zhihao Li, Xianhui Lu, Zhiwei Wang, Ruida Wang, Ying Liu, Yinhang Zheng, Lutan Zhao, Kunpeng Wang, and Rui Hou. Faster NTRU-based bootstrapping in less than 4 ms. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(3):418–451, 2024. doi:10.46586/tches.v2024.i3.418-451.

[LMK+23] Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo. Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 227–256, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30620-4_8.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-13190-5_1.

[LSL+25] Zhihao Li, Xuan Shen, Xianhui Lu, Ruida Wang, Yuan Zhao, Zhiwei Wang, and Benqiang Wei. Leveled functional bootstrapping via external product tree. Cryptology ePrint Archive, Paper 2025/022, 2025. URL: https://eprint.iacr.org/2025/022.

[LW23a] Feng-Hao Liu and Han Wang. Batch bootstrapping I: A new framework for SIMD bootstrapping in polynomial modulus. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 321–352, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30620-4_11.

[LW23b] Feng-Hao Liu and Han Wang. Batch bootstrapping II: Bootstrapping in polynomial modulus only requires $\tilde{O}(1)$ FHE multiplications in amortization. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 353–384, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30620-4_12.

[LWWC25] Yuliang Lin, Baosheng Wang, Yi Wang, and Rongmao Chen. NPIR: High-rate PIR for databases with moderate-size records. Cryptology ePrint Archive, Paper 2025/2257, 2025. URL: https://eprint.iacr.org/2025/2257.

[LWZ+26] Zhihao Li, Hongyu Wang, Yuan Zhao, Lichun Li, Zhiwei Wang, Jiaxing He, Changzheng Wei, Ying Yan, and Lifeng Guo. BatchBoot: Fast batched bootstrapping for TFHE scheme and practical applications. In *USENIX Security 2026: 35th USENIX Security Symposium*, Baltimore, MD, USA, August 12–14, 2026. USENIX Association. To appear. URL: https://www.usenix.org/conference/usenixsecurity26/presentation/li-zhihao.

[LY26] Kang Hoon Lee and Ji Won Yoon. Homomorphic field trace revisited: Breaking the cubic noise barrier. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2026(1):105–134, 2026. doi:10.46586/tches.v2026.i1.105-134.

[MCR21] Muhammad Haris Mughees, Hao Chen, and Ling Ren. OnionPIR: Response efficient single-server PIR. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021: 28th Conference on Computer and Communications Security*, pages 2292–2306, Virtual Event, Republic of Korea, November 15–19, 2021. ACM Press. doi:10.1145/3460120.3485381.

[MHWW24] Shihe Ma, Tairong Huang, Anyu Wang, and Xiaoyun Wang. Faster BGV bootstrapping for power-of-two cyclotomics through homomorphic NTT. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part I*, volume 15484 of *Lecture Notes in Computer Science*, pages 143–175, Kolkata, India, December 9–13, 2024. Springer, Singapore, Singapore. doi:10.1007/978-981-96-0875-1_5.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors,

Advances in Cryptology – EUROCRYPT 2012, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_41.

[MP21] Daniele Micciancio and Yuriy Polyakov. Bootstrapping in FHEW-like cryptosystems. In *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC'21, pages 17–28, New York, NY, USA, 2021. ACM. doi:10.1145/3474366.3486924.

[MPSY26] Rasoul Akhavan Mahdavi, Sarvar Patel, Joon Young Seo, and Kevin Yeo. InsPIRe: Communication-efficient PIR with server-side preprocessing. In *2026 IEEE Symposium on Security and Privacy*, pages 1463–1482, 2026. doi:10.1109/SP63933.2026.00061.

[MW22] Samir Jordan Menon and David J. Wu. SPIRAL: Fast, high-rate single-server PIR via FHE composition. In *2022 IEEE Symposium on Security and Privacy*, pages 930–947, San Francisco, CA, USA, May 22–26, 2022. IEEE Computer Society Press. doi:10.1109/SP46214.2022.9833700.

[MW24] Samir Jordan Menon and David J. Wu. YPIR: High-throughput single-server PIR with silent preprocessing. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024: 33rd USENIX Security Symposium*, pages 5985–6002, Philadelphia, PA, USA, August 14–16, 2024. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/menon.

[OPP23] Hiroki Okada, Rachel Player, and Simon Pohmann. Homomorphic polynomial evaluation using Galois structure and applications to BFV bootstrapping. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 69–100, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8736-8_3.

[OPP25] Hiroki Okada, Rachel Player, and Simon Pohmann. Fheanor: a new, modular FHE library for designing and optimising schemes. Cryptology ePrint Archive, Report 2025/864, 2025. URL: https://eprint.iacr.org/2025/864.

[PMH+25] Thales B. Paiva, Gabrielle De Micheli, Syed Mahbub Hafiz, Marcos A. Simplicio Jr., and Bahattin Yildiz. Faster amortized bootstrapping using the incomplete NTT for free. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(4):520–543, 2025. doi:10.46586/tches.v2025.i4.520-543.

[RCG+23] Xuanle Ren, Zhaohui Chen, Zhen Gu, Yanheng Lu, Ruiguang Zhong, Wenjie Lu, Jiansong Zhang, Yichi Zhang, Hanghang Wu, Xiaofu Zheng, Heng Liu, Tingqiang Chu, Cheng Hong, Changzheng Wei, Dimin Niu, and Yuan Xie. CHAM: A customized homomorphic encryption accelerator for fast matrix-vector product. In *60th ACM/IEEE Design Automation Conference (DAC 2023)*, pages 1–6, San Francisco, CA, USA, July 9–13, 2023. IEEE. doi:10.1109/DAC56929.2023.10247696.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MD, USA, May 22–24, 2005. ACM Press. doi:10.1145/1060590.1060603.

[RSG+22] Xuanle Ren, Le Su, Zhen Gu, Sheng Wang, Feifei Li, Yuan Xie, Song Bian, Chao Li, and Fan Zhang. HEDA: multi-attribute unbounded aggregation over homomorphically encrypted database. *Proceedings of the VLDB Endowment*, 16(4):601–614, 2022. doi:10.14778/3574245.3574248.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635, Tokyo, Japan, December 6–10, 2009. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-10366-7_36.

[SV14] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic SIMD operations. *Designs, Codes and Cryptography*, 71(1):57–81, 2014. doi:10.1007/s10623-012-9720-4.

[Was97] Lawrence C. Washington. *Introduction to Cyclotomic Fields*, volume 83 of *Graduate Texts in Mathematics*. Springer, New York, 2 edition, 1997. Second edition. doi:10.1007/978-1-4612-1934-7.

[WHS+25] Ruida Wang, Jincheol Ha, Xuan Shen, Xianhui Lu, Chunling Chen, Kunpeng Wang, and Jooyoung Lee. Refined TFHE leveled homomorphic evaluation and its application. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025: 32nd Conference on Computer and Communications Security*, pages 2309–2323, Taipei, Taiwan, October 13–17, 2025. ACM Press. doi:10.1145/3719027.3744873.

[WLX+25] Han Wang, Ming Luo, Han Xia, Mingsheng Wang, and Hanxu Hou. Accelerating FHEW-like bootstrapping via new configurations of the underlying cryptosystems. Cryptology ePrint Archive, Paper 2025/1711, 2025. URL: https://eprint.iacr.org/2025/1711.

[WWL+24] Ruida Wang, Yundi Wen, Zhihao Li, Xianhui Lu, Benqiang Wei, Kun Liu, and Kunpeng Wang. Circuit bootstrapping: Faster and smaller. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 342–372, Zurich, Switzerland, May 26–30, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-58723-8_12.

[XLW24] Han Xia, Feng-Hao Liu, and Han Wang. More efficient functional bootstrapping for general functions in polynomial modulus. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part IV*, volume 15367 of *Lecture Notes in Computer Science*, pages 130–163, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-78023-3_5.

[XW24a] Han Xia and Mingsheng Wang. LightPIR: Single-server PIR via FHE without Gaussian noise. In Jianying Zhou, Tony Q. S. Quek, Debin Gao, and Alvaro A. Cárdenas, editors, *ASIACCS 2024: 19th ACM Symposium on Information, Computer and Communications Security*, pages 1049–1065, Singapore, July 1–5, 2024. ACM Press. doi:10.1145/3634737.3661140.

[XW24b] Han Xia and Mingsheng Wang. Single-server PIR via NTRU-based FHE: Simpler, smaller, and faster. In *2024 IEEE European Symposium on Security and Privacy*, pages 293–310, Vienna, Austria, July 8–12, 2024. IEEE Computer Society Press. doi:10.1109/EuroSP60621.2024.00023.

[XW25] Han Xia and Mingsheng Wang. Efficient and compact full-domain functional bootstrapping via subring folding. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(4):737–762, 2025. doi:10.46586/tches.v2025.i4.737-762.

[XZD+23] Binwu Xiang, Jiang Zhang, Yi Deng, Yiran Dai, and Dengguo Feng. Fast blind rotation for bootstrapping FHEs. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part IV*, volume 14084 of *Lecture Notes in Computer Science*, pages 3–36, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-38551-3_1.

[YXS+21] Zhaomin Yang, Xiang Xie, Huajie Shen, Shiying Chen, and Jun Zhou. TOTA: Fully homomorphic encryption with smaller parameters and stronger security. Cryptology ePrint Archive, Report 2021/1347, 2021. URL: https://eprint.iacr.org/2021/1347.

[Zam22] Zama. TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data, 2022. https://github.com/zama-ai/tfhe-rs.

[ZLW25] Xiaopeng Zheng, Hongbo Li, and Dingkang Wang. A new framework for fast homomorphic matrix multiplication. *Designs, Codes and Cryptography*, 93(7):2671–2693, 2025. doi:10.1007/S10623-025-01614-Y.

[ZWC25] Qi Zhang, Mingqiang Wang, and Xiaopeng Cheng. Faster FHEW bootstrapping with adaptive key update. In Yongdae Kim, Atsuko Miyaji, and Mehdi Tibouchi, editors, *CANS 2025: 24th International Conference on Cryptology and Network Security*, volume 16351 of *Lecture Notes in Computer Science*, pages 292–312, Osaka, Japan, November 17–20, 2025. Springer, Singapore, Singapore. doi:10.1007/978-981-95-4434-9_14.

A Natural Coefficient Embedding from LWE to RLWE

Here we recall the natural coefficient embedding from LWE to RLWE as used in [CDKS21]. Let $\boldsymbol{c} = (\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ be an LWE ciphertext satisfying $b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e + m \pmod q$, where $\boldsymbol{s} \in \mathbb{Z}_q^n$, $e \in \mathbb{Z}$, and $m \in \mathbb{Z}$ denote the secret key, noise term, and plaintext message, respectively. We can embed it into a RLWE ciphertext in R_q^2 as follows.

Write $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{Z}_q^n$ and $\boldsymbol{s} = (s_0, s_1, \ldots, s_{n-1}) \in \mathbb{Z}_q^n$, and define the following ring elements:

$$\hat{a} = \sum_{i \in [n]} a_i \cdot \zeta_{2n}^{i} \in R_q, \qquad \hat{s} = \sum_{i \in [n]} s_i \cdot \zeta_{2n}^{-i} \in R_q, \qquad \hat{b} = b \in R_q.$$

Then, the constant coefficient of $\hat{a}\hat{s}$ is $\langle \boldsymbol{a}, \boldsymbol{s} \rangle$. Define the ring element

$$M := \hat{b} - \hat{a}\hat{s} - e \in R_q.$$

By construction, its constant coefficient is $b - \langle \boldsymbol{a}, \boldsymbol{s} \rangle - e = m \pmod q$. Therefore, for $\hat{\boldsymbol{c}} := (\hat{a}, \hat{b})$ and $\mathtt{sk} = (-\hat{s}, 1) \in R_q^2$, the phase of $\hat{\boldsymbol{c}}$ under $\mathtt{sk}$ is given by

$$\langle \hat{\boldsymbol{c}}, \mathtt{sk} \rangle = \hat{b} - \hat{a}\hat{s} = M + e \pmod q.$$

Hence, $\hat{\boldsymbol{c}}$ is an RLWE encoding of M: the constant coefficient of its phase is $m + e$, corresponding to an encryption of m with noise e, while the remaining coefficients of M carry no semantic information. Moreover, since

$$\zeta_{2n}^{j} \cdot \hat{\boldsymbol{c}} \in \mathsf{RLWE}_{\mathtt{sk}}(\zeta_{2n}^{j} \cdot M) \quad \text{for all } j \in [n],$$

the message m can equivalently be embedded into any coefficient of the RLWE plaintext.

In the packing algorithm of [CDKS21] and in this work, the packing process is fundamentally based on trace evaluation, which serves to eliminate the unwanted coefficients and enables efficient packing of multiple LWE ciphertexts into a single RLWE ciphertext.

B Supplementary Materials for Section 5

We prove the following fact that was used in the proof of Theorem 3.

Proposition 2. *Let n be a power of two. For $k = 1, \ldots, \log n - 1$, define*

$$\mathcal{S}_k := \left\{ \left[\prod_{t=k}^{\log n - 1} \left(1 + \frac{n}{2^t}\right)^{\epsilon_t} \right]_{2n} \;\middle|\; \epsilon_t \in \{0, 1\} \right\},$$

and set $\mathcal{S}_{\log n} := \{1\}$. Then, for every $k = 1, \ldots, \log n$,

$$\mathcal{S}_k \bmod 2n/2^k := \left\{ [x]_{2n/2^k} \;\middle|\; x \in \mathcal{S}_k \right\} = \mathbb{Z}^*_{2n/2^k}.$$

In particular, the elements of $\mathcal{S}_k$ are pairwise distinct modulo $2n/2^k$, and $|\mathcal{S}_k| = n/2^k$.

Proof. We prove the claim by induction on $\log n - k$. The base case $k = \log n$ is immediate, since $\mathcal{S}_{\log n} = \{1\}$ and $\mathbb{Z}^*_2 = \{1\}$.

Suppose the claim holds for $k + 1$. By separating the factor corresponding to $t = k$, every element of $\mathcal{S}_k$ is obtained from some $s \in \mathcal{S}_{k+1}$ either as

$$s \qquad \text{or} \qquad \left[s \left(1 + \frac{n}{2^k}\right) \right]_{2n}.$$

Since every $s \in \mathcal{S}_{k+1}$ is odd, $(s-1)n/2^k$ is divisible by $2n/2^k$. Moreover, since $2n/2^k \mid 2n$,

$$\left[\left[s \left(1 + \frac{n}{2^k}\right) \right]_{2n} \right]_{2n/2^k} = \left[s \left(1 + \frac{n}{2^k}\right) \right]_{2n/2^k} = \left[s + \frac{n}{2^k} \right]_{2n/2^k}.$$

By the induction hypothesis, as s ranges over $\mathcal{S}_{k+1}$, the residues $[s]_{n/2^k}$ are pairwise distinct and exhaust $\mathbb{Z}^*_{n/2^k}$. For each such s, consider the two residues

$$[s]_{2n/2^k} \qquad \text{and} \qquad \left[s + \frac{n}{2^k} \right]_{2n/2^k}.$$

They are distinct modulo $2n/2^k$ and both reduce to $[s]_{n/2^k}$ modulo $n/2^k$. Since the former modulus is twice the latter, these are the only two such lifts. Moreover, since s is odd and $n/2^k$ is even, both belong to $\mathbb{Z}^*_{2n/2^k}$. Thus, the two residues under consideration form precisely the fiber over $[s]_{n/2^k}$ under the natural reduction map from $\mathbb{Z}^*_{2n/2^k}$ to $\mathbb{Z}^*_{n/2^k}$. As s ranges over $\mathcal{S}_{k+1}$, these fibers are pairwise disjoint and exhaust $\mathbb{Z}^*_{2n/2^k}$. Therefore,

$$\mathcal{S}_k \bmod 2n/2^k = \mathbb{Z}^*_{2n/2^k},$$

and

$$|\mathcal{S}_k| = 2|\mathcal{S}_{k+1}| = 2 \cdot \frac{n}{2^{k+1}} = \frac{n}{2^k}.$$

Hence, the claim holds for k as well, completing the induction. □