High Precision Noise Cleaning for CKKS with Application to Asynchronous Threshold CKKS

Zhihao Li, Cheng Hong, and Tao Wei

Ant Group

Abstract. Noise cleaning in discrete CKKS uses Hermite interpolation to reduce the input error while preserving the encrypted discrete message. Cryptographic applications based on noise flooding require the error after cleaning to be sufficiently small for the added noise to provide statistical hiding without preventing exact recovery. This requirement is more stringent than decryption correctness alone, but the achievable noise reduction is limited by errors introduced during homomorphic evaluation. In particular, key switching and rescaling can dominate the output error even when a higher interpolation order substantially reduces the polynomial residual.

We propose a high precision noise cleaning method for CKKS that preserves encrypted integer messages while attaining a prescribed bound on normalized output noise. We jointly select the cleaning polynomials and evaluation parameters according to a complete error bound that accounts for the input noise and the errors introduced during homomorphic evaluation. By composing Hermite polynomials of lower degree, we can achieve the same output noise bound with a smaller ciphertext modulus and lower evaluation cost than a single higher degree polynomial. In addition, to reduce the ciphertext tensor degree, we construct polynomial representations with noisy auxiliary powers that cancel selected auxiliary error terms. For integer messages, we extract the bits through multi-value functional bootstrapping, then clean and recombine them into a single ciphertext. Finally, we apply high precision noise cleaning to construct an asynchronous threshold CKKS scheme that achieves strong simulation security in the random oracle model.

We implement binary and integer noise cleaning in OpenFHE. For binary messages and a normalized output noise target of 2^{-300}, the selected composition reduces cleaning time from 9.63 to 1.54 seconds compared with a single Hermite evaluation under the same input error bound and security requirements. The composition also requires a smaller input ciphertext modulus and less evaluation key storage. We also evaluate the complete construction on 8 bit integers packed into all 32,768 slots. The maximum measured normalized output noise is below 2^{-300} with either leveled digit extraction or functional bootstrapping.

Keywords: CKKS · Noise cleaning · Hermite interpolation · Polynomial evaluation · Threshold decryption

1 Introduction

Fully homomorphic encryption (FHE) enables arbitrary computation on encrypted data without decryption. The CKKS scheme supports approximate arithmetic on vectors of real or complex numbers, with multiple values packed into a single ciphertext for parallel evaluation [9]. Recent constructions extend this capability to Boolean operations, small integer arithmetic, and lookup table evaluation [2, 5, 6]. Although the underlying arithmetic remains approximate, these discrete CKKS constructions permit exact recovery of the output when the decoding error is sufficiently small.

Cryptographic applications based on noise flooding can require a substantially smaller ciphertext error than exact recovery alone [11]. In threshold CKKS with exact outputs, for example, the added noise must statistically hide the ciphertext error, while the total error after reconstruction must still permit exact recovery. Lagrange interpolation in asynchronous threshold decryption may further amplify the independently sampled flooding noise [14]. Consequently, even computations on small integer messages can require very small normalized noise before decryption. This requirement motivates noise cleaning, which reduces the noise without changing the encrypted discrete message [2, 12].

Hermite interpolation provides a general approach to noise cleaning in discrete CKKS. Alexandru, Kim, and Polyakov (AKP) construct interpolants that agree with the target function at the message values and whose derivatives vanish up to a prescribed order [2]. These derivative conditions determine the order of input noise reduction. Sparse-THI extends trigonometric Hermite interpolation to arbitrary orders with sparse coefficients, reducing both real and imaginary input noise [3]. Hermite noise cleaning also supports high precision CKKS bootstrapping. In particular, Choe et al. combine digit extraction with iterative cleaning and integer recombination [11]. In this construction, successive cleaning steps reduce the digit errors, allowing digit extraction and early evaluations to use less stringent error bounds than those required for the final output.

The noise reduction achieved by these methods depends on both the Hermite residual and the errors introduced during homomorphic evaluation. In particular, key switching and rescaling contribute errors whose magnitudes depend on the evaluation parameters. When these errors dominate the interpolation residual, a higher interpolation order need not reduce the normalized output noise, as shown by the analysis of Sparse-THI [3]. The central problem is therefore to evaluate Hermite cleaning polynomials efficiently while satisfying a prescribed bound on normalized output noise.

1.1 Our Contributions

1) High precision noise cleaning for bits. We first propose a high precision noise cleaning method for bits that attains a prescribed bound on normalized output noise. The method composes Hermite polynomials so that successive cleaning steps reduce the input error together with the errors introduced by earlier evaluations. We derive a recurrence for the complete ciphertext error,

including key switching and rescaling. The recurrence determines admissible intermediate errors and allows us to jointly select the cleaning composition, scaling factors, and key switching parameters. The parameter selection algorithm minimizes the specified computational cost over a finite family of single Hermite evaluations and compositions, subject to security, error, and resource constraints.

2) Low degree Hermite representations and auxiliary error cancellation. To reduce the tensor degree of a cleaning step, we construct polynomial representations with noisy auxiliary powers. We observe that the propagation of auxiliary error depends on the polynomial representation even when the cleaning function remains unchanged. We therefore impose vanishing conditions on the coefficient of the linear auxiliary error at the message values, reducing its contribution to the output error. For the binary Hermite family, the resulting construction attains minimum total degree among polynomial representations with one auxiliary cube and preserves the coefficient denominator. We further characterize the relation between the auxiliary error bound and the attainable order of noise reduction. The analysis gives finite error bounds that include all mixed terms, together with matching error lower bounds in the specified representation spaces.

3) High precision noise cleaning for integer messages. We extend the bit construction to integer messages through digit extraction and recombination, with either leveled extraction or multi-value functional bootstrapping. To combine extraction with cleaning, we jointly evaluate the final conjugation in digit extraction and the first Hermite cleaning polynomial. Since integer recombination multiplies digit errors by their weights, we select the remaining cleaning steps to satisfy the normalized noise bound on the recombined output. We also combine the ciphertext tensors from the last cleaning step for each digit with their integer weights before a single final relinearization. Consequently, the error introduced by this relinearization is not multiplied by the digit weights. We derive a complete bound on the normalized output noise that includes the extraction error, all subsequent evaluation errors, and the recombination weights.

4) Application to asynchronous threshold CKKS. We construct an asynchronous threshold CKKS scheme with exact integer recovery within the threshold FHE framework of Kim et al. (KLLPS) [15]. The decryption procedure rounds the decoded slots and applies exact integer division to recover the output. To satisfy both statistical hiding and exact recovery after Lagrange reconstruction, we derive sufficient conditions on the flooding distribution from the complete cleaning error bound. Under these conditions, uniform error bounds, and the stated encryption and RLWE assumptions, we prove correctness and strong simulation security in the random oracle model with trusted setup and static corruption. To reduce communication, we further give conditions for rescaling before decryption that allow the parties to generate partial decryptions at a smaller modulus with their existing secret key shares.

5) Implementation and evaluation. We implement noise cleaning in OpenFHE [1] for bit and 8 bit integer messages. For bit messages, we compare compositions with single Hermite evaluations at normalized precision targets of 128, 200, and 300 bits under the same input error and security requirements. The selected compositions reduce cleaning time, input ciphertext modulus, and key storage. At the 300 bit target, composition reduces cleaning time from 9.63 to 1.54 seconds. For 8 bit messages, we evaluate the complete integer construction with leveled extraction and functional bootstrapping, each followed by separate cleaning and recombination. Both yield more than 300 bits of measured normalized precision in all 32768 slots. Relative to ordinary evaluation, auxiliary evaluation reduces cleaning key storage by 13.8% and 14.6%, respectively, with less than a 1% increase in total running time.

1.2 Related Work

Noise cleaning in discrete CKKS. Bae et al. construct CKKS bootstrapping procedures for binary circuits and small integer functions [5,6]. BLEACH also reduces accumulated errors in discrete computations through polynomial approximations of step functions [12]. For general function evaluation, AKP [2] uses Hermite interpolation to evaluate functions and reduce noise in the same computation. The polynomial construction supports evaluation without bootstrapping, whereas the trigonometric construction incorporates function evaluation and noise cleaning into functional bootstrapping. Sparse-THI [3] extends trigonometric Hermite interpolation to arbitrary orders with sparse coefficients and reduces both real and imaginary input noise. A different approximation method, due to Bian et al. [7], uses Fourier extensions, with error bounds determined by the smoothness of the extended function.

High precision arithmetic and bootstrapping. META-BTS [4] increases bootstrapping precision by extracting the error from an earlier output and applying an additional bootstrap to that error. In contrast, Choe et al. [11] recover the integer overflow introduced by modulus raising through bit extraction, iterative cleaning, and recombination. By separating tensor multiplication, relinearization, and rescaling, their cleaning procedure reduces modulus consumption.

Homomorphic polynomial evaluation. Lee et al. [16] minimize the error variance in CKKS bootstrapping by accounting for both polynomial approximation error and errors in the polynomial basis. The lazy-BSGS algorithm reduces the latter contribution by delaying relinearization and rescaling. Cheon et al. [8] reduce the modulus consumption of multiplication through a quotient and remainder representation of ciphertexts. The corresponding relinearization and rescaling procedures compensate errors between the high and low components to preserve accuracy after recombination. Our auxiliary representations instead reduce the tensor degree of a fixed Hermite cleaning polynomial while cancelling selected error terms introduced by auxiliary generation.

Threshold CKKS. Mouchet et al. [20] construct multiparty CKKS with additive secret sharing, where decryption requires all parties. The subsequent construction of Mouchet, Bertrand, and Hubaux [19] supports a t-out-of-n access structure by converting Shamir shares into additive shares for a designated set of decryptors. Consequently, each party must know the responding set before generating its partial decryption. Colin de Verdière, Passelègue, and Stehlé [22] identify attacks that exploit responses generated for different sets in existing synchronized constructions and provide a repair based on pseudorandom masks.

For protocols with approximate outputs, the simulation must also account for the noise in the decryption result. Hwang et al. [13] analyze CKKS protocols with independently sampled flooding noise and establish liberal security, in which the simulator additionally receives the exact output. To obtain standard simulation security for the approximate functionality, they instead use distributed noise sampling. In the asynchronous setting, KLLPS [14] provides a Shamir based threshold construction with improved bounds on the reconstruction and simulation coefficients, together with an extension to CKKS. The extended version [15] identifies an attack on strong simulation security using responses to related ciphertexts and adds masks derived from a sharing of zero to prevent it. Our application uses the repaired construction for exact discrete outputs, with the flooding parameters determined by the complete cleaning error bound.

1.3 Concurrent Work

Concurrent work reduces the modulus consumption of high precision CKKS by allowing different error magnitudes across slots. Cheon and Lee introduce Focused CKKS [10], which uses cyclotomic units to reduce the errors introduced by rescaling and key switching in focus slots while permitting larger errors in background slots. Generalized CKKS (GCKKS) [18] obtains a related tradeoff through a polynomial decryption factor, using fewer message slots to support high precision arithmetic with smaller scaling factors.

Discrete noise cleaning also plays an important role in the bootstrapping constructions of these schemes. Focused CKKS cleans fully packed inputs through branches that collectively cover all message slots, whereas GCKKS bootstrapping combines iterative digit cleaning with changes to the decryption factor and ciphertext splitting [17]. These constructions modify the ciphertext representation and the associated arithmetic to reduce modulus consumption. Our construction instead retains ordinary CKKS arithmetic and reduces the cost of noise cleaning through Hermite compositions and representations with noisy auxiliary powers. The resulting evaluations account for intermediate evaluation errors and operate on all message slots without splitting the slots among separate ciphertexts.

2 Preliminaries

2.1 Notation

All logarithms are base 2 unless stated otherwise. Let N be a power of two, $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$, and $\mathcal{R}_Q = \mathcal{R}/Q\mathcal{R}$. We use centered representatives for

elements of $\mathcal{R}_Q$. Vectors are denoted in bold, and functions on vectors are applied componentwise. The notation $\|\cdot\|_\infty$ denotes the maximum absolute value of the entries of a vector or the coefficients of a polynomial. We write $\|\cdot\|_1$ for the sum of their absolute values. The canonical embedding $\sigma : \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{R} \to \mathbb{C}^{N/2}$ retains one coordinate from each conjugate pair, and $\sigma_\ell(a)$ denotes its ℓth coordinate. The canonical embedding norm of a is $\|\sigma(a)\|_\infty$ [2]. For $a \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{R}$, the coefficient and embedding norms satisfy $\|a\|_\infty \le \|\sigma(a)\|_\infty \le \|a\|_1 \le N\|a\|_\infty$. The notation $\lfloor \cdot \rceil$ denotes coefficientwise nearest integer rounding.

2.2 The CKKS Scheme

CKKS supports approximate arithmetic on vectors of real or complex numbers [2, 3]. We use the RNS variant, with ciphertext moduli $Q_\ell = \prod_{i=0}^{\ell} q_i$ for distinct primes $q_0, \ldots, q_L$. A ciphertext at level ℓ belongs to $\mathcal{R}_{Q_\ell}^2$ and has a scaling factor $\Delta > 0$. For a vector $\mathbf{z} \in \mathbb{C}^{N/2}$ and a plaintext polynomial $m \in \mathcal{R}$, encoding and decoding are defined by

$$\mathrm{Encode}_\Delta(\mathbf{z}) = \left\lfloor \Delta \sigma^{-1}(\mathbf{z}) \right\rceil, \qquad \mathrm{Decode}_\Delta(m) = \frac{\sigma(m)}{\Delta}.$$

Under a secret key $s \in \mathcal{R}$, a ciphertext $\mathsf{ct} = (b, a) \in \mathcal{R}_{Q_\ell}^2$ encrypting m satisfies $b + as = m + e \pmod{Q_\ell}$, where e is the encryption error. Decryption computes $b + as$ modulo Q_ℓ and decodes the centered representative at scale Δ to obtain the approximate message vector.

We use the following homomorphic operations on ciphertexts under the same secret key [2].

- **Addition.** For ciphertexts $\mathsf{ct}_i = (b_i, a_i)$ with the same modulus Q_ℓ and scaling factor Δ, homomorphic addition computes $(b_1 + b_2, a_1 + a_2) \pmod{Q_\ell}$. A plaintext u encoded at scale Δ is added by replacing b with $b + u$. Both operations preserve the scaling factor Δ.
- **Integer multiplication.** Multiplication by $k \in \mathbb{Z}$ computes $(kb, ka) \pmod{Q_\ell}$ without changing the scaling factor Δ or consuming a level.
- **Plaintext multiplication.** For a public vector $\mathbf{u}$, let $u = \mathrm{Encode}_{\Delta'}(\mathbf{u})$. Multiplication by u computes $(ub, ua) \pmod{Q_\ell}$ at scaling factor $\Delta\Delta'$.
- **Ciphertext multiplication.** For ciphertexts ct_1 and ct_2 with common modulus Q_ℓ and scaling factors Δ_1 and Δ_2, the tensor product is

$$(c_0, c_1, c_2) = \mathsf{ct}_1 \otimes \mathsf{ct}_2 = (b_1 b_2,\ b_1 a_2 + a_1 b_2,\ a_1 a_2) \pmod{Q_\ell}.$$

  The resulting three component ciphertext has scaling factor $\Delta_1 \Delta_2$.
- **Relinearization.** To reduce (c_0, c_1, c_2) to two components, relinearization uses an evaluation key to switch from s^2 to s. We denote the resulting ciphertext in $\mathcal{R}_{Q_\ell}^2$ by $\mathsf{Relin}(c_0, c_1, c_2)$.
- **Rotations and conjugation.** Homomorphic rotations cyclically permute the slots by applying a ring automorphism followed by key switching. Homomorphic conjugation, denoted by Conj, conjugates the value in each slot.

– **Rescaling.** For $\ell \geq 1$, a rescaling step consumes one level by reducing the modulus from Q_ℓ to $Q_{\ell-1}$ according to

$$\mathsf{Rescale}_{q_\ell}(b, a) = \left(\left\lfloor \frac{b}{q_\ell} \right\rceil, \left\lfloor \frac{a}{q_\ell} \right\rceil \right) \pmod{Q_{\ell-1}}.$$

The output has scaling factor Δ/q_ℓ.

2.3 Noise Cleaning

Discrete CKKS uses CKKS arithmetic to evaluate functions on discrete messages, with noise cleaning enabling exact recovery [3]. In the hybrid construction of AKP, the messages are encrypted and decrypted in coefficient encoding using an RLWE scheme. For a plaintext modulus p, a ciphertext modulus q, and a message $m \in \mathcal{R}_p$, an RLWE ciphertext $\mathsf{ct} = (b, a) \in \mathcal{R}_q^2$ satisfies

$$b + as = \frac{q}{p} m + e \pmod{q}.$$

The message is recovered by multiplying the centered representative of $b + as \bmod q$ by p/q, rounding each coefficient to the nearest integer, and reducing modulo p [2].

For functional bootstrapping, the input RLWE ciphertext is switched to modulus q_0 and interpreted as a CKKS ciphertext with initial scaling factor $\Delta = q_0$. After modulus raising and CoeffsToSlots, the decoded slot values have the form

$$\mathbf{I} + \frac{\mathbf{m}}{p} + \mathbf{v},$$

where $\mathbf{m}$ contains the integer message entries, $\mathbf{I}$ is the integer vector introduced by modulus raising, and $\mathbf{v}$ is the normalized noise [3].

Under the same normalization, an evaluation of $f : \mathbb{Z}_p \to \mathbb{Z}_p$ has output $f(\mathbf{m})/p + \mathbf{v}'$, where $\mathbf{v}'$ denotes the normalized output noise. Following Sparse-THI, the evaluation achieves noise cleaning when

$$\|\mathbf{v}'\|_\infty \leq \|\mathbf{v}\|_\infty.$$

For the identity function, noise cleaning preserves the message. When the normalized noise in the representation used for decryption is smaller than $1/(2p)$, nearest integer rounding recovers the message exactly.

2.4 Functional Bootstrapping

CKKS functional bootstrapping evaluates a function f, specified by a lookup table, on the coefficients of an encrypted message polynomial. For an RLWE ciphertext encrypting $m(X) = \sum_{i=0}^{N-1} m_i X^i$, the output encrypts a polynomial whose ith coefficient approximates $f(m_i)$ [2].

AKP implements this function evaluation through trigonometric Hermite interpolation. For a real function on $\{0, \ldots, p-1\}$, the interpolant can be written as

$$\Phi(y) = \operatorname{Re} T_f\left(e^{2\pi i y/p}\right),$$
$$\Phi(k) = f(k), \quad \Phi'(k) = 0 \qquad (0 \le k < p),$$

where T_f is a complex polynomial. To evaluate this interpolant, functional bootstrapping converts the message to slot encoding and returns the function values in coefficient encoding through the following steps.

- **ModSwitch.** The input RLWE ciphertext is switched from modulus q to q_0, giving a ciphertext with message term $(q_0/p)m(X)$.
- **ModRaise and CoeffsToSlots.** Modulus raising to Q_L introduces an integer multiple $q_0 I(X)$ into the decryption result. CoeffsToSlots then applies homomorphic encoding to represent the coefficients of this result in the CKKS slots.
- **EvalExp.** EvalExp uses a Chebyshev approximation of the complex exponential to obtain encrypted approximations of $e^{2\pi i m_i/p}$.
- **EvalLUT.** EvalLUT evaluates T_f at these approximate roots of unity and takes the real parts, yielding slot values $\widetilde{m}_i$ that approximate $f(m_i)$. The corresponding normalized noise is $(\widetilde{m}_i - f(m_i))/p$, under the convention of Section 2.3.
- **SlotsToCoeffs.** Let $\widetilde{m}(X) = \sum_{i=0}^{N-1} \widetilde{m}_i X^i$, and let Q' be the ciphertext modulus after this step. To return to coefficient encoding, SlotsToCoeffs applies homomorphic decoding with factor $Q'/(\Delta p)$, where Δ is the scaling factor of the ciphertext entering SlotsToCoeffs. The resulting ciphertext encrypts $(Q'/p)\widetilde{m}(X)$ at modulus Q'.
- **ModSwitch.** A final modulus switch from Q' to the desired output modulus Q gives an encryption of $(Q/p)\widetilde{m}(X)$.

For full packing, CoeffsToSlots represents the N input coefficients in two CKKS ciphertexts. EvalExp and EvalLUT are applied to both, and SlotsToCoeffs combines the results into a single ciphertext in coefficient encoding.

Multi-value functional bootstrapping. A single bootstrap can evaluate several lookup tables on the same encrypted input [2]. The common input allows the evaluations to share the steps through EvalExp and reuse the ciphertext powers needed for EvalLUT, while each lookup table uses its own interpolation coefficients.

When the outputs are required in slot encoding for subsequent CKKS computations, SlotsToCoeffs and the final modulus switch are omitted. For each batch of $N/2$ slots and functions $f_0, \ldots, f_{w-1}$, this variant returns ciphertexts encrypting $f_0(\mathbf{m}), \ldots, f_{w-1}(\mathbf{m})$ at a common level and scaling factor.

3 High Precision Noise Cleaning

We construct noise cleaning procedures for discrete CKKS that preserve the message while attaining a prescribed output noise bound. The main difficulty is to reduce the input error without allowing the errors introduced during evaluation to dominate the output. For bits, we address this problem by composing Hermite polynomials and choosing their evaluation parameters according to the complete error at each step. Within a step, representations with noisy auxiliary powers reduce the tensor degree while controlling the propagation of the auxiliary error. We extend these constructions to integers through digit extraction and recombination.

3.1 Noise Cleaning for Bits

Let $\mathsf{ct} = (b, a) \in \mathcal{R}_Q^2$ be a CKKS ciphertext with positive integer scaling factor Δ, and let $Z \in \mathcal{R}$ be the centered representative of $b + as \bmod Q$. In each slot ℓ, write $x_\ell = \sigma_\ell(Z)/\Delta = \zeta_\ell + h_\ell$, where $\zeta_\ell \in \{-1, 1\}$ is the signed bit and $h_\ell \in \mathbb{C}$ is the total input error, with $|h_\ell| \le B < 1$. For an integer $r \ge 1$, consider the binary Hermite cleaning polynomial

$$R_r(x) = c_r \sum_{j=0}^{r} \frac{(-1)^j \binom{r}{j}}{2j+1} x^{2j+1}, \qquad c_r = \frac{(2r+1)\binom{2r}{r}}{4^r}. \tag{1}$$

For $\zeta \in \{-1, 1\}$, this polynomial satisfies $R_r(\zeta) = \zeta$ and $R_r^{(j)}(\zeta) = 0$ for $1 \le j \le r$. The derivative $R_r'(x) = c_r(1 - x^2)^r$ gives the finite residual bound

$$|R_r(\zeta + h) - \zeta| \le \varepsilon_r(B) := c_r \sum_{j=0}^{r} \binom{r}{j} \frac{2^{r-j} B^{r+j+1}}{r+j+1} \tag{2}$$

for every complex perturbation h with $|h| \le B$. Appendix A.1 gives the derivation.

The residual bound (2) applies to exact polynomial evaluation. Homomorphic evaluation, however, also introduces key switching errors whose effect depends on where they enter the circuit. In particular, an error introduced by intermediate relinearization may be multiplied by subsequent ciphertexts or polynomial coefficients. For example, consider an evaluation of $R_1(x) = (3x - x^3)/2$ that relinearizes $\mathsf{ct} \otimes \mathsf{ct}$ before multiplying the result by ct. If this relinearization introduces decryption error κ and no rescaling is performed, the resulting tensor has scale $2\Delta^3$ and decrypts to

$$Z\big(3\Delta^2 - (Z^2 + \kappa)\big) = 2\Delta^3 R_1(Z/\Delta) - Z\kappa \pmod{Q}.$$

The term $-Z\kappa$ can be nonzero even when the input error vanishes. To avoid this multiplication of the switching error, we use delayed relinearization [16] and combine the weighted tensor powers before relinearizing their sum.

For this purpose, we express all polynomial terms at a common scaling factor before forming their sum. Write $R_r(x) = \sum_{j=0}^{D} a_j x^j$, where $D = 2r + 1$ and

$a_j \in \mathbb{Q}$. We denote the jth tensor power of ct by $\mathsf{ct}^{\otimes j}$ and set $\mathsf{ct}^{\otimes 0} = (1)$. Since $\mathsf{ct}^{\otimes j}$ has scale Δ^j, multiplication by Sa_j/Δ^j represents the term $a_j x^j$ at scale S. We therefore choose a positive integer S such that

$$w_j = \frac{Sa_j}{\Delta^j} \in \mathbb{Z} \qquad (0 \le j \le D).$$

Multiplication by these integer weights requires no approximate plaintext encoding [11]. A common denominator L of the coefficients gives the admissible choice $S = L\Delta^D$. More generally, the smallest admissible S is the least common multiple of the reduced denominators of a_j/Δ^j.

After extending each tensor to $D+1$ components by appending zeros, we form

$$\mathbf{T} = \sum_{j=0}^{D} w_j \, \mathsf{ct}^{\otimes j} \in \mathcal{R}_Q^{D+1},$$

without intermediate relinearization or rescaling. The tensor operations and integer multiplications are exact modulo Q, so decryption of $\mathbf{T}$ gives $SR_r(Z/\Delta) \pmod{Q}$. Under the canonical embedding, the polynomial $SR_r(Z/\Delta)$ has slot value $SR_r(x_\ell)$, whose deviation from $S\zeta_\ell$ is bounded by $S\varepsilon_r(B)$.

To obtain a ciphertext in $\mathcal{R}_Q^2$, we relinearize $\mathbf{T}$ by switching each component associated with s^j, for $2 \le j \le D$, to s. We denote this operation by Relin, with subscripts distinguishing evaluation key sets. The output retains modulus Q and scaling factor S. Let κ_j denote the complete decryption error of the component switch from s^j to s, including modulus reduction, and suppose that $\|\kappa_j\|_\infty \le A$ for $2 \le j \le D$. Then the aggregate error $\kappa = \sum_{j=2}^{D} \kappa_j$ satisfies $\|\kappa\|_\infty \le (D-1)A$, and the resulting ciphertext decrypts to

$$SR_r(Z/\Delta) + \kappa \pmod{Q}.$$

Since the weighted polynomial terms are combined before relinearization, the final switching errors enter the sum without further multiplication by the integer weights. The estimate $\|\sigma(\kappa)\|_\infty \le (D-1)NA$ then bounds their contribution to the decoded error by $(D-1)NA/S$.

After relinearization, we may reduce the output modulus and scaling factor by applying a prescribed sequence of CKKS rescalings, as defined in Section 2.2, whose primes have product $d \mid Q$. We denote this operation by $\mathsf{Rescale}_d$, with $\mathsf{Rescale}_1$ denoting the identity. The resulting modulus is $Q' = Q/d$, and the scaling factor is S/d. To bound the accumulated rounding error, we divide each individual bound in the canonical embedding norm by the product of all subsequent rescaling primes and denote their sum by ρ. The individual bounds follow from Lemma B.1 for nearest integer rounding. Without rescaling, $d = 1$ and $\rho = 0$.

The rescaled ciphertext has scale S/d, which need not be integral. Since the next cleaning step uses integer input scales, we decode an intermediate output at an integer Δ'. This changes the decoded polynomial value by a factor of $S/(d\Delta')$, contributing $(S/(d\Delta') - 1)\zeta_\ell$ to the error, in addition to the scaled residual. The

final output has no such integrality requirement, so it may retain the positive rational scale $\Delta' = S/d$ and avoid this contribution.

Proposition 3.1 (Error bound for one cleaning step). *Suppose that the input error, individual key switching errors, and rescaling errors satisfy the bounds above. If*

$$S\big(1+\varepsilon_r(B)\big) + (D-1)NA + d\rho < \frac{Q}{2},$$

then no modular wraparound occurs after relinearization or during rescaling. At scale Δ', the output ciphertext $\mathsf{ct}' \in \mathcal{R}^2_{Q'}$ has decoded values x'_ℓ whose maximum error $B' := \max_\ell |x'_\ell - \zeta_\ell|$ satisfies

$$B' \le \frac{S\varepsilon_r(B) + (D-1)NA + |S - d\Delta'|}{d\Delta'} + \frac{\rho}{\Delta'}.$$

The bound holds simultaneously in all slots. If each assumed bound holds except with a specified failure probability, the sum of these probabilities bounds the failure probability of the conclusion.

Proof. The polynomial $SR_r(Z/\Delta) + \kappa$ has canonical embedding norm at most $S(1+\varepsilon_r(B)) + (D-1)NA$. At each stage of rescaling, the accumulated rounding bound multiplied by the product of the primes already removed is at most $d\rho$. The modulus condition therefore keeps this norm below half the current modulus throughout rescaling. Since $\|a\|_\infty \le \|\sigma(a)\|_\infty$, the same inequality holds for the coefficient norm, so no modular wraparound occurs.

Let $e \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{R}$ denote the accumulated rounding error, with $\|\sigma(e)\|_\infty \le \rho$. Decryption of the output ciphertext therefore gives the centered representative $Z' = (SR_r(Z/\Delta) + \kappa)/d + e$. Hence, at scale Δ',

$$\begin{aligned} x'_\ell - \zeta_\ell = \frac{S}{d\Delta'}\big(R_r(x_\ell) - \zeta_\ell\big) + \left(\frac{S}{d\Delta'} - 1\right)\zeta_\ell \\ + \frac{\sigma_\ell(\kappa)}{d\Delta'} + \frac{\sigma_\ell(e)}{\Delta'}. \end{aligned}$$

The claimed bound follows from $|R_r(x_\ell) - \zeta_\ell| \le \varepsilon_r(B)$, $|\zeta_\ell| = 1$, and $\|\sigma(\kappa)\|_\infty \le (D-1)NA$. A union bound gives the probability statement. □

Without rescaling, we take $d = 1$, $\Delta' = S$, and $\rho = 0$, so the bound reduces to $B' \le \varepsilon_r(B) + (D-1)NA/S$. For a fixed polynomial R_r and switching bound A, increasing S reduces the relinearization contribution $(D-1)NA/S$ without changing the residual $\varepsilon_r(B)$.

3.2 Composition and Parameter Selection

We construct a high precision noise cleaning method by composing the Hermite evaluations of Section 3.1. Given an input ciphertext and a prescribed output noise bound, we jointly select the number of cleaning steps, the polynomial degree at each step, and the corresponding scales and key switching parameters.

In a composition, each intermediate relinearization or rescaling error becomes part of the input to the next Hermite polynomial. Later steps can therefore reduce this error along with the preceding polynomial residual. By contrast, errors introduced after the last polynomial evaluation remain in the output. This difference allows intermediate evaluations to tolerate larger errors, provided that the subsequent steps reduce them sufficiently. The bound of Section 3.1 determines the admissible errors at each step.

Composition. For an integer $k \geq 1$ and positive integers $r_1, \ldots, r_k$, we define the composite cleaning polynomial

$$F = R_{r_k} \circ \cdots \circ R_{r_1}.$$

The polynomial R_{r_1} is evaluated first, and each subsequent polynomial is applied to the output of the preceding evaluation. The Hermite conditions in Section 3.1 give $R_{r_i}(\zeta + h) - \zeta = O(h^{r_i+1})$ for $\zeta \in \{-1, 1\}$ as $h \to 0$. Consequently, the exact composition satisfies

$$F(\zeta + h) - \zeta = O\Big(h^{\prod_{i=1}^{k}(r_i+1)}\Big), \qquad \deg F = \prod_{i=1}^{k}(2r_i + 1).$$

The contraction order is therefore the product of the individual orders. However, homomorphic evaluation need not construct a tensor of degree $\deg F$, since each constituent polynomial can be evaluated separately. For $k = 0$, we take $F(x) = x$.

To evaluate this composition homomorphically, let the input ciphertext have positive integer scale Δ_0, modulus Q_0, and decoded signed bit error bounded by $B_0 < 1$. For a target $0 < B_\star < 1$, step i evaluates R_{r_i} at positive integer scale S_i, relinearizes the resulting tensor, and optionally rescales by an ordered sequence of primes with product d_i. The input to this step has scale Δ_{i-1} and modulus Q_{i-1}. The output is decoded at positive integer scale Δ_i and has modulus $Q_i = Q_{i-1}/d_i$.

As in Section 3.1, A_i bounds each key switching error in the coefficient norm, and ρ_i bounds the accumulated rescaling error in the canonical embedding norm. Without rescaling, $d_i = 1$ and $\rho_i = 0$. With these bounds, Proposition 3.1 gives the recurrence

$$B_i = \frac{S_i \varepsilon_{r_i}(B_{i-1}) + 2r_i N A_i + |S_i - d_i\Delta_i|}{d_i\Delta_i} + \frac{\rho_i}{\Delta_i}.$$

Because B_{i-1} bounds the complete input error at step i, $\varepsilon_{r_i}(B_{i-1})$ accounts for the propagation of all preceding errors through R_{r_i}. For this recurrence to apply

at every step, we require $B_{i-1} < 1$, $d_i \mid Q_{i-1}$, and the integer weight condition of Section 3.1, together with

$$S_i\big(1 + \varepsilon_{r_i}(B_{i-1})\big) + 2r_i N A_i + d_i \rho_i < \frac{Q_{i-1}}{2}.$$

This modulus condition prevents modular wraparound after relinearization and throughout rescaling.

Proposition 3.2 (Composition). *Suppose that every step satisfies the conditions above. Under the assumed input, key switching, and rescaling error bounds, the final ciphertext preserves the signed bits and has decoded error at most B_k, at scale Δ_k and modulus Q_k. In particular, $B_k \le B_\star$ suffices to meet the output requirement. The failure probability is at most the sum of the failure probabilities of the assumed bounds.*

Proof. Proposition 3.1 applies successively to the outputs of the preceding steps, giving the error bound by induction. A union bound gives the failure probability. □

Intermediate accuracy and scales. The final step determines the accuracy required of the preceding output. For fixed parameters $r, S, d, \Delta', A, \rho$, its error bound meets the target under the sufficient input condition

$$\varepsilon_r(B) \le \frac{d\Delta' B_\star - 2rNA - |S - d\Delta'| - d\rho}{S}.$$

When the right hand side is nonnegative, monotonicity of ε_r determines the admissible input errors for the final step. The preceding steps must meet this requirement while retaining a scale and modulus compatible with the final evaluation.

The intermediate scale determines the integer weights of the next polynomial, so reducing this scale can lower the modulus required for that evaluation. However, the preceding rescaling consumes modulus and introduces rounding error. The choice of decoding scale Δ' instead of S/d also contributes to the error entering the next step. These contributions must remain within the admissible input bound, which couples the choice of intermediate scale to the polynomial order.

Example. Consider one evaluation of R_3 and two successive evaluations of R_1, with relinearization after each evaluation. Near the message 1, the exact polynomials satisfy

$$R_3(1+h) - 1 = -\frac{35}{8}h^4 + O(h^5),$$
$$R_1\big(R_1(1+h)\big) - 1 = -\frac{27}{8}h^4 + O(h^5).$$

Both residuals are of order h^4. Although the composite polynomial has degree nine rather than seven, evaluating its two constituents separately requires only

cubic tensors. For a positive integer input scale $\Delta > 1$, the common denominator choices $S = L\Delta^D$ from Section 3.1 give the comparison below.

Table 1. Direct tensor evaluation without rescaling, using common denominator scales.

Evaluation	Maximum tensor degree per step	Total component switches	Final scale
R_3	7	6	$16\Delta^7$
$R_1 \to R_1$	3	4	$16\Delta^9$

The composition uses fewer component switches, at the cost of a larger final scale. The first evaluation gives $2\Delta^3$, and the second gives $2(2\Delta^3)^3 = 16\Delta^9$. If we instead rescale between the two evaluations and decode at integer scale Δ_1, the second evaluation uses $2\Delta_1^3$. The residual is then bounded by $\varepsilon_1(B_1)$, where B_1 includes the first relinearization, rescaling, and decoding errors. The benefit of intermediate rescaling therefore depends on how much modulus can be removed without violating the input error bound required by the second evaluation.

Parameter selection. Let $\mathcal{C}$ be a finite family of candidate compositions with at most $k_{\max}$ steps, including the empty composition. Each candidate specifies the polynomial orders, tensor evaluation circuits, common scales, ordered rescaling primes, integer decoding scales, and key switching parameters. These choices determine the bounds A_i and ρ_i. We restrict $\mathcal{C}$ to configurations satisfying the prescribed security requirements for ciphertexts and evaluation keys, the total failure probability budget, and the storage limit for distinct evaluation keys.

For a candidate Π, let $C(\Pi)$ denote its arithmetic cost under a fixed model. The cost includes tensor multiplication, addition, integer multiplication, complete key switching, and rescaling at their respective moduli, with tensor powers reused according to the selected evaluation circuits. For storage, compatible evaluation keys are counted once, whereas every application contributes to the computation cost.

Let $\mathcal{O}$ denote the permitted pairs of final scale and modulus. In Algorithm 1, $r, S, d, \Delta', A, \rho$ denote the current step parameters and bounds, and a_j is the coefficient of x^j in R_r, for $0 \le j \le 2r + 1$.

Algorithm 1 Selection of a composition using direct tensor evaluation

Input: (B_0, Δ_0, Q_0), target $B_\star$, candidate family $\mathcal{C}$, permitted outputs $\mathcal{O}$, and cost function C

Output: A feasible composition of minimum cost in $\mathcal{C}$, or $\perp$

$\Pi^\star \leftarrow \perp$
for each $\Pi \in \mathcal{C}$ **do**
  $(B, \Delta, Q) \leftarrow (B_0, \Delta_0, Q_0)$
  for each step of Π **do**
    if $B \geq 1$ **or** $d \nmid Q$ **or** $\exists j \in \{0, \ldots, 2r+1\} : \ Sa_j/\Delta^j \notin \mathbb{Z}$ **then**
      continue outer loop
    if $S(1 + \varepsilon_r(B)) + 2rNA + d\rho \geq Q/2$ **then**
      continue outer loop
    $B \leftarrow \dfrac{S\varepsilon_r(B) + 2rNA + |S - d\Delta'|}{d\Delta'} + \dfrac{\rho}{\Delta'}$
    $(\Delta, Q) \leftarrow (\Delta', Q/d)$
  if $B \leq B_\star$ **and** $(\Delta, Q) \in \mathcal{O}$ **then**
    if $\Pi^\star = \perp$ **or** $C(\Pi) < C(\Pi^\star)$ **then**
      $\Pi^\star \leftarrow \Pi$
return $\Pi^\star$

Every accepted candidate satisfies the output noise bound by Proposition 3.2. The algorithm examines the entire finite family $\mathcal{C}$ and therefore returns a composition of minimum cost among those satisfying the stated error, modulus, and output conditions.

3.3 Evaluation with Noisy Auxiliary Powers

Direct tensor evaluation avoids intermediate key switching, but an evaluation of R_r produces a tensor of degree $2r+1$ and requires evaluation keys for $s^2, \ldots, s^{2r+1}$. Higher Hermite orders therefore increase both the tensor degree and the number of required keys. We address these costs through representations that reduce the tensor degree while meeting the prescribed output noise bound for the same cleaning polynomial.

An intermediate relinearization can reduce the degree of subsequent tensor products, but its error enters the remaining polynomial computation. Consider an auxiliary ciphertext for x^3, where $x = \zeta + h$ is the decoded input and $\zeta \in \{-1, 1\}$. The auxiliary ciphertext represents $T = x^3 + \eta$, with error η. A polynomial representation $P(x, T)$ satisfying $P(x, x^3) = R_r(x)$ preserves the Hermite residual when $\eta = 0$. However, the coefficient of η in $P(x, x^3 + \eta) - R_r(x)$ is $\partial_T P(x, x^3)$, which need not vanish at the message values. The auxiliary error can therefore dominate the output even when the Hermite polynomial substantially reduces the input error.

We observe that the propagation of this auxiliary error depends on the polynomial representation, although the represented function remains fixed. Specifically, for any $K \in \mathbb{Q}[x]$, the correction term $-(T - x^3)K(x)$ leaves $P(x, x^3)$ unchanged and subtracts $K(x)$ from the coefficient of η. This freedom allows us

to impose vanishing conditions on that coefficient at both message values. For the Hermite family, these conditions can be satisfied at the minimum total degree without increasing the coefficient denominator. The resulting representation can therefore tolerate a larger auxiliary error without changing the cleaning function.

Representations of Hermite polynomials. Let $r \geq 2$ and write $R_r(x) = \sum_{m=0}^{2r+1} a_m x^m$. We first group the exponents according to their remainder modulo three:

$$N_r(x,T) = \sum_{\substack{0 \leq i < 3,\ j \geq 0 \\ i+3j \leq 2r+1}} a_{i+3j} x^i T^j.$$

Thus, $N_r(x, x^3) = R_r(x)$. A monomial $x^i T^j$ has total degree $i + j$ in the representation and degree $i + 3j$ after substitution of $T = x^3$.

Set $\nu = \lfloor (r+1)/3 \rfloor$. The polynomial N_r has total degree $2\nu+1$, so corrections of the form $(T-x^3)(x^2-1)^\ell$, with $\ell < \nu$, do not increase this degree. To determine their coefficients, expand the even polynomial $\partial_T N_r(x, x^3)$ as

$$\partial_T N_r(x, x^3) = \sum_{\ell=0}^{r-1} \beta_\ell (x^2-1)^\ell.$$

To cancel the first ν terms without changing R_r, define

$$P_r(x,T) = N_r(x,T) - (T - x^3) \sum_{\ell=0}^{\nu-1} \beta_\ell (x^2-1)^\ell,$$

$$\partial_T P_r(x, x^3) = \sum_{\ell=\nu}^{r-1} \beta_\ell (x^2-1)^\ell.$$

The correction term vanishes when $T = x^3$, so P_r still represents R_r. Since $x^2 - 1 = h(2\zeta + h)$ at $x = \zeta + h$, the remaining coefficient of η vanishes to order at least ν at both message values.

Theorem 3.3 (Hermite representations). *For every integer $r \geq 2$, the polynomial P_r has minimum total degree among all polynomials in $\mathbb{Q}[x,T]$ satisfying $P(x,x^3) = R_r(x)$. The total degree D_r and the least positive integer L_r such that $L_r P_r \in \mathbb{Z}[x,T]$ are*

$$D_r = 2\left\lfloor \frac{r+1}{3} \right\rfloor + 1, \qquad L_r = 2^{\,2r - s_2(r)},$$

where $s_2(r)$ is the number of ones in the binary expansion of r. Every monomial $x^i T^j$ in P_r satisfies $i + 3j \leq 2r + 1$. For fixed r, uniformly over $\zeta \in \{-1, 1\}$ and complex perturbations $|h| \leq B$, $|\eta| \leq \theta$,

$$\left| P_r\big(\zeta + h, (\zeta+h)^3 + \eta\big) - \zeta \right| = O\big(B^{r+1} + B^\nu \theta + \theta^2\big)$$

as $B, \theta \to 0$.

The degree lower bound follows by minimizing $i + j$ for each exponent $i + 3j$ required by R_r. The representation N_r attains this bound, and the correction preserves it. Moreover, the correction coefficients are integer linear combinations of the coefficients of R_r, so the denominator does not increase. Appendix A gives the complete proof.

For example, when $r = 4$, the initial representation satisfies $\partial_T N_4(\zeta, \zeta^3) = -297/128$. The correction $297(T - x^3)/128$ gives

$$\begin{aligned} 128P_4(x, T) = 315x - 297x^3 - 123T \\ + 378x^2T - 180xT^2 + 35T^3. \end{aligned}$$

The resulting auxiliary error satisfies the exact identity

$$\begin{aligned} 128\big(P_4(x, x^3 + \eta) - R_4(x)\big) = (x^2 - 1)(105x^4 - 255x^2 + 123)\eta \\ + (-180x + 105x^3)\eta^2 + 35\eta^3. \end{aligned} \tag{3}$$

The linear contribution is therefore $O(B\theta)$, rather than $O(\theta)$, without increasing the total degree or coefficient denominator. However, the quadratic coefficient remains nonzero at the message values, so the required auxiliary accuracy also depends on the higher powers of η.

Auxiliary accuracy and finite error. To bound these contributions for finite errors, let $E(P; B, \theta)$ denote the maximum absolute output error over both message values and the complex discs $|h| \le B$, $|\eta| \le \theta$. For nonnegative integers u, v, let $e_{\zeta,u,v}$ be the coefficient of $h^u\eta^v$ in the error polynomial. The error expansion and its coefficient bound are

$$\begin{aligned} P\big(\zeta + h, (\zeta + h)^3 + \eta\big) - \zeta = \sum_{u,v} e_{\zeta,u,v} h^u \eta^v, \\ \varepsilon(P; B, \theta) = \max_{\zeta=\pm 1} \sum_{u,v} |e_{\zeta,u,v}| B^u \theta^v \end{aligned} \tag{4}$$

and the triangle inequality gives $E(P; B, \theta) \le \varepsilon(P; B, \theta)$. The terms with $v = 0$ give the Hermite residual, while those with $v \ge 1$ account for all powers of the auxiliary error and their products with the input error.

The same expansion characterizes the auxiliary accuracy required for a given contraction order. For a fixed representation P and constants $c, \alpha > 0$,

$$\begin{aligned} E(P; B, cB^\alpha) = \Theta(B^\tau), \\ \tau = \min_{\substack{\zeta=\pm 1,\ u,v\ge 0 \\ e_{\zeta,u,v}\neq 0}} (u + \alpha v), \end{aligned} \tag{5}$$

as $B \to 0$. The coefficient sum bounds the error from above, while Cauchy's coefficient estimate gives a lower bound of the same order.

Consequently, representations with a prescribed contraction order can be obtained through linear constraints on their coefficients. For a total degree bound

D, write $P(x,T) = \sum_{i+j\le D} p_{ij} x^i T^j$. A contraction order of at least γ requires

$$\sum_{\substack{i+j\le D \\ i+3j=m}} p_{ij} = a_m \quad \text{for every } m,$$
$$e_{\zeta,u,v} = 0 \qquad \text{whenever } u+\alpha v < \gamma, \quad \zeta = \pm 1,$$

where $a_m = 0$ outside the coefficient range of R_r. The first set of equations preserves the cleaning function. The second removes the error terms that would prevent the required order. Together, these equations form a linear system over $\mathbb{Q}$. Appendix A derives the coefficient constraints and the corresponding lower bounds within the prescribed representation space.

For R_4 and R_6, this characterization gives matching constructions and lower bounds.

Proposition 3.4 (Attainable contraction orders). *Let $\theta = cB^\alpha$, where $c > 0$ is fixed. Among representations of R_4 with total degree at most four, the maximum contraction order is*

$$\tau_4(\alpha) = \min\{5, 1+\alpha, 2\alpha\}, \qquad \alpha > 0,$$

and P_4 attains this order. Among representations of R_6 with total degree at most six, the maximum contraction order is

$$\tau_6(\alpha) = \min\{7, \max\{3+\alpha, 2\alpha\}\}, \qquad \alpha \ge 2.$$

Two representations of total degree five, each with eleven nonzero terms and coefficient denominator 1024*, attain the two branches.*

Thus, R_4 retains fifth order contraction when $\alpha \ge 4$. For R_6, seventh order contraction is attainable when $\alpha \ge 7/2$, whereas the general representation P_6 requires $\alpha \ge 5$. The optimized representations use two additional terms but retain the same degree and denominator. The coefficients and the proofs of the lower bounds appear in Appendix A.

The contraction orders describe the dependence on auxiliary accuracy as the input error tends to zero. At finite B and θ, the coefficients and higher order terms also contribute to the output error. Parameter selection therefore uses $\varepsilon(P; B, \theta)$ to account for all terms in the error polynomial.

Homomorphic evaluation. The finite error bound requires an auxiliary ciphertext with error at most θ. We choose the auxiliary scale and rescaling parameters so that this error bound holds and sufficient modulus remains for evaluation of P. Let $\mathsf{ct} \in \mathcal{R}_Q^2$ have positive integer scale Δ, with Z defined as in Section 3.1, so that the decoded values are $x_\ell = \sigma_\ell(Z)/\Delta = \zeta_\ell + h_\ell$, with $|h_\ell| \le B < 1$.

For a positive integer u, we construct the auxiliary ciphertext as

$$\mathsf{ct}_T = \mathsf{Rescale}_{d_a}\left(\mathsf{Relin}_a(u\,\mathsf{ct}^{\otimes 3})\right) \in \mathcal{R}^2_{Q/d_a},$$

where $d_a \mid Q$ is the product of a prescribed sequence of rescaling primes. We decode this ciphertext at a positive integer scale Γ, chosen jointly with u and d_a. Let A_a bound the complete coefficient error of each of the two component switches, and let ρ_a bound the accumulated rounding error in the canonical embedding norm, as defined in Section 3.1.

Under the modulus conditions below, the auxiliary values satisfy $T_\ell = x_\ell^3 + \eta_\ell$, with $|\eta_\ell| \le \theta$, provided that

$$\left|\frac{u\Delta^3}{d_a\Gamma} - 1\right|(1+B)^3 + \frac{2NA_a}{d_a\Gamma} + \frac{\rho_a}{\Gamma} \le \theta. \tag{6}$$

Here the first term comes from decoding at Γ instead of $u\Delta^3/d_a$, and the other two terms come from key switching and rescaling. Without auxiliary rescaling, the choice $u = d_a = 1$, $\Gamma = \Delta^3$ reduces this condition to $2NA_a/\Delta^3 \le \theta$.

The two ciphertexts must have a common modulus to evaluate P. We reduce the input ciphertext components modulo Q/d_a while retaining scale Δ. For a representation $P(x,T) = \sum_{i,j} p_{ij} x^i T^j$ of total degree D, the tensor corresponding to $x^i T^j$ has scale $\Delta^i \Gamma^j$. We therefore choose a positive integer S satisfying

$$w_{ij} = \frac{S p_{ij}}{\Delta^i \Gamma^j} \in \mathbb{Z} \qquad \text{for every } p_{ij} \ne 0. \tag{7}$$

The least admissible S is the least common multiple of the reduced denominators of $p_{ij}/(\Delta^i\Gamma^j)$. With these integer weights, the evaluation forms

$$\mathbf{T} = \sum_{p_{ij}\ne 0} w_{ij}\big(\mathsf{ct}^{\otimes i} \otimes \mathsf{ct}_T^{\otimes j}\big) \in \mathcal{R}_{Q/d_a}^{D+1}$$

without further relinearization or rescaling, using the tensor conventions of Section 3.1. Under the modulus conditions below, the decryption polynomial has canonical embedding coordinates $SP(x_\ell, T_\ell)$.

For P_r at $\Gamma = \Delta^3$, the choice $S = L_r\Delta^{2r+1}$ is admissible and is minimal when Δ is a power of two. The scale depends on the weighted degree $i + 3j$, whereas the tensor degree depends on $i + j$. Thus, the representation reduces the final tensor degree while retaining the exponent $2r + 1$ in this common scale.

Let π specify the representation, integer multiplier, scales, ordered rescaling primes, evaluation keys, and tensor evaluation circuit. We denote the computation through $\mathbf{T}$ by $\mathsf{EvalTensor}(\mathsf{ct};\pi)$ and define

$$\mathsf{Clean}(\mathsf{ct};\pi) = \mathsf{Rescale}_d\big(\mathsf{Relin}_f(\mathsf{EvalTensor}(\mathsf{ct};\pi))\big),$$

where $d_a d \mid Q$. The output has modulus $Q/(d_a d)$ and is decoded at scale $\Delta' > 0$, following the convention of Section 3.1. A subsequent cleaning step uses an integer Δ'. An empty rescaling sequence has divisor product one and rounding bound zero. For direct evaluation, we use $P(x,T) = R_r(x)$ and omit auxiliary generation, under the conditions of Proposition 3.1.

Complete error and evaluation cost. The finite bound $\varepsilon(P; B, \theta)$ accounts for the input and auxiliary errors after evaluation of the represented polynomial. Since tensor evaluation uses exact modular operations, the remaining error comes from final relinearization, rescaling, and decoding at Δ'.

Theorem 3.5 (Complete cleaning bound). *Consider an auxiliary evaluation satisfying the integer weight and divisibility conditions above. Let A_f bound the complete coefficient error of each final component switch, and let ρ bound the accumulated rounding error of the final rescaling sequence in the canonical embedding norm. Assume that the input and auxiliary error bounds hold in every slot and that*

$$\begin{aligned} \Delta(1+B) &< \frac{Q}{2d_a}, \\ u\Delta^3(1+B)^3 + 2NA_a + d_a\rho_a &< \frac{Q}{2}, \\ S\big(1+\varepsilon(P;B,\theta)\big) + (D-1)NA_f + d\rho &< \frac{Q}{2d_a}. \end{aligned} \tag{8}$$

Then the reduced input and the polynomials obtained after auxiliary and final relinearization have centered coefficients, as do the polynomials obtained at each subsequent rescaling step. The output of Clean*, decoded at scale Δ', has maximum absolute error bounded by*

$$B' \le \frac{S\varepsilon(P;B,\theta) + (D-1)NA_f + |S - d\Delta'|}{d\Delta'} + \frac{\rho}{\Delta'}. \tag{9}$$

The bound holds simultaneously in all slots. The failure probability is at most the sum of the exceptional probabilities of the assumed input, key switching, and rescaling bounds.

Before final relinearization, the decryption polynomial has canonical embedding coordinates $S(\zeta_\ell + e_\ell)$, with $|e_\ell| \le \varepsilon(P; B, \theta)$. The final component switches contribute at most $(D-1)NA_f$ in the canonical embedding norm. The error analysis of Section 3.1 then gives the stated bound after rescaling and decoding. Appendix B provides the complete proof, including the modulus conditions at every intermediate step.

For fixed final evaluation parameters, the requirement $B' \le B_\star$ limits the admissible auxiliary error through the monotonicity of $\varepsilon(P; B, \theta)$ in θ. The auxiliary parameters must satisfy (6) with this error bound. Although a larger auxiliary scale Γ can reduce the decoded switching and rounding errors, it also changes the integer weights and the minimum admissible S. We therefore select the representation, auxiliary scale, and parameters of both switching stages jointly under the complete output bound.

To compare the error bounds directly, suppose that the auxiliary evaluation uses $\Gamma = \Delta^3$, $u = d_a = 1$, and that both it and the direct evaluation use the same S, d, Δ', A_f, ρ. The auxiliary evaluation has no larger complete error bound exactly when

$$S\big(\varepsilon(P;B,\theta) - \varepsilon_r(B)\big) \le (2r - D + 1)NA_f.$$

Thus, at these common parameters, the reduction in final switching error must compensate for the additional polynomial error caused by the auxiliary ciphertext.

The auxiliary construction requires two component switches to generate ct_T and at most $D-1$ for final relinearization, compared with $2r$ for direct evaluation. The complete cost also includes auxiliary tensor multiplication, evaluation of P, integer multiplications, and both rescaling sequences at their respective moduli. Compatible evaluation keys may be shared between the two switching stages.

Auxiliary evaluations enter Algorithm 1 under the conditions of Theorem 3.5, including the auxiliary error requirement (6). The algorithm uses (9) for the error update and replaces the scale and modulus update by

$$(\Delta, Q) \leftarrow \big(\Delta', Q/(d_a d)\big).$$

The candidate family can therefore combine direct and auxiliary steps under the same output, security, and resource constraints.

3.4 Noise Cleaning for Integer Messages

To clean an integer message, we use functional bootstrapping to extract its binary digits and apply the preceding cleaning procedures to each digit. The required digit accuracy is determined by integer recombination, which multiplies each digit error by its bit weight. We therefore select the cleaning parameters for the recombined output and perform the final relinearization only after adding the weighted digit tensors. The first cleaning polynomial can also be evaluated jointly with the conjugation that completes digit extraction.

Let $w \geq 1$ and $p = 2^w$. For $m \in \{0, \ldots, p-1\}$, define the signed digit functions

$$f_j(m) = 2\big(\lfloor m/2^j \rfloor \bmod 2\big) - 1, \qquad 0 \leq j < w.$$

The values $\zeta_j = f_j(m) \in \{-1, 1\}$ give the centered integer representation

$$v := 2m - (p-1) = \sum_{j=0}^{w-1} 2^j \zeta_j.$$

Digit extraction and the first cleaning step. We apply the multi-value functional bootstrapping of Section 2.4 to the signed digit functions. The evaluations share the transformations preceding EvalLUT and reuse the same powers in polynomial evaluation. Under the interpolation convention of Section 2.4, we evaluate $T_{f_j}/2$ for each function f_j, so addition of the conjugate gives its real output. To combine this operation with the first cleaning step, we retain the ciphertexts before conjugation and denote the resulting procedure by DigitExtract, with outputs $U_0, \ldots, U_{w-1}$.

For the joint evaluation, let ι be the automorphism $X \mapsto X^{-1}$, which satisfies $\sigma(\iota(Z)) = \overline{\sigma(Z)}$. Homomorphic conjugation applies ι to the ciphertext components and switches the secret key from $\iota(s)$ to s. We write $\mathrm{Tr}(Z) = Z + \iota(Z)$, so that $\sigma(\mathrm{Tr}(Z)) = 2\,\mathrm{Re}\,\sigma(Z)$.

Suppose that the outputs U_j have modulus Q_0 and positive integer scale Δ_0, and denote by Z_j the centered representative obtained by decrypting U_j. The extraction parameters provide bounds $0 \le B_0 < 1$ and $M \ge 0$ such that

$$\left| \frac{\sigma_\ell(\mathrm{Tr}(Z_j))}{\Delta_0} - \zeta_{j,\ell} \right| \le B_0, \qquad \|\sigma(Z_j)\|_\infty \le M\Delta_0$$

for every digit and slot. The first inequality bounds the signed digit error before the conjugation switch. The bound on the complex values represented by U_j controls their multiplication with an intermediate relinearization error in the joint evaluation below.

A separate conjugation introduces a switching error before the first cleaning polynomial is evaluated. To postpone this conjugation, we evaluate R_1 through the identity

$$H_{\Delta_0}(\mathrm{Tr}(Z)) = \mathrm{Tr}\big(3\Delta_0^2 Z - Z^2\big(Z + 3\iota(Z)\big)\big),$$
$$H_{\Delta_0}(Y) = 3\Delta_0^2 Y - Y^3 = 2\Delta_0^3 R_1(Y/\Delta_0).$$

Since the trace appears outside the polynomial expression, the conjugation can be completed after this evaluation. The integer coefficients give output scale $S_1 = 2\Delta_0^3$ and, as in Section 3.1, require no approximate plaintext encoding.

To evaluate the expression inside the trace, we first relinearize $U_j \otimes U_j$ at scale Δ_0^2. The resulting ciphertext decrypts to $Z_j^2 + \kappa_0$ modulo Q_0. Substitution into the trace expression then gives a quadratic expression in s and $\iota(s)$. We switch the component associated with $s\iota(s)$ using a conjugation key and shift the ciphertext components to represent multiplication by s. We then relinearize the resulting tensor and add its homomorphic conjugate. The construction requires four component switches using only the two key families for s^2 and $\iota(s)$. Appendix B.2 gives the component formulas and the complete error analysis.

Let A_2 and A_ι bound the complete coefficient errors of switches using these respective key families, including the errors from modulus reduction. For a ternary secret key, the additional decryption error has canonical embedding norm at most

$$E_1 = N\left((8M\Delta_0 + 2)A_2 + (2N+1)A_\iota\right).$$

The term containing $M\Delta_0$ bounds the propagation of κ_0 through the trace expression. The other terms bound the key switching errors introduced when handling the mixed component, relinearizing the resulting tensor, and applying conjugation.

All four switches precede rescaling. Let $d_1 \mid Q_0$ be the product of the prescribed rescaling primes, let ρ_1 be the bound on the accumulated rounding error defined in Section 3.1, and let Δ_1 be the positive integer output decoding scale. Under the modulus condition

$$S_1\big(1+\varepsilon_1(B_0)\big)+E_1+d_1\rho_1<\frac{Q_0}{2},$$

the output ciphertext ct_j has modulus Q_0/d_1 and signed digit error bounded by

$$\begin{aligned} B_1 &= \frac{S_1\varepsilon_1(B_0)+E_1+|S_1-d_1\Delta_1|}{d_1\Delta_1}+\frac{\rho_1}{\Delta_1}, \\ \varepsilon_1(B_0) &= \frac{3}{2}B_0^2+\frac{1}{2}B_0^3. \end{aligned} \tag{10}$$

Further cleaning and integer recombination. After the joint evaluation of the first R_1, the remaining cleaning steps start from error bound B_1, scale Δ_1, and modulus Q_0/d_1. We select these steps as in Section 3.2, with the representations of Section 3.3 as additional candidates.

For digit j, let $k_j \geq 0$ be the number of remaining steps. When $k_j \geq 1$, the first k_j-1 steps use Clean, whereas the last uses only $\mathsf{EvalTensor}$. Denote the final tensor by $\mathbf{T}_j$. At a common positive integer scale S, the polynomial evaluation for digit j gives $S(\zeta_j+e_j)$ in each slot, where $|e_j| \leq \varepsilon_j$. The bound ε_j includes all preceding errors and any auxiliary error introduced in the last evaluation. When $k_j = 0$, we take $\mathbf{T}_j = (S/\Delta_1)\mathsf{ct}_j$, requiring $S/\Delta_1 \in \mathbb{Z}$, and set $\varepsilon_j = B_1$.

For recombination, the common scale S must satisfy the integer weight conditions of every final evaluation. We also choose q dividing all final tensor moduli. After reduction of the components modulo q and zero padding of shorter tensors, we form

$$\mathbf{T}=\sum_{j=0}^{w-1}2^j\mathbf{T}_j \pmod q.$$

The corresponding polynomial sum has embedding $S(v+\sum_j 2^j e_j)$. Consequently, the digit errors are multiplied by their bit weights, whereas the error of relinearizing $\mathbf{T}$ is added after recombination and is not multiplied by these weights.

Let $D \geq 1$ bound the degree of $\mathbf{T}$, and let A_f bound the complete coefficient error of each final component switch. After relinearization, an optional rescaling sequence with product $d \mid q$ gives a ciphertext modulo q/d, decoded at positive rational scale Δ'. Let ρ bound the accumulated rounding error in the canonical embedding norm.

If

$$S\left(p-1+\sum_{j=0}^{w-1}2^{j}\varepsilon_{j}\right)+(D-1)NA_{f}+d\rho<\frac{q}{2},$$

then the maximum decoded error relative to v satisfies

$$E_{v}\leq\frac{S\sum_{j=0}^{w-1}2^{j}\varepsilon_{j}+(D-1)NA_{f}+(p-1)|S-d\Delta'|}{d\Delta'}+\frac{\rho}{\Delta'}. \tag{11}$$

This bound holds simultaneously in all slots whenever the extraction, switching, and rescaling bounds hold. The failure probability is at most the sum of their exceptional probabilities, with shared events counted once. Appendix B.1 derives the error bound from decryption of the weighted tensor sum.

For a decoded value $\widehat{v}$, define $\widehat{m}=(\widehat{v}+p-1)/2$. The normalized integer noise therefore satisfies

$$\frac{|\widehat{m}-m|}{p}\leq\frac{E_{v}}{2p}.$$

Hence, normalized noise at most 2^{-b} is guaranteed by $E_{v}\leq 2p\,2^{-b}$. If $E_{v}<1$, rounding $\mathrm{Re}\,\widehat{m}$ recovers m exactly.

For the final integer output, we may retain the exact scaling factor $\Delta'=S/d$. The term $(p-1)|S-d\Delta'|$ in (11) then vanishes, while the accumulated rounding error still contributes $d\rho/S$ to the decoded error.

Parameter selection and algorithm. A candidate specifies the extraction and first cleaning parameters, the remaining sequence for each digit, and common final parameters S,q,d,Δ'. Intermediate steps use the complete error recurrence of Sections 3.2 and 3.3. The last tensor evaluation for each digit contributes ε_{j} to the weighted integer bound above, which determines whether the selected parameters satisfy the output noise bound.

The cost includes the shared computation in digit extraction, the evaluation for every digit, and the final relinearization and rescaling sequence. We select a feasible candidate of minimum cost within the prescribed finite family, using the security, failure probability, and key storage constraints of Section 3.2.

Let π_{1} collect the extraction and joint evaluation parameters, and let $\pi_{j,i}$ specify the ith remaining step for digit j. We write $\mathsf{Joint}(U_{j};\pi_{1})$ for the joint cubic evaluation and its prescribed rescalings. Algorithm 2 uses parameters satisfying the conditions above and processes one batch of $N/2$ slots. For a fully packed input in coefficient encoding, the algorithm is applied to each of the two slot batches described in Section 2.4.

Algorithm 2 Integer noise cleaning with a joint first step

Input: A ciphertext ct of a w bit integer accepted by multi-value functional bootstrapping, parameters $\pi_1, (\pi_{j,i}), S, q, d, \Delta'$ satisfying the conditions for target 2^{-b}, and the required evaluation keys

Output: A ciphertext of $v = 2m - (p-1)$ modulo q/d, decoded at scale Δ', with normalized integer noise at most 2^{-b}

$(U_0, \ldots, U_{w-1}) \leftarrow \mathsf{DigitExtract}(\mathsf{ct}; f_0, \ldots, f_{w-1}, \pi_1)$
$\mathbf{T} \leftarrow (0)$
for $j = 0, \ldots, w-1$ **do**
    $\mathsf{ct}_j \leftarrow \mathsf{Joint}(U_j; \pi_1)$
    if $k_j = 0$ **then**
        $\mathbf{T}_j \leftarrow (S/\Delta_1)\mathsf{ct}_j$
    else
        for $i = 1, \ldots, k_j - 1$ **do**
            $\mathsf{ct}_j \leftarrow \mathsf{Clean}(\mathsf{ct}_j; \pi_{j,i})$
        $\mathbf{T}_j \leftarrow \mathsf{EvalTensor}(\mathsf{ct}_j; \pi_{j,k_j})$
    $\mathbf{T} \leftarrow \mathbf{T} + 2^j(\mathbf{T}_j \bmod q)$
return $\mathsf{Rescale}_d(\mathsf{Relin}_f(\mathbf{T}))$

Alternatively, conjugation and the first cleaning step can be evaluated separately. A conjugation switch with coefficient error bound A_ι gives the initial signed digit bound $B_0 + NA_\iota/\Delta_0$, which must be smaller than one for the cleaning bound to apply. In this alternative, the algorithm replaces Joint by $U_j + \mathsf{Conj}(U_j)$ followed by Clean for the first R_1 evaluation. The configurations evaluated in Section 5 use separate conjugation followed by cleaning.

Remark 3.6. When sufficient levels remain, leveled lookup table evaluation can supply the signed digit ciphertexts instead of functional bootstrapping [2]. The subsequent cleaning and integer recombination satisfy the same bounds. The joint first step also applies when the lookup table evaluation ends with the conjugation operation described above.

Output modulus and scale. The modulus of a completed integer ciphertext can be reduced further if the rounding error leaves its output within the prescribed noise bound. Let Q_0 and $\Delta > 0$ be its modulus and rational scaling factor, and write its centered decryption polynomial as $\Delta\sigma^{-1}(\mathbf{v}) + e_a + u_0$. Here $e_a, u_0 \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{R}$ satisfy $\|\sigma(e_a)\|_\infty \leq \Delta\epsilon$ and $\|u_0\|_\infty \leq U_0$. In particular, the complete bound above permits $\epsilon = E_v$ and $U_0 = 0$. Let $d \mid Q_0$, with $Q_0/d > 1$, be the product of the prescribed rescaling primes, and let ρ_c bound the accumulated rounding error in the coefficient norm. As with ρ, each individual bound is divided by the product of the subsequent rescaling primes.

The sufficient condition

$$\Delta(p-1+\epsilon)+U_0+d\rho_c<Q_0/2 \tag{12}$$

prevents modular wraparound throughout the sequence. At the exact output scale, the resulting parameters are

$$q=\frac{Q_0}{d},\qquad \Delta'=\frac{\Delta}{d},\qquad U=\frac{U_0}{d}+\rho_c. \tag{13}$$

The decoded error relative to $\mathbf{v}$ is at most

$$\epsilon+\frac{NU}{\Delta'}=\epsilon+\frac{NU_0}{\Delta}+\frac{Nd\rho_c}{\Delta}.$$

At scale Δ', the error already present in the input contributes the same decoded bound as before rescaling. Thus, the modulus can be reduced whenever the additional rounding contribution $Nd\rho_c/\Delta$ keeps the total error within the required output bound. Appendix B.1 proves these estimates and the modulus condition for every rescaling step.

4 Application to Asynchronous Threshold CKKS

We combine the integer noise cleaning of Section 3.4 with the KLLPS construction for strong simulation security [15]. Noise flooding must hide the complete ciphertext error while preserving exact recovery after Lagrange reconstruction. Since the KLLPS responses also contain RLWE errors, we derive joint conditions on the cleaning error and flooding distribution that account for these contributions.

4.1 Construction

We consider a t-out-of-n scheme with trusted setup and static corruption of fewer than t parties. The public parameters fix an odd sharing modulus Q and a decryption modulus $q>1$ dividing Q. Let $p=2^w$, where $w\geq 1$, and let $\mathbf{m}\in\{0,\dots,p-1\}^{N/2}$ be the integer output. The construction in Section 3.4 produces a ciphertext $\mathsf{ct}=(b,a)\in\mathcal{R}_q^2$ representing $\mathbf{v}=2\mathbf{m}-(p-1)$ at a positive rational scaling factor Δ'.

Suppose that $1\leq t\leq n$, where n is a power of two dividing $2N$. As in KLLPS [14], we use the interpolation points $\alpha_0=0$ and $\alpha_i=X^{(i-1)2N/n}$ for $i\in[n]=\{1,\dots,n\}$. The differences between distinct points are units in $\mathcal{R}_Q$, so Shamir secret sharing is well defined over this ring. The dealer samples a polynomial $f\in\mathcal{R}_Q[Y]$ of degree at most $t-1$, subject to $f(0)=s \bmod Q$, and gives $s_i=f(\alpha_i)$ to party i. The nonconstant coefficients are uniform in $\mathcal{R}_Q$ and independent of one another and of CKKS key generation.

The dealer also samples an independent polynomial $f_0 \in \mathcal{R}_q[Y]$ uniformly subject to $\deg f_0 < t$ and $f_0(0) = 0$, and gives $z_i = f_0(\alpha_i)$ to party i. The pair (s_i, z_i) forms its partial decryption key.

Let $H : \{0,1\}^* \to \mathcal{R}_q$ and $H_\chi : \{0,1\}^* \to \mathcal{R}$ be independent random oracles, where H_χ samples from an efficiently samplable RLWE error distribution χ with $\|\eta\|_\infty \le B_\chi$ for a public bound $B_\chi \ge 1$. The canonical encoding of a ciphertext includes its modulus, scale, and message parameter p. For party i, define $\eta_i = H_\chi((s_i \bmod q, z_i), \mathsf{ct})$, so that repeated queries for the same ciphertext and party use the same η_i, as in KLLPS.

We set $c = n$ and use c^2 and c as the decryption and flooding factors, respectively. This choice clears the denominators required for reconstruction and simulation. For each partial decryption, party i samples a fresh polynomial $g_i \in \mathcal{R}$ with independent coefficients uniform in $\{-B_{\mathrm{sm}}, \dots, B_{\mathrm{sm}}\}$, where $B_{\mathrm{sm}} \ge 1$ is a public integer, and returns

$$d_i = c^2\big(a(s_i \bmod q) + H(\mathsf{ct})z_i + \eta_i\big) + c\, g_i \pmod q. \tag{14}$$

The flooding samples are independent across parties and invocations. Since partial decryption does not require the responding set, the parties can generate their responses without knowing which other parties will participate in reconstruction.

Let $S \subseteq [n]$ be a responding set of size t, and let $\lambda_i^S \in \mathcal{R}_q$ denote the Lagrange coefficient for interpolation at zero. The chosen interpolation points give integer polynomials $\omega_{S,i} = \prod_{u \in [n]\setminus S}(1 - \alpha_i \alpha_u^{-1}) \in \mathcal{R}$ satisfying $\omega_{S,i} \equiv c\lambda_i^S \pmod q$. Since $\sum_{i\in S} \lambda_i^S z_i = 0$, the masks cancel in reconstruction. Thus, for the centered representative Z of $b + as \bmod q$, we obtain

$$c^2 b + \sum_{i\in S} \lambda_i^S d_i \equiv c^2 Z + \sum_{i\in S} \omega_{S,i}(c\eta_i + g_i) \pmod q. \tag{15}$$

To recover $\mathbf{m}$, final decryption decodes the centered representative of the reconstructed polynomial at scale Δ' and rounds each real and imaginary coordinate to the nearest integer. Let $\mathbf{y} \in \mathbb{Z}^{N/2}$ denote the rounded real coordinates. The algorithm returns $\perp$ unless every rounded imaginary coordinate is zero and every coordinate of $\mathbf{y}$ belongs to $c^2\{-(p-1), -(p-3), \dots, p-1\}$. Otherwise, final decryption returns

$$\frac{1}{2}\left(\frac{\mathbf{y}}{c^2} + p - 1\right) \in \{0, \dots, p-1\}^{N/2}.$$

4.2 Correctness and Strong Simulation Security

Let $M = \sigma^{-1}(\mathbf{v})$. The centered polynomial Z from Section 4.1 satisfies

$$\begin{aligned} Z &= \Delta' M + e_a + u, \\ \|\sigma(e_a)\|_\infty \le \Delta'\epsilon, &\qquad \|u\|_\infty \le U, \end{aligned} \tag{16}$$

where $e_a, u \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{R}$ and $\epsilon, U \ge 0$ are public error bounds. Before the additional output rescaling, the bound of Section 3.4 gives $\epsilon = E_v$ and $U = 0$. After rescaling, ϵ is unchanged, while Δ' and U follow (13) and include the additional rounding error.

We use the strong simulation experiment of KLLPS [15] with a static corrupted set $C \subseteq [n]$, $|C| < t$. The adversary may adaptively request partial decryptions of ciphertexts generated by the prescribed encryption and evaluation algorithms. For each ciphertext, the simulator receives the exact integer output only when the union of the corrupted and responding parties contains at least t distinct indices. Repeated queries do not advance this count, but every response query contributes to the public bound q_{pd}.

Once the output is available, the simulator replaces Z with the rounded encoding $m_0 = \lfloor \Delta' M \rceil$. The resulting error $e = Z - m_0 \in \mathcal{R}$ then satisfies $\|e\|_\infty \le \Delta'\epsilon + U + 1/2$. The difference between real and simulated responses depends on this error, but also on the queried party's RLWE error and those of the $t-1$ parties whose deterministic components have already been fixed. The flooding distribution must therefore hide their combined contribution.

To bound the response difference, let T be a set of $t-1$ parties and let ℓ_j denote the Lagrange basis polynomial at α_j for $\{0\} \cup \{\alpha_j : j \in T\}$. Reconstruction uses the uniform bound $\|\omega_{S,i}\|_1 \le B_2$ for every set S of size t and $i \in S$, whereas simulation uses $\|c\ell_j(\alpha_i)\|_1 \le B_3$ for every $j \in T$ and $i \notin T$. Appendix C derives an integer representative for the latter bound. With $K = NtB_2$, the flooding contribution in the canonical embedding is at most KB_{sm}, and the RLWE contribution is at most KcB_χ. The chosen interpolation points admit

$$\begin{aligned} &B_2 = 2^{n-t}, \qquad B_3 = 2^{n-1}, \qquad K = NtB_2, \\ &D_* \ge N\big[c2^{t-1}(\Delta'\epsilon + U + 1/2) + (c + (t-1)B_3)B_\chi\big]. \end{aligned} \tag{17}$$

Here D_* bounds the coefficient ℓ_1 norm of the response difference after division by the flooding factor c, as shown in Appendix C. To use the same flooding distribution throughout the experiment, we choose D_* satisfying (17) for every scale and error bound permitted at the fixed modulus q.

Let $\delta \in (0,1)$ bound the statistical distance accumulated from noise flooding over all response queries, and let $\nu \geq 0$ bound the numerical decoding error in each complex slot. We choose one integer flooding bound $B_{\mathrm{sm}} \geq 1$ satisfying, for every permitted configuration,

$$c^2\Delta'(p-1+\epsilon) + c^2U + tB_2(cB_\chi + B_{\mathrm{sm}}) < q/2, \tag{18}$$

$$c^2\epsilon + \frac{c^2NU + K(cB_\chi + B_{\mathrm{sm}})}{\Delta'} + \nu < 1/2, \tag{19}$$

$$\frac{q_{\mathrm{pd}}D_*}{2B_{\mathrm{sm}}+1} \leq \delta. \tag{20}$$

The first two conditions limit the flooding noise to preserve exact recovery. Condition (18) prevents modular wraparound during reconstruction, and (19) then ensures that the real coordinates round to $c^2\mathbf{v}$ and the imaginary coordinates round to zero. Final decryption therefore returns $\mathbf{m}$. In contrast, (20) requires enough flooding noise to bound the statistical distance over all response queries.

Theorem 4.1 (Exact recovery and strong simulation security). *Assume that the stated error bounds hold uniformly throughout the response experiment, except with total probability at most ϵ_{bad}, and that (18)–(20) hold for every permitted configuration with the same B_{sm}. For any responding set of t distinct parties, final decryption applied to responses generated as in Section 4.1 recovers $\mathbf{m}$ exactly, except with probability at most ϵ_{bad}.*

Assume the decisional $\mathrm{RLWE}_{q,\chi}$ assumption used by KLLPS and IND-CPA security of the underlying encryption scheme with the required public evaluation keys. In the random oracle model, there exists a polynomial time simulator for the strong simulation experiment above. For every probabilistic polynomial time adversary, the distinguishing advantage between the real and ideal experiments is at most $\delta + \epsilon_{\mathrm{bad}}$ plus a negligible term from the KLLPS RLWE hybrids. If δ and ϵ_{bad} are negligible, the construction is strongly simulation secure against static corruption of fewer than t parties.

For fixed modulus, scale, and coefficient rounding bound, a smaller ϵ reduces the ciphertext contribution to both the required flooding and the reconstructed error. Noise cleaning therefore relaxes both constraints on the flooding distribution. Appendix C proves the theorem by adapting the KLLPS simulator to m_0 and bounding the complete reconstruction error.

4.3 Reducing the Decryption Modulus

The modulus needed during high precision cleaning need not be retained for threshold decryption. To reduce response sizes, we select the output rescaling of Section 3.4 together with the flooding parameter. Let $Q_0 \mid Q$ be the modulus before rescaling. For a sequence satisfying (12), the resulting parameters are $q = Q_0/d$, $\Delta' = \Delta/d$, and $U = U_0/d + \rho_c$, as in (13).

The integer interpolation bounds remain valid at q, so we choose

$$D_* \geq N \left[c2^{t-1} \left(\frac{\Delta\epsilon + U_0}{d} + \rho_c + \frac{1}{2} \right) + (c + (t-1)B_3)B_\chi \right],$$
$$B_{\mathrm{sm}} = \max \left\{ 1, \left\lceil \frac{q_{\mathrm{pd}} D_* / \delta - 1}{2} \right\rceil \right\}.$$

The RLWE errors are introduced after rescaling, during partial decryption, so their contribution is not divided by d. When several rescaling sequences are permitted at the same q, the first bound must hold for every sequence. The second expression gives the smallest integer $B_{\mathrm{sm}} \geq 1$ satisfying (20). Since both correctness bounds increase with B_{sm}, a flooding parameter satisfies all three conditions if and only if this smallest choice satisfies (18) and (19) for every permitted sequence. The selected pair (q, χ) must also satisfy the RLWE assumption of Theorem 4.1.

The modulus q is fixed at setup, before the dealer generates the zero shares. Since $q \mid Q$, reduction of the existing CKKS sharing gives $s_i \bmod q = (f \bmod q)(\alpha_i)$, with constant term $s \bmod q$. The nonconstant coefficients remain independent and uniform over $\mathcal{R}_q$, so the parties use their locally reduced CKKS shares together with the zero shares at q. Each choice of q defines a separate instantiation of the strong construction.

Each party stores one additional element of $\mathcal{R}_q$ for its zero share, while a partial decryption remains one element of $\mathcal{R}_q$. With fixed length coefficient encoding, a response requires $N\lceil \log_2 q \rceil$ bits. Relative to responses at modulus Q_0, the t responses used for reconstruction therefore save

$$tN \left(\lceil \log_2 Q_0 \rceil - \lceil \log_2 q \rceil \right)$$

bits.

Remark 4.2 (Known response sets). The construction also applies when the responding set is known before partial decryption, with the same algorithms and the guarantees of Theorem 4.1.

5 Evaluation

5.1 Experimental Setup

We implement binary and integer noise cleaning in OpenFHE [1] and use Sparse-THI [3] for functional bootstrapping and digit extraction. The experiments run on a single thread of an Intel Xeon Platinum 8575C processor with approximately 185 GiB of RAM, using ring dimension $N = 65536$ and all 32768 complex slots. The main secret follows the uniform ternary distribution. The moduli, including the special primes for key switching, satisfy OpenFHE's parameter tables for 128 bit classical security under this distribution. Functional bootstrapping additionally uses a secret of Hamming weight 64, for which these table checks do not provide

an independent security estimate. Leveled extraction uses no sparse auxiliary secret.

The binary experiments compare single Hermite evaluations with compositions on the same messages, at a common initial scale and input error bound. For integer messages, we compare three methods for evaluating a fixed cleaning composition. Within each extraction route, the methods share the extraction procedure, modulus chain, and rescaling schedule, but use separate key switching parameters. Appendix D gives the parameters and complete error bounds.

5.2 Noise Cleaning for Bits

We compare the compositions of Section 3.2 with single Hermite evaluations at normalized output noise targets of 2^{-128}, 2^{-200}, and 2^{-300}, starting from a signed input error bound of 2^{-16}. Table 2 pairs each single evaluation with the fastest composition among those evaluated, with fully delayed evaluation in both cases. The output bounds include key switching and rescaling errors and satisfy the respective targets under the input and evaluation key error assumptions in Appendix D.4.

Table 2. Binary noise cleaning at three target precisions. Evaluations are listed in execution order. Normalized precision in bits is $-\log_2(\max_\ell |\widehat{\zeta}_\ell - \zeta_\ell|/4)$ for signed bits $\zeta_\ell \in \{-1, 1\}$.

Target	Evaluations	Normalized precision		$\log_2 Q_0$	Time	Key storage
(bits)		Lower bound	Measured		(s)	(MiB)
128	R_8	139.43	140.90	718.02	1.000	400
	$R_2 \to R_2$	140.71	142.18	618.02	0.501	188
200	R_{13}	209.25	217.03	1120.02	2.762	962
	$R_2 \to R_2 \to R_1$	203.49	227.86	782.02	1.020	286
300	R_{19}	305.00	308.26	1620.52	9.632	3496
	$R_1 \to R_1 \to R_1 \to R_2$	331.19	346.72	1081.52	1.540	556

The selected compositions reduce running time by factors of 1.99, 2.71, and 6.26 relative to the corresponding single evaluations, while requiring smaller input moduli and less evaluation key storage. The larger speedup at the 300 bit target reflects the cost of evaluating a high degree polynomial in a single step. Direct evaluation of R_{19} produces a tensor of degree 39, whereas the selected composition keeps the tensor degree at most five. Moreover, rescaling between successive evaluations allows the later polynomials to be evaluated at smaller moduli.

At the 300 bit target, three evaluations of R_1 followed by R_2 take 1.540 seconds, compared with 2.519 seconds for five successive evaluations of R_1. The

final quintic therefore reduces the number of rounds, the input modulus, and key storage in this comparison. At the 200 bit target, four cubic evaluations instead offer a storage saving: they require 18.2 percent less key storage than $R_2 \to R_2 \to R_1$, with a 3.4 percent increase in running time and the same input modulus. Appendix D.4 gives the additional compositions and their parameters.

5.3 Noise Cleaning for Integer Messages

The modulus remaining after homomorphic computation determines whether digit extraction can proceed without bootstrapping. Leveled extraction uses the remaining levels, whereas functional bootstrapping accepts a low level input and raises the modulus. Both routes extract digits from the same vector covering all 256 possible 8 bit values. The subsequent computation is common to both routes: after conjugation, we apply $R_2 \circ R_1 \circ R_1 \circ R_1$ to each digit and recombine the results as in Section 3.4.

For the cubic rounds, *ordinary evaluation* relinearizes intermediate products, whereas *fully delayed evaluation* relinearizes after polynomial evaluation. Both evaluate the final quintic directly as a ciphertext tensor. To examine the cost of the auxiliary representation, *auxiliary evaluation* retains the ordinary cubic rounds and replaces the final quintic evaluation by $P_2(x, T)$ from Section 3.3.

All three methods exceed 300 bits of measured normalized precision in both routes (Table 3). Fully delayed evaluation is the fastest: the complete computation takes 60.80 seconds with leveled extraction and 178.22 seconds with functional bootstrapping, with measured precisions of 317.14 and 308.05 bits, respectively. The conditional error bounds for these parameters are given separately in Appendix D.

Table 3. Precision and cost of noise cleaning for 32768 integers. Measured normalized precision is $-\log_2(\max_\ell |\widehat{m}_\ell - m_\ell|/256)$. Cleaning includes recombination, and total time additionally includes digit extraction.

Evaluation	Measured precision (bits)	Time (s)		Cleaning keys (MiB)
		Cleaning	Total	
Leveled extraction				
Ordinary	315.99	10.71	63.00	246
Fully delayed	317.14	8.51	60.80	452
Auxiliary	316.66	11.21	63.50	212
Functional bootstrapping				
Ordinary	308.36	9.24	179.56	246
Fully delayed	308.05	7.90	178.22	444
Auxiliary	307.92	10.51	180.83	210

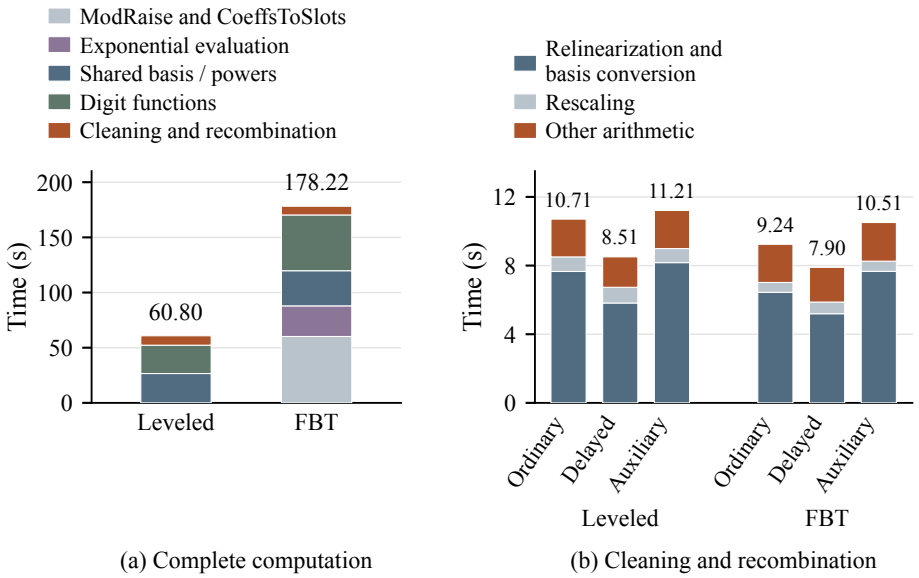

Fig. 1. Running time breakdown for 8 bit integer noise cleaning. Panel (a) compares the complete computations with fully delayed evaluation. Panel (b) separates the cleaning and recombination costs for all three evaluation methods. FBT denotes functional bootstrapping, and Delayed denotes fully delayed evaluation. The panels use different vertical scales.

Digit extraction dominates the complete computation, accounting for 86.0 percent of leveled evaluation and 95.6 percent of functional bootstrapping with fully delayed cleaning (Figure 1(a)). The eight digit functions share ciphertext powers and, in functional bootstrapping, the transformations and exponential evaluation. In the leveled route, preparing the shared polynomial basis takes 26.60 seconds, nearly as much as evaluating the eight digit functions together (25.68 seconds). Functional bootstrapping incurs additional costs for modulus raising and the coefficient to slot transformation (60.17 seconds) and exponential evaluation (27.76 seconds). The subsequent preparation of shared powers and evaluation of the digit functions take 31.85 and 50.54 seconds, respectively.

Relinearization and basis conversion account for 65.8–72.9 percent of cleaning and recombination time (Figure 1(b)). Fully delayed evaluation reduces these costs, bringing cleaning and recombination to 8.51 and 7.90 seconds, but requires 452 and 444 MiB of cleaning keys. The reduction in total time is smaller because the digit extraction procedure is unchanged.

Auxiliary evaluation reduces key storage relative to ordinary evaluation, from 246 to 212 MiB for leveled extraction and from 246 to 210 MiB for functional bootstrapping, saving 13.8 and 14.6 percent. The same keys for s^2 and s^3 serve

both auxiliary generation and final relinearization, replacing the final switching keys for $s^2, \ldots, s^5$. However, each digit requires an auxiliary cube, and the final switches use a larger special modulus. As a result, auxiliary evaluation saves key storage but takes longer than ordinary evaluation. Appendix D.2 gives the switching parameters and operation times.

6 Conclusion

We presented a high precision noise cleaning method for CKKS that preserves encrypted integer messages while meeting a prescribed bound on normalized output noise. The complete error analysis supports joint selection of the cleaning composition, scales, and key switching parameters. For binary messages, the selected compositions reduce running time, input ciphertext modulus, and key storage relative to single Hermite evaluations at the same output noise targets. The complete integer construction attains measured normalized precision above 300 bits in all 32768 slots with either leveled digit extraction or functional bootstrapping. The auxiliary representations lower tensor degree while cancelling selected auxiliary error terms and reduce cleaning key storage in both integer routes. Fully delayed evaluation instead achieves the shortest measured running time with larger cleaning keys. For the asynchronous threshold application, we use the integer error bound to derive sufficient conditions on the flooding distribution for exact integer recovery and strong simulation security in the random oracle model.

References

1. Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Quah, I., Polyakov, Y., R.V., S., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., Zucca, V.: OpenFHE: Open-source fully homomorphic encryption library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. pp. 53–63 (2022). https://doi.org/10.1145/3560827.3563379
2. Alexandru, A., Kim, A., Polyakov, Y.: General functional bootstrapping using CKKS. In: Advances in Cryptology – CRYPTO 2025. pp. 304–337 (2025). https://doi.org/10.1007/978-3-032-01881-6_10, https://eprint.iacr.org/2024/1623
3. Alexandru, A., Kim, A., Polyakov, Y., Zheng, H.: Sparse Hermite interpolation method for discrete-CKKS functional bootstrapping. Cryptology ePrint Archive, Paper 2026/1026 (2026), https://eprint.iacr.org/2026/1026
4. Bae, Y., Cheon, J.H., Cho, W., Kim, J., Kim, T.: META-BTS: Bootstrapping precision beyond the limit. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 223–234 (2022). https://doi.org/10.1145/3548606.3560696, https://eprint.iacr.org/2022/1167
5. Bae, Y., Cheon, J.H., Kim, J., Stehlé, D.: Bootstrapping bits with CKKS. In: Advances in Cryptology – EUROCRYPT 2024. pp. 94–123 (2024). https://doi.org/10.1007/978-3-031-58723-8_4, https://eprint.iacr.org/2024/767

6. Bae, Y., Kim, J., Stehlé, D., Suvanto, E.: Bootstrapping small integers with CKKS. In: Advances in Cryptology – ASIACRYPT 2024. pp. 330–360 (2024). https://doi.org/10.1007/978-981-96-0875-1_11, https://eprint.iacr.org/2024/1637
7. Bian, S., Fu, Y., Shen, R., Pan, H., Wang, A., Guan, Z.: High-precision functional bootstrapping for CKKS from Fourier extension. In: Advances in Cryptology – EUROCRYPT 2026. pp. 274–303 (2026). https://doi.org/10.1007/978-3-032-25327-9_10
8. Cheon, J.H., Cho, W., Kim, J., Stehlé, D.: Homomorphic multiple precision multiplication for CKKS and reduced modulus consumption. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 696–710 (2023). https://doi.org/10.1145/3576915.3623086, https://eprint.iacr.org/2023/1788
9. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology – ASIACRYPT 2017. pp. 409–437 (2017). https://doi.org/10.1007/978-3-319-70694-8_15, https://eprint.iacr.org/2016/421
10. Cheon, J.H., Lee, J.: High-precision CKKS with focal units. Cryptology ePrint Archive, Paper 2026/2531 (2026), https://eprint.iacr.org/2026/2531
11. Choe, H., Kim, J., Stehlé, D., Suvanto, E.: Leveraging discrete CKKS to bootstrap in high precision. In: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security. pp. 1083–1097 (2025). https://doi.org/10.1145/3719027.3765080
12. Drucker, N., Moshkowich, G., Pelleg, T., Shaul, H.: BLEACH: Cleaning errors in discrete computations over CKKS. Journal of Cryptology **37** (2024). https://doi.org/10.1007/s00145-023-09483-1, https://eprint.iacr.org/2022/1298, article 3
13. Hwang, I., Hwang, Y., Kim, M., Lee, D., Song, Y.: Provably secure approximate computation protocols from CKKS. Cryptology ePrint Archive, Paper 2025/395 (2025), https://eprint.iacr.org/2025/395
14. Kim, W., Lee, C., Lee, J., Passelègue, A., Stehlé, D.: Asynchronous Lagrange-based threshold FHE with smaller modulus overhead. In: Advances in Cryptology – CRYPTO 2026. pp. 3–35 (2026), https://eprint.iacr.org/2026/973
15. Kim, W., Lee, C., Lee, J., Passelègue, A., Stehlé, D.: Asynchronous Shamir-based threshold FHE: Smaller moduli and strong simulation security. Cryptology ePrint Archive, Paper 2026/973 (2026), https://eprint.iacr.org/2026/973, extended version, revised September 17, 2026
16. Lee, Y., Lee, J.W., Kim, Y.S., Kim, Y., No, J.S., Kang, H.: High-precision bootstrapping for approximate homomorphic encryption by error variance minimization. In: Advances in Cryptology – EUROCRYPT 2022. pp. 551–580 (2022). https://doi.org/10.1007/978-3-031-06944-4_19, https://eprint.iacr.org/2020/1549
17. Ma, S., Liu, J., Wang, A., Wang, X.: High-precision bootstrapping for GCKKS. Cryptology ePrint Archive, Paper 2026/2600 (2026), https://eprint.iacr.org/2026/2600
18. Ma, S., Liu, J., Wang, A., Wang, X.: Homomorphic encryption for high-precision arithmetic of approximate numbers. Cryptology ePrint Archive, Paper 2026/2524 (2026), https://eprint.iacr.org/2026/2524
19. Mouchet, C., Bertrand, E., Hubaux, J.P.: An efficient threshold access-structure for RLWE-based multiparty homomorphic encryption. Journal of Cryptology **36**(2) (2023). https://doi.org/10.1007/s00145-023-09452-8, https://eprint.iacr.org/2022/780, article 10

20. Mouchet, C., Troncoso-Pastoriza, J.R., Bossuat, J.P., Hubaux, J.P.: Multiparty homomorphic encryption from ring-learning-with-errors. Proceedings on Privacy Enhancing Technologies **2021**(4), 291–311 (2021). https://doi.org/10.2478/popets-2021-0071, https://petsymposium.org/popets/2021/popets-2021-0071.pdf
21. The MPFR Team: GNU MPFR 4.2.1 Manual (2023), https://www.mpfr.org/mpfr-4.2.1/mpfr.html
22. Colin de Verdière, F., Passelègue, A., Stehlé, D.: On threshold fully homomorphic encryption with synchronized decryptors. Cryptology ePrint Archive, Paper 2026/031 (2026), https://eprint.iacr.org/2026/031

A Polynomial Representations and Error Bounds

The bounds in this appendix hold for complex perturbations, as required for the slot errors in Sections 3.1 and 3.3.

A.1 Hermite Residual and Coefficient Denominator

The derivative of (1) is $c_r(1-x^2)^r$. Since $1-(1+h)^2 = -h(2+h)$, integration from 0 to h gives

$$R_r(1+h) - 1 = (-1)^r c_r \sum_{j=0}^{r} \binom{r}{j} \frac{2^{r-j} h^{r+j+1}}{r+j+1}.$$

By odd symmetry, the expansion at -1 has the same coefficient magnitudes. The triangle inequality therefore proves (2). The normalization $R_r(1) = 1$ follows either from the beta integral $c_r \int_0^1 (1-x^2)^r \, dx = 1$ or by evaluating the finite sum. Thus the polynomial satisfies the Hermite interpolation conditions exactly.

An equivalent polynomial expression is

$$R_r(x) = x \sum_{j=0}^{r} \frac{(-1)^j \binom{2j}{j}}{4^j} (x^2-1)^j. \tag{21}$$

To establish (21), consider the truncated series for $(1+u)^{-1/2}$ at $u = x^2 - 1$. The product with x equals $1 + O((x-1)^{r+1})$ near 1. Odd symmetry gives the corresponding conditions at -1. The resulting polynomial has degree at most $2r+1$ and agrees with R_r in $2r+2$ Hermite conditions, which uniquely determine it.

Let $v_2(a)$ denote the exponent of two in a nonzero integer a. For a positive integer j, the identity $v_2\binom{2j}{j} = s_2(j)$ follows from $v_2(j!) = j - s_2(j)$. The reduced denominator of the jth coefficient in (21) is therefore $2^{2j-s_2(j)}$. If j has a trailing binary ones, the next exponent increases by $1+a$. Consequently, every denominator divides $L_r = 2^{2r-s_2(r)}$. The leading coefficient has exactly this reduced denominator, which proves that L_r is the least common denominator of the coefficients of R_r.

A.2 Minimum Degree and the Explicit Family

The correction in Theorem 3.3 uses the coefficients

$$\beta_\ell = \sum_{\substack{3\le m\le 2r+1\\ m \text{ odd}}} a_m \left\lfloor \frac{m}{3} \right\rfloor \binom{(m-3)/2}{\ell}, \qquad 0 \le \ell < r. \tag{22}$$

Indeed, the contribution of a_m to $\partial_T N_r(x, x^3)$ is $a_m \lfloor m/3 \rfloor x^{m-3}$. Expansion in powers of $x^2 - 1$ then gives (22).

Proof (Theorem 3.3). Assign weight one to x and weight three to T. Under substitution $T = x^3$, the monomial $x^i T^j$ contributes only to weight $i + 3j$. For a fixed odd m, a nonzero coefficient of this degree requires at least one monomial with $i+3j = m$. The minimum total degree is attained at $j = \lfloor m/3 \rfloor$, $i = m \bmod 3$, and equals $\lfloor m/3 \rfloor + (m \bmod 3)$. The maximum over $m = 1, 3, \ldots, 2r + 1$ is $2\lfloor (r+1)/3 \rfloor + 1$. Since terms of other weights cannot contribute to the coefficient of x^m, this maximum is a lower bound on the total degree. The representation N_r attains the bound by using a minimum degree monomial for every weight.

With β_ℓ from (22), let $K(x) = \sum_{\ell=0}^{\nu-1} \beta_\ell (x^2-1)^\ell$. Then $(T - x^3)K$ has total degree at most $2\nu + 1 = D_r$ and weighted degree at most $2\nu + 1 \le 2r + 1$. The correction vanishes after substitution, so $P_r(x, x^3) = R_r(x)$ and P_r retains the minimum degree. Moreover, differentiation gives $\partial_T P_r(x, x^3) = \sum_{\ell=\nu}^{r-1} \beta_\ell (x^2-1)^\ell$.

All coefficients in this construction are integer linear combinations of the a_m. Hence $L_r P_r$ has integer coefficients. Any integer clearing the coefficient denominators of P_r also clears those of $P_r(x, x^3) = R_r(x)$, so L_r remains the least common denominator.

To bound the auxiliary error, define $G_j(x) = \sum_{m=3j}^{2r+1} a_m \binom{\lfloor m/3 \rfloor}{j} x^{m-3j}$ for $j \ge 2$. Since the correction is linear in $T - x^3$, it changes only the term linear in the auxiliary error. Hence

$$P_r(x, x^3 + \eta) - R_r(x) = \eta \sum_{\ell=\nu}^{r-1} \beta_\ell (x^2-1)^\ell + \sum_{j=2}^{\lfloor (2r+1)/3 \rfloor} G_j(x) \eta^j.$$

At $x = \zeta + h$, $|x| \le 1 + B$ and $|x^2 - 1| \le 2B + B^2$. The triangle inequality therefore gives the explicit bound

$$\varepsilon_r(B) + \theta \sum_{\ell=\nu}^{r-1} |\beta_\ell| (2B + B^2)^\ell + \sum_{j=2}^{\lfloor (2r+1)/3 \rfloor} \theta^j \sum_{m=3j}^{2r+1} |a_m| \binom{\lfloor m/3 \rfloor}{j} (1+B)^{m-3j}.$$

An empty sum contributes zero. The coefficient bound in (4) may be tighter because it collects equal powers of h and η before applying the triangle inequality. For fixed r, the linear auxiliary contribution is $O(B^\nu\theta)$, the higher powers contribute $O(\theta^2)$, and the Hermite residual is $O(B^{r+1})$. This proves the stated uniform error bound. □

Table 4. Representations from Theorem 3.3. ν is the guaranteed vanishing order of the linear auxiliary error.

r	$\deg R_r$	D_r	ν	L_r	Terms
2	5	3	1	8	4
3	7	3	1	16	5
4	9	3	1	128	6
5	11	5	2	256	8
6	13	5	2	1024	9
8	17	7	3	32768	13

A.3 Contraction Conditions and Lower Bounds

The coefficient criterion extends to any finite message set $\mathcal{A}$, polynomial F satisfying $F(a) = f(a)$ for $a \in \mathcal{A}$, and auxiliary polynomial A. For a representation satisfying $P(x, A(x)) = F(x)$, put $G_j(x) = \partial_T^j P(x, A(x))/j!$. Taylor expansion in the auxiliary error gives the exact identity

$$P(a+h, A(a+h)+\eta) - f(a) = F(a+h) - f(a) + \sum_{j\geq 1} G_j(a+h)\eta^j. \tag{23}$$

Let k_a be the order of $F(a+h) - f(a)$ and $\nu_{a,j}$ the order of $G_j(a+h)$ at $h = 0$, with order infinity for a zero polynomial. For fixed $c, \alpha > 0$ and $|\eta| \leq cB^\alpha$, the worst case error has order

$$\min_{a\in\mathcal{A}} \min\{k_a, \nu_{a,1} + \alpha, \nu_{a,2} + 2\alpha, \ldots\}. \tag{24}$$

For a fixed representation P, let $e_{a,u,v}$ denote the coefficient of $h^u\eta^v$ in the error polynomial at a, and let $E(P; B, \theta)$ be the maximum absolute error over all $a \in \mathcal{A}$ and the two complex discs. The coefficient bound gives an upper bound of order B^τ, where τ is the minimum of $u + \alpha v$ over the nonzero coefficients. Conversely, Cauchy's coefficient estimate gives

$$E(P; B, cB^\alpha) \geq |e_{a,u,v}| c^v B^{u+\alpha v}.$$

A nonzero coefficient attaining the minimum gives the matching lower bound. An identically zero error polynomial has infinite order. The expansion in (23) identifies the minimum with (24). For the cubic auxiliary and the binary Hermite

function, this proves (5). The estimate is uniform over both perturbations and requires no independence assumption.

For the cubic model, fix a finite set of monomials $\mathcal{I}$ and write $P = \sum_{(i,j)\in\mathcal{I}} p_{ij} x^i T^j$. The identity $P(x, x^3) = R_r(x)$ gives the equations

$$\sum_{i+3j=m} p_{ij} = a_m \quad \text{for every weight represented in } \mathcal{I},$$

with $a_m = 0$ outside R_r. A missing nonzero coefficient makes the system infeasible. Equations with $a_m = 0$ also allow cancellation between monomials of weights higher than $2r + 1$. For a polynomial F, write $[h^u \eta^v]F$ for its coefficient of $h^u \eta^v$. Expansion of each monomial after the substitution $x = 1 + h$, $T = (1 + h)^3 + \eta$ gives

$$[h^u \eta^v] P(1 + h, (1 + h)^3 + \eta) = \sum_{i,j} p_{ij} \binom{j}{v} \binom{i + 3(j - v)}{u}.$$

After subtracting the constant message value for $(u, v) = (0, 0)$, we obtain the error coefficients as affine functions of the representation coefficients. The function identity and the vanishing conditions of weighted order less than τ therefore form a finite rational linear system. Every solution has error $O(B^\tau)$. Conversely, Cauchy's coefficient estimate on the complex bidisc shows that a nonzero lower order coefficient prevents this error bound for a fixed representation.

The same coefficient estimates also give a uniform lower bound when the vanishing constraints are inconsistent. Elimination of the unknown p_{ij} produces an affine identity $\sum_{u,v} \lambda_{uv} e_{uv} = 1$, where e_{uv} are coefficients of the error polynomial. If the supremum of the absolute error is E, then $|e_{uv}| \le E B^{-u} \theta^{-v}$, and hence

$$E \ge \left(\sum_{u,v} |\lambda_{uv}| B^{-u} \theta^{-v} \right)^{-1}.$$

If the equations at weighted order τ are the first to make the system inconsistent, the bound is $\Omega(B^\tau)$, while the preceding consistent system gives a representation with error $O(B^\tau)$. The fixed input residual also limits the order to $r + 1$. Odd symmetrization $P_o(x, T) = (P(x, T) - P(-x, -T))/2$ preserves $P(x, x^3) = R_r(x)$ and does not increase the worst case error over both message values. Thus odd monomials suffice when the support constraint is a total degree bound.

For rational B, θ and rational nodes, minimizing the coefficient sum in (4) over the fixed rational affine space is a linear program, with auxiliary variables for the absolute values. If at most J error monomials occur at any node, the coefficient bound and the maximum error, with both maxima taken over $\mathcal{A}$, satisfy

$$\frac{\varepsilon(P; B, \theta)}{\sqrt{J}} \le E(P; B, \theta) \le \varepsilon(P; B, \theta).$$

Indeed, orthogonality on the boundary torus identifies the weighted coefficient ℓ_2 norm with the mean square norm of the polynomial. The supremum bounds

this norm, and Cauchy–Schwarz bounds the coefficient ℓ_1 norm. A prescribed common denominator imposes additional integrality constraints not included in this rational relaxation.

The degree nine function.

Proof (Proposition 3.4, R_4 case). The odd part of a representation preserves the function identity and does not increase the worst case error over the two message values. At total degree at most four, its monomials are therefore among $x, T, x^3, x^2T, xT^2, T^3$. The function identity fixes all coefficients except the split between x^3 and T, so every such odd representation is $P_4 + \mu(T - x^3)$ for some $\mu \in \mathbb{Q}$. The error coefficients of h^5, $h\eta$ and η^2 at 1 are $63/8$, $-27/64$ and $-75/128$, respectively, independently of μ. Cauchy's estimate gives

$$E(P; B, \theta) \geq \max\left\{\frac{63}{8}B^5, \frac{27}{64}B\theta, \frac{75}{128}\theta^2\right\}. \tag{25}$$

For P_4, (3) and the Hermite residual give the matching upper order $O(B^5 + B\theta + \theta^2)$. Substitution of $\theta = cB^\alpha$ proves the claim. The coefficient lower bound also applies when the representation is selected as a function of B. □

To obtain a finite error bound, we retain the higher order terms as well. Let $A(B)$ and $C(B)$ be the coefficient sums of the linear and quadratic auxiliary terms, respectively. Expansion of (3) at either message value gives

$$\begin{aligned}\varepsilon(P_4; B, \theta) &= \varepsilon_4(B) + \frac{A(B)\theta + C(B)\theta^2 + 35\theta^3}{128},\\ A(B) &= 54B + 207B^2 + 660B^3 + 1215B^4 + 630B^5 + 105B^6,\\ C(B) &= 75 + 135B + 315B^2 + 105B^3.\end{aligned}$$

Thus $E(P_4; B, \theta) \leq \varepsilon(P_4; B, \theta)$ for complex $|h| \leq B$ and $|\eta| \leq \theta$.

As the auxiliary error decreases, the bound approaches the Hermite residual (Figure 2).

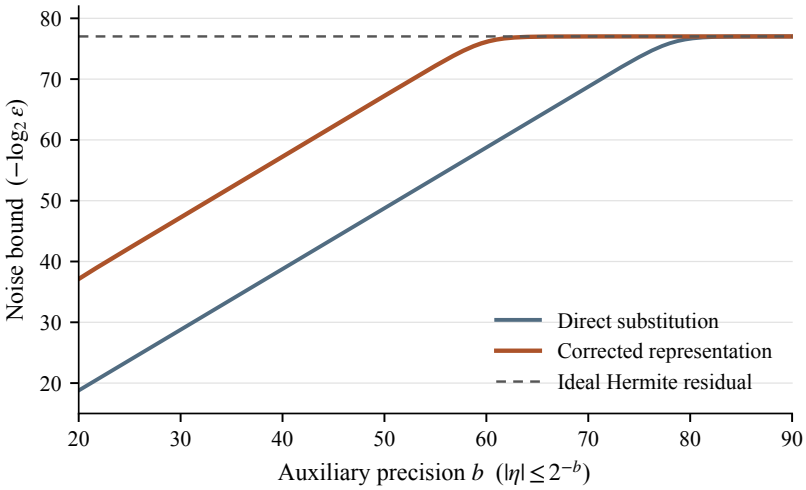

Fig. 2. Explicit polynomial error bounds for R_4 at $B = 2^{-16}$, as a function of the auxiliary accuracy $\theta = 2^{-b}$. The correction reduces the auxiliary error contribution until the input residual dominates. Final CKKS key switching and rescaling are treated in Section 3.3.

The degree thirteen function.

Proof (Proposition 3.4, R_6 case). For $a, b \in \mathbb{Q}$, define $P_{a,b}$ by

$$\begin{aligned} 1024P_{a,b}(x,T) = {} & 231xT^4 - 1638x^2T^3 - bx^3T^2 - ax^4T \\ & + (4698 + 2a + 3b)x^5 + (5005 + b)T^3 \\ & + (a - 8580)xT^2 + (4311 - 2a - 3b)x^2T \\ & + (-7830 - a - 2b)x^3 + (1824 + a + 2b)T + 3003x. \end{aligned}$$

Substitution gives $P_{a,b}(x, x^3) = R_6(x)$ for every a, b. For $(a, b) = (6561, -4734)$, the polynomials $\partial_T^j P_{a,b}(x, x^3)$, $1 \le j \le 4$, have orders of vanishing $(3, 1, 0, 0)$ at both $x = 1$ and $x = -1$. For $(a, b) = (6156, -4599)$, the corresponding orders are $(4, 0, 0, 0)$. Differentiation and expansion at $x = \pm 1$ establish these orders. Both polynomials have degree five, eleven nonzero terms and least common coefficient denominator 1024. Equation (5) gives respective orders $\min\{7, 3 + \alpha\}$ and $\min\{7, 2\alpha\}$ for $\alpha \ge 2$, proving attainability.

For the lower bound, odd symmetrization reduces any degree at most six representation to degree at most five. Write its error coefficients at 1 as $e_{uv} = [h^u \eta^v](P(1+h, (1+h)^3 + \eta) - 1)$. Every polynomial in this space with $P(x, x^3) = R_6(x)$ satisfies

$$1 + \frac{64}{45}e_{11} - \frac{128}{45}e_{21} + \frac{128}{135}e_{31} + \frac{1024}{135}e_{02} = 0. \tag{26}$$

To prove the identity, put $F(x) = P(x, x^3)$ and $G_1(x) = \partial_T P(x, x^3)$. For every odd monomial $P = x^i T^j$ of total degree at most five, the expression

$$F - F' + \frac{3}{5}F'' - \frac{4}{15}F''' + \frac{2}{27}F^{(4)} - \frac{1}{135}F^{(5)} + \frac{1}{6075}F^{(6)} + \frac{64}{45}G_1' - \frac{64}{45}G_1'' + \frac{64}{405}G_1''' + \frac{512}{135}\partial_T^2 P$$

evaluated at $x = T = 1$ is zero. Linearity extends it to the entire space. The Hermite conditions for $F = R_6$ then give (26). Cauchy's estimate gives $|e_{uv}| \leq E(P; B, \theta)B^{-u}\theta^{-v}$, and hence

$$E(P; B, \theta) \geq \left(\frac{64}{45B\theta} + \frac{128}{45B^2\theta} + \frac{128}{135B^3\theta} + \frac{1024}{135\theta^2} \right)^{-1}.$$

Thus its contraction order is at most $\max\{3+\alpha, 2\alpha\}$. The nonzero input residual of R_6 also limits it to seven. Combined with the two constructions, these bounds prove the R_6 case of Proposition 3.4. They apply uniformly to representations selected as a function of B within the degree constraint. □

The two constructions improve different auxiliary terms relative to P_6: the first increases the vanishing order of the linear term while also making the quadratic term vanish at the messages, whereas the second further increases the linear order but retains a nonzero quadratic term. Table 5 gives the corresponding orders of $G_j(x) = \partial_T^j P(x, x^3)/j!$ at either message value.

Table 5. Auxiliary error for representations of R_6. All have total degree five, weighted degree thirteen and coefficient denominator 1024. The contraction orders hold for $\theta = cB^\alpha$ with fixed $c > 0$ and $\alpha \geq 2$.

Representation	Terms	Orders of $G_1, \ldots, G_4$	Contraction order
Explicit family P_6	9	$(2, 0, 0, 0)$	$\min\{7, 2+\alpha\}$
First construction	11	$(3, 1, 0, 0)$	$\min\{7, 3+\alpha\}$
Second construction	11	$(4, 0, 0, 0)$	$\min\{7, 2\alpha\}$

A.4 Auxiliary Polynomials Beyond the Cube

The general criterion (24) also applies to auxiliary powers other than a cube. The following construction cancels successive powers of the auxiliary error for every x. For $F(x) = \sum_m c_m x^m$ and auxiliary $T = x^e + \eta$, with integer $e \geq 1$, write $m = eq_m + b_m$, $0 \leq b_m < e$. For an integer $k \geq 0$, define

$$P^{[k]}(x, T) = \sum_m c_m x^{b_m} \sum_{j=0}^{\min(k, q_m)} \binom{q_m}{j} T^{q_m - j} (x^e - T)^j.$$

The inner sum retains the terms of degree at most k in $x^e - T$ from the expansion of $(T + (x^e - T))^{q_m}$. Thus $P^{[k]}(x, x^e) = F(x)$, and $P^{[k]}(x, x^e + \eta) - F(x)$ is divisible by η^{k+1}. The construction uses integer combinations of the c_m and preserves their least common denominator, by the same substitution argument as above. The total degree is at most $\max_m\{b_m + q_m + (e-1)\min(k, q_m)\}$. For inputs x near a finite message set, the local conditions in (24) can attain the required error order with a lower degree.

B Error Analysis for CKKS Evaluation

B.1 Integer Scales and Modulus Conditions

For positive integer input scales Δ, Γ, a monomial tensor for $x^i T^j$ has scale $\Delta^i \Gamma^j$. The integer weight in (7) therefore gives the contribution $S p_{ij} x^i T^j$ in the canonical embedding. The sum represents $SP(x, T)$, while all ciphertext operations are performed modulo the current modulus. Each weight is integral exactly when the reduced denominator of $p_{ij}/(\Delta^i \Gamma^j)$ divides S, proving the least common multiple rule.

For P_r at $\Gamma = \Delta^3$, the weighted degree bound and $L_r P_r \in \mathbb{Z}[x, T]$ make $L_r \Delta^{2r+1}$ admissible. If Δ is a power of two, the leading coefficient of R_r has odd numerator and denominator L_r. At least one coefficient of P_r with weight $2r+1$ must therefore have this reduced denominator. Otherwise, their sum could not equal the leading coefficient of R_r. Division of that coefficient by Δ^{2r+1} has reduced denominator $L_r \Delta^{2r+1}$, which proves minimality. For general integer Δ, the least common multiple rule remains exact.

Repeated rescaling. Let M be the modulus before a prescribed sequence of prime divisions $q_1, \dots, q_m$, with product $d \mid M$. Put $d_j = \prod_{i=1}^{j} q_i$ and $d_0 = 1$. Suppose that decryption of the input ciphertext gives $W \bmod M$, with $\|\sigma(W)\|_\infty \le C$. Let $\delta_j \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Q}$ be the decryption error introduced by rounding at the jth division, with $\|\sigma(\delta_j)\|_\infty \le \rho_j$. The division relation gives the following polynomial modulo M/d_j:

$$\begin{aligned} W_j &= \frac{W}{d_j} + \sum_{i=1}^{j} \frac{\delta_i}{q_{i+1} \cdots q_j}, \\ \rho &= \sum_{i=1}^{m} \frac{\rho_i}{q_{i+1} \cdots q_m}, \qquad d\rho = \sum_{i=1}^{m} d_i \rho_i. \end{aligned} \tag{27}$$

Empty products equal one. Equivalently, the recurrence $\rho \leftarrow \rho/q_j + \rho_j$, initialized at zero, gives the accumulated bound after the last division.

A sufficient condition for every W_j to have centered coefficients is $C + d\rho < M/2$. Indeed, for every $0 \le j \le m$,

$$d_j \|\sigma(W_j)\|_\infty \le C + \sum_{i=1}^{j} d_i \rho_i \le C + d\rho < M/2.$$

The coefficient norm is at most the canonical embedding norm, so $\|W_j\|_\infty < M/(2d_j)$. Hence W_j is the centered representative at modulus M/d_j, which justifies the division relation at every step of the induction. Since $d\rho$ bounds every partial sum, the condition also covers intermediate polynomials whose errors may cancel in the final sum. For the empty sequence, $d = 1$ and $\rho = 0$, so the condition reduces to $C < M/2$.

The same recurrence gives a coefficient bound. Suppose instead that $\|W\|_\infty \le C$ and $\|\delta_i\|_\infty \le \beta_i$, and put $\rho_c = \sum_{i=1}^{m} \beta_i/(q_{i+1} \cdots q_m)$. Then $d\rho_c = \sum_{i=1}^{m} d_i \beta_i$, and every prefix satisfies

$$d_j \|W_j\|_\infty \le C + \sum_{i=1}^{j} d_i \beta_i \le C + d\rho_c.$$

Hence $C + d\rho_c < M/2$ also suffices for all intermediate polynomials to have centered coefficients.

Complete cleaning error.

Proof (Theorem 3.5). Let κ_a be the sum of the two component switching errors in the relinearization of $u\,\mathsf{ct}^{\otimes 3}$. The coefficient bounds imply $\|\sigma(\kappa_a)\|_\infty \le 2NA_a$. Since $\sigma_\ell(Z) = \Delta x_\ell$ and $|x_\ell| \le 1 + B$, the polynomial $uZ^3 + \kappa_a$ has canonical embedding norm at most $u\Delta^3(1+B)^3 + 2NA_a$. The second inequality in (8) therefore ensures that the preceding rescaling criterion holds with $M = Q$ and divisor product d_a.

Let ξ_a be the accumulated auxiliary rescaling error, so that $\|\sigma(\xi_a)\|_\infty \le \rho_a$. Decryption of the auxiliary ciphertext gives the centered representative $Z_a = (uZ^3 + \kappa_a)/d_a + \xi_a$. At scale Γ, its value in slot ℓ is

$$T_\ell = \frac{u\Delta^3}{d_a\Gamma}x_\ell^3 + \frac{\sigma_\ell(\kappa_a)}{d_a\Gamma} + \frac{\sigma_\ell(\xi_a)}{\Gamma}.$$

Consequently, $T_\ell - x_\ell^3$ has magnitude at most the left hand side of (6), proving the auxiliary bound.

The first inequality in (8) ensures that Z is still centered after the input ciphertext is reduced to modulus Q/d_a. Integral weights then give the polynomial $W = \sum_{i,j} w_{ij} Z^i Z_a^j \in \mathcal{R}$, and decryption of the final tensor gives $W \bmod (Q/d_a)$. Moreover, $\sigma_\ell(W) = SP(x_\ell, T_\ell)$. The input and auxiliary bounds imply $P(x_\ell, T_\ell) = \zeta_\ell + e_\ell$, with $|e_\ell| \le \varepsilon(P; B, \theta)$.

Let κ_f be the sum of the final component switching errors, each bounded by A_f in the coefficient norm. A degree D tensor has at most $D-1$ components to switch, so $\|\sigma(\kappa_f)\|_\infty \le (D-1)NA_f$. The polynomial $W + \kappa_f$ has canonical embedding norm at most $S(1+\varepsilon(P;B,\theta)) + (D-1)NA_f$. Consequently, the third inequality in (8) ensures that the rescaling criterion holds at $M = Q/d_a$. The same inequality also ensures that W is centered before the final relinearization.

Let ξ be the accumulated final rescaling error, with $\|\sigma(\xi)\|_\infty \le \rho$. Decryption of the output gives $Z' = (W + \kappa_f)/d + \xi$, and hence

$$\frac{\sigma_\ell(Z')}{\Delta'} - \zeta_\ell = \frac{Se_\ell}{d\Delta'} + \frac{\sigma_\ell(\kappa_f)}{d\Delta'} + \frac{\sigma_\ell(\xi)}{\Delta'} + \left(\frac{S}{d\Delta'} - 1\right)\zeta_\ell.$$

The triangle inequality and $|\zeta_\ell| = 1$ prove (9) simultaneously in all slots. The proof is deterministic whenever the assumed error bounds hold. A union bound therefore gives the stated failure probability without requiring independence of these events. □

Integer recombination and output rescaling. For integer recombination, let W be the corresponding weighted sum of the polynomials represented by the final digit tensors. In slot ℓ, $\sigma_\ell(W) = S(v_\ell + \sum_j 2^j e_{j,\ell})$. Let κ_f denote the error of their shared relinearization and ξ the accumulated rescaling error. Decryption of the output gives $Z' = (W + \kappa_f)/d + \xi$, so

$$\frac{\sigma_\ell(Z')}{\Delta'} - v_\ell = \frac{S\sum_j 2^j e_{j,\ell}}{d\Delta'} + \frac{\sigma_\ell(\kappa_f)}{d\Delta'} + \frac{\sigma_\ell(\xi)}{\Delta'} + \left(\frac{S}{d\Delta'} - 1\right)v_\ell.$$

Since $|v_\ell| \le p - 1$, the triangle inequality gives (11). The modulus condition of Section 3.4 ensures that the repeated rescaling criterion holds for $W + \kappa_f$ at modulus q. For $k_j = 0$, integral multiplication by S/Δ_1 preserves the decoded digit and its error bound B_1, so the same argument applies.

These output error identities hold for every positive rational Δ'. In particular, $\Delta' = S/d$ makes the coefficient of v_ℓ vanish without changing the rescaling operation. For the additional output rescaling in Section 3.4, let $W_0 = \Delta\sigma^{-1}(\mathbf{v}) + e_a + u_0$ at modulus Q_0. The assumed bounds give $\|W_0\|_\infty \le \Delta(p-1+\epsilon) + U_0$. Condition (12) therefore guarantees that the coefficient criterion above holds throughout rescaling. The final centered polynomial is

$$W' = \frac{\Delta}{d}\sigma^{-1}(\mathbf{v}) + \frac{e_a}{d} + \frac{u_0}{d} + \xi, \qquad \|\xi\|_\infty \le \rho_c.$$

At scale $\Delta' = \Delta/d$, the error e_a/d contributes at most ϵ in each decoded slot. Since the remaining error has coefficient norm at most $U_0/d + \rho_c = U$, its contribution in each decoded slot is at most NU/Δ'. These estimates prove the output bound of Section 3.4, with parameters (13).

If the final tensor is multiplied by an integer $u_f \ge 1$ before relinearization and the decoding scale is increased by the same factor, then S and Δ' become $u_f S$ and $u_f\Delta'$. With the same rescaling divisors and switching error bound, the final switching contribution to the decoded error is divided by u_f, whereas the propagated polynomial error is unchanged. The resulting polynomial must satisfy the final modulus condition for the binary or integer output, respectively, with the new scale.

RNS key switching. For HYBRID switching, let the g active partitions have products $Q^{(j)}$ and contain $\ell_j \ge 1$ ordinary primes. Let P_{ks} be the product of $\beta \ge 1$ special primes. Suppose each evaluation key error coefficient has magnitude at most B_{key} and $\|s\|_1 \le N$. A conservative complete coefficient bound for one component switch is

$$A = \frac{2NB_{\mathrm{key}}}{P_{\mathrm{ks}}} \sum_{j=1}^{g} \ell_j Q^{(j)} + 2\beta(N+1). \tag{28}$$

To obtain this bound, approximate CRT extension expresses each coefficient of a partition digit as a sum of ℓ_j CRT terms, each smaller than $Q^{(j)}$. The coefficient magnitude is at most $(\ell_j+1)Q^{(j)}$. Multiplication by the corresponding key error and division by P_{ks} contribute at most $NB_{\mathrm{key}}\sum_j(\ell_j+1)Q^{(j)}/P_{\mathrm{ks}}$. In approximate modulus reduction, each output component subtracts a CRT sum of β terms smaller than P_{ks}. After division, its correction has coefficient magnitude at most $\beta+1$. The resulting decryption error has coefficient norm at most $(\beta+1)(N+1)$. The inequalities $\ell_j + 1 \le 2\ell_j$ and $\beta+1 \le 2\beta$ give (28), whose second term bounds the modulus reduction remainder. Since the derivation uses the ranges of the decomposed residues, the bound also applies after integer weights have been multiplied into the tensors and the tensors have been added.

B.2 Joint Evaluation of Conjugation and the First Cleaning Step

Let $U = (b, a) \in \mathcal{R}_{Q_0}^2$ have positive integer scale Δ_0, and let Z be the centered representative of $b + as \bmod Q_0$. Let ι be the conjugation automorphism and

$\mathrm{Tr}(Z) = Z + \iota(Z)$. Assume that $\|\sigma(Z)\|_\infty \le M\Delta_0$ and that $\sigma_\ell(\mathrm{Tr}(Z))/\Delta_0 = \zeta_\ell + h_\ell$, where $\zeta_\ell \in \{-1, 1\}$ and $|h_\ell| \le B_0$.

We first relinearize $U \otimes U$. Write the resulting ciphertext as (b_2, a_2), so that

$$W = b_2 + a_2 s = Z^2 + \kappa_0 \pmod{Q_0}.$$

The auxiliary ciphertext retains scale Δ_0^2, without rescaling. Define the public ring elements

$$\begin{aligned} C_0 &= 3\Delta_0^2 b - \big(b_2 + 3\iota(b_2)\big)b, \\ C_1 &= 3\Delta_0^2 a - \big(b_2 + 3\iota(b_2)\big)a - a_2\big(b + 3\iota(b)\big), \\ C_2 &= -a_2 a, \qquad C_{11} = -3a_2\iota(a). \end{aligned}$$

For the expression $V = C_0 + C_1 s + C_2 s^2 + C_{11} s\iota(s)$, expansion gives

$$\mathrm{Tr}(V) = \mathrm{Tr}\big(3\Delta_0^2 Z - W\big(Z + 3\iota(Z)\big)\big) \pmod{Q_0}.$$

The mixed component is evaluated using a key from $\iota(s)$ to s. Let its output be (β, α), so that

$$\beta + \alpha s = C_{11}\iota(s) + \kappa_1 \pmod{Q_0}.$$

The public tensor $(C_0, C_1+\beta, C_2+\alpha)$ consequently decrypts to $V + s\kappa_1 \pmod{Q_0}$. We then relinearize this tensor and add its homomorphic conjugate. Let κ_2 and κ_3 denote the respective errors. The final ciphertext decrypts to

$$\begin{aligned} Y &= H_{\Delta_0}(\mathrm{Tr}(Z)) + \kappa \pmod{Q_0}, \\ \kappa &= \mathrm{Tr}\big(-\kappa_0\big(Z + 3\iota(Z)\big) + s\kappa_1 + \kappa_2\big) + \kappa_3. \end{aligned}$$

Indeed, substitution of $W = Z^2 + \kappa_0$ into the preceding trace expression gives

$$\mathrm{Tr}(3\Delta_0^2 Z - Z^2(Z + 3\iota(Z))) = 3\Delta_0^2 \,\mathrm{Tr}(Z) - \mathrm{Tr}(Z)^3.$$

Suppose that $\|\kappa_0\|_\infty, \|\kappa_2\|_\infty \le A_2$ and $\|\kappa_1\|_\infty, \|\kappa_3\|_\infty \le A_\iota$. Conjugation preserves the canonical embedding norm, and $\|\sigma(\mathrm{Tr}(F))\|_\infty \le 2\|\sigma(F)\|_\infty$. For a ternary secret, $\|\sigma(s)\|_\infty \le \|s\|_1 \le N$. Therefore,

$$\begin{aligned} \|\sigma(\kappa)\|_\infty &\le 8M\Delta_0 N A_2 + 2N^2 A_\iota + 2NA_2 + NA_\iota \\ &= N\left((8M\Delta_0 + 2)A_2 + (2N+1)A_\iota\right) =: E_1. \end{aligned}$$

The term $2N^2 A_\iota$ bounds the trace of $s\kappa_1$ using $\|s\|_1 \le N$.

Since $R_1(\zeta+h) - \zeta = -3\zeta h^2/2 - h^3/2$, the ideal polynomial residual is bounded by $\varepsilon_1(B_0) = 3B_0^2/2 + B_0^3/2$. At scale $S_1 = 2\Delta_0^3$, the polynomial $H_{\Delta_0}(\mathrm{Tr}(Z)) + \kappa$ therefore has canonical embedding norm at most $S_1(1 + \varepsilon_1(B_0)) + E_1$.

Let $q_1, \ldots, q_m$ be the prescribed rescaling primes, with product $d_1 \mid Q_0$, and let ρ_1 be the bound on the accumulated rounding error defined by (27). The recurrence in (27) shows that

$$S_1(1 + \varepsilon_1(B_0)) + E_1 + d_1\rho_1 < Q_0/2$$

prevents modular wraparound before and throughout rescaling. Decryption of the output gives the centered representative

$$Z_1 = \frac{H_{\Delta_0}(\mathrm{Tr}(Z)) + \kappa}{d_1} + \xi, \qquad \|\sigma(\xi)\|_\infty \le \rho_1.$$

Consequently, decoding at positive integer scale Δ_1 gives

$$\frac{\sigma_\ell(Z_1)}{\Delta_1} - \zeta_\ell = \frac{S_1}{d_1\Delta_1}\big(R_1(\zeta_\ell + h_\ell) - \zeta_\ell\big) + \frac{\sigma_\ell(\kappa)}{d_1\Delta_1} + \frac{\sigma_\ell(\xi)}{\Delta_1} + \left(\frac{S_1}{d_1\Delta_1} - 1\right)\zeta_\ell.$$

The triangle inequality proves the bound B_1 stated in (10) of Section 3.4.

The joint evaluation therefore requires two switches from s^2 and two from $\iota(s)$, without an additional key for $s\iota(s)$. All subsequent rescalings act on two component ciphertexts, so Lemma B.1 bounds their rounding errors. A union bound combines the exceptional probabilities of the extraction and evaluation key bounds.

B.3 Ordinary Polynomial Evaluation

In ordinary evaluation with integer multiplication before key switching, intermediate switching errors enter subsequent tensor products. Let A bound the coefficient error of each component switch, so that the canonical embedding norm is bounded by $E = NA$. For $R_1 = (3x - x^3)/2$, relinearization of $\mathsf{ct} \otimes \mathsf{ct}$ gives a ciphertext that decrypts to $U \bmod Q$, where $U = Z^2 + e_1$ and $e_1 \in \mathcal{R}$ is the switching error with $\|\sigma(e_1)\|_\infty \le E$. The next tensor decrypts to $Z(3\Delta^2 - U) \pmod Q$. Before its final switch, the decoded signed digit error is at most

$$\varepsilon_1(B) + \frac{(1+B)E}{2\Delta^2}.$$

For $R_2 = (15x - 10x^3 + 3x^5)/8$, form $U = Z^2 + e_1$ and $V = 3U^2 + e_2$, with the multiplier three applied before the second relinearization and $\|\sigma(e_2)\|_\infty \le E$. Before final relinearization, the tensor decrypts to $Z(15\Delta^4 - 10\Delta^2 U + V) \pmod Q$ at scale $8\Delta^5$. Expansion in e_1, e_2 gives the bound

$$\varepsilon_2(B) + \frac{(1+B)(4+12B+6B^2)E}{8\Delta^2} + \frac{3(1+B)E^2 + (1+B)E}{8\Delta^4}.$$

Indeed, the linear coefficient of e_1 is $Z(-10\Delta^2 + 6Z^2)$ and the remaining terms are $3Ze_1^2 + Ze_2$. The canonical embedding norm of the linear coefficient is at most $\Delta^3(1+B)(4+12B+6B^2)$. The terms in e_1 and e_2 can be bounded separately when the two switches use different parameters. After the final switching error is included, the preceding rescaling analysis gives the complete output bound, with any integer multiplier before that switch incorporated into the scale.

B.4 Rescaling Error

CKKS rescaling divides both ciphertext components by an RNS prime and rounds their coefficients to the nearest integers, as in Section 2.2 [2]. For an odd prime q and an integer coefficient z, the rounding error $\lfloor z/q \rceil - z/q$ has magnitude at most $(1 - 1/q)/2$.

Lemma B.1 (Rescaling error). *Let $s \in \mathcal{R}$ have ternary coefficients, and let q be an odd prime dividing the ciphertext modulus Q. For a ciphertext $(b, a) \in \mathcal{R}_Q^2$, let $(b', a') = \mathsf{Rescale}_q(b, a)$ and define $r_b = \lfloor b/q \rceil - b/q$ and $r_a = \lfloor a/q \rceil - a/q$ using centered representatives. The rescaled ciphertext satisfies*

$$b' + a's = \frac{b + as}{q} + \delta \pmod{Q/q}, \qquad \delta = r_b + r_a s.$$

The error bounds are

$$\|\delta\|_\infty \le \frac{1 + \|s\|_1}{2}(1 - 1/q) \le \frac{N+1}{2}(1 - 1/q),$$
$$\|\sigma(\delta)\|_\infty \le \frac{N(1 + \|s\|_1)}{2}(1 - 1/q) \le \frac{N(N+1)}{2}(1 - 1/q).$$

Proof. Nearest integer rounding gives $\|r_b\|_\infty, \|r_a\|_\infty \le (1 - 1/q)/2$. Since negacyclic multiplication satisfies $\|r_a s\|_\infty \le \|r_a\|_\infty \|s\|_1$, the coefficient bound follows from $\delta = r_b + r_a s$ and $\|s\|_1 \le N$. The inequality $\|\sigma(\delta)\|_\infty \le N\|\delta\|_\infty$ then gives the canonical embedding bound. □

These deterministic bounds apply at each step of a rescaling sequence. Equation (27) accumulates the errors at the output modulus. Consequently, under the stated modulus conditions, the failure probability of a cleaning computation is at most $\epsilon_{\text{in}} + \epsilon_{\text{key}}$, where these terms bound failures of the input error and evaluation key error assumptions over the complete computation.

C Proof of the Threshold Application

C.1 Model and Uniform Bounds

We use the strong simulation experiment of KLLPS [15] at the fixed decryption modulus q, with response query bound q_{pd}. After receiving the public key and evaluation keys, the adversary chooses a static corrupted set C, $|C| < t$, and receives the corresponding pairs (s_i, z_i). The encryption and evaluation oracles record each ciphertext together with its exact integer output. A response query identifies a recorded ciphertext and a party. Both experiments allow adaptive and repeated queries, reject invalid queries and requests beyond the query bound, and give the adversary access to the same random oracles as the scheme algorithms.

For each ciphertext, the ideal experiment maintains a set $P[\mathsf{ct}]$ initialized to C. A query for party i first updates the set to $P[\mathsf{ct}] \cup \{i\}$. The simulator receives the exact output only if $|P[\mathsf{ct}]| \ge t$. Different records of the same canonical ciphertext

share this set, so neither repeated queries nor duplicate records advance the count. Responses from corrupted parties use their known keys.

Theorem 4.1 assumes that the error bounds hold uniformly over permitted computations and query histories, except with total probability at most ϵ_{bad}. We use the same public bounds D_* and B_{sm} for all permitted scales and error bounds at q. In each response, the error bounds apply before the fresh flooding noise is sampled.

The reconstruction and simulation bounds both depend on integer representatives of Lagrange coefficients. For reconstruction, fix a responding set S and $i \in S$. The identity $\prod_{u\in[n]\setminus\{i\}}(1-\alpha_i\alpha_u^{-1}) = n$ and the Lagrange formula give

$$c\lambda_i^S \equiv \prod_{u\in[n]\setminus S}(1-\alpha_i\alpha_u^{-1}) = \omega_{S,i} \pmod q.$$

Each factor has coefficient ℓ_1 norm at most two, which yields $\|\omega_{S,i}\|_1 \le 2^{n-t} = B_2$. For $a, b \in \mathcal{R}\otimes_{\mathbb{Z}}\mathbb{R}$, we use $\|a\|_\infty \le \|\sigma(a)\|_\infty \le \|a\|_1 \le N\|a\|_\infty$ and $\|ab\|_\infty \le \|a\|_1\|b\|_\infty$. Consequently,

$$\left\|\sigma\left(\sum_{i\in S}\omega_{S,i}g_i\right)\right\|_\infty \le N\sum_{i\in S}\|\omega_{S,i}\|_1\|g_i\|_\infty \le Nt2^{n-t}B_{\mathrm{sm}}.$$

Likewise, the reconstructed RLWE error $\|\sigma(\sum_{i\in S} c\omega_{S,i}\eta_i)\|_\infty$ is bounded by KcB_χ.

For simulation, let T have size $t-1$ and let $i \notin T$. The Lagrange basis polynomial at zero satisfies $\ell_0(\alpha_i) = \prod_{j\in T}(1-\alpha_i\alpha_j^{-1}) \in \mathcal{R}$, and therefore $\|c\ell_0(\alpha_i)\|_1 \le c2^{t-1}$. For $j \in T$, the remaining basis values are

$$\ell_j(\alpha_i) = \frac{\alpha_i}{\alpha_j}\prod_{k\in T\setminus\{j\}}\frac{\alpha_i-\alpha_k}{\alpha_j-\alpha_k}.$$

To clear the denominators, we use the identity $\prod_{u=1}^n(Y-\alpha_u) = Y^n - 1$. Differentiation at α_j gives $\prod_{u\ne j}(\alpha_j-\alpha_u) = n\alpha_j^{n-1}$. Since $c = n$ and $\alpha_j^n = 1$, we obtain the integer polynomial

$$c\ell_j(\alpha_i) = \alpha_i\prod_{k\in T\setminus\{j\}}(\alpha_i-\alpha_k)\prod_{u\in[n]\setminus T}(\alpha_j-\alpha_u).$$

The two products contain $n-1$ factors, so $\|c\ell_j(\alpha_i)\|_1 \le 2^{n-1} = B_3$. When $t = 1$, this family is empty. Thus the flooding factor c clears the reconstruction denominators, while the ratio $c^2/c = c$ clears the simulation denominators.

The error relative to $m_0 = \lfloor \Delta' M \rceil$ satisfies

$$\begin{aligned} e &= e_a + u + (\Delta' M - m_0),\\ \|e\|_\infty &\le \Delta'\epsilon + U + \tfrac{1}{2}. \end{aligned}$$

The simulation argument below gives an integer representative of the response difference after division by c, with coefficient ℓ_1 norm

$$\begin{aligned}&\left\| c\ell_0(\alpha_i)e + c\eta_i - \sum_{j\in T} c\ell_j(\alpha_i)\eta_j \right\|_1 \\ &\quad \le N\left[c2^{t-1}(\Delta'\epsilon + U + \tfrac{1}{2}) + (c + (t-1)B_3)B_\chi\right] \le D_*.\end{aligned}$$

By (17), the chosen D_* therefore bounds this difference for every responding party, set T, and permitted scale and error bound at q.

C.2 Correctness

Fix a ciphertext satisfying the stated error bounds and a responding set S of size t. The zero sharing gives $\sum_{i\in S} \lambda_i^S z_i = 0$ in $\mathcal{R}_q$. The masks therefore cancel in (15). By (16), reconstruction gives the following integer polynomial modulo the decryption modulus:

$$\begin{aligned}c^2 Z + \sum_{i\in S} \omega_{S,i}(c\eta_i + g_i) = c^2\Delta' M + c^2 e_a + c^2 u \\ + \sum_{i\in S} \omega_{S,i}(c\eta_i + g_i).\end{aligned} \tag{29}$$

Since $\|M\|_\infty \le \|\mathbf{v}\|_\infty \le p-1$, the polynomial in (29) has coefficient norm at most $c^2\Delta'(p-1+\epsilon) + c^2 U + tB_2(cB_\chi + B_{\mathrm{sm}})$. By (18), this polynomial is therefore the centered representative computed by final decryption.

The decoded error relative to $c^2\mathbf{v}$ is bounded by

$$\left\| \frac{\sigma(c^2 e_a + c^2 u + \sum_{i\in S} \omega_{S,i}(c\eta_i + g_i))}{\Delta'} \right\|_\infty \le c^2\epsilon + \frac{c^2 NU + K(cB_\chi + B_{\mathrm{sm}})}{\Delta'}.$$

The numerical decoding error contributes at most ν to this bound. Condition (19) therefore places each real coordinate within distance $1/2$ of the corresponding integer in $c^2\mathbf{v}$ and each imaginary coordinate within distance $1/2$ of zero. Hence $\mathbf{y} = c^2\mathbf{v}$, all membership checks succeed, and exact integer division gives $(\mathbf{v} + p - 1)/2 = \mathbf{m}$. The argument applies to every responding set and every noise sample satisfying the assumed bounds. Correctness therefore fails with probability at most ϵ_{bad}. Output rescaling in Section 3.4 preserves this decryption relation with parameters (13), as proved in Appendix B.1. The same argument therefore establishes correctness after rescaling.

C.3 Strong Simulation Security

We adapt the KLLPS simulator [15] to the reference encoding m_0. Uniform flooding hides the response difference derived above, with the statistical distance bounded by the following lemma.

Lemma C.1 (Uniform flooding). *Let $B \geq 0$ be an integer, let $g \in \mathcal{R}$ have independent coefficients uniform on $\{-B, \ldots, B\}$, and let $d \in \mathcal{R}$. Then $\mathsf{SD}(g, g + d) \leq \min\{1, \|d\|_1/(2B+1)\}$. The same bound holds after reduction modulo any positive integer.*

Proof. Write $d = \sum_{j=0}^{N-1} d_j X^j$. For one coefficient, the overlap of the uniform intervals before and after translation has size $\max\{0, 2B+1-|d_j|\}$. The statistical distance is therefore $\min\{1, |d_j|/(2B+1)\}$. A hybrid over coordinates gives the ℓ_1 bound, with statistical distance at most one. The data processing inequality preserves the bound after reduction. □

Simulator. The simulator generates the corrupted CKKS shares from a sharing of zero over $\mathcal{R}_Q$, and their masking shares from an independent sharing of zero over $\mathcal{R}_q$. Only the entries indexed by C are given to the adversary. For each canonical ciphertext, the simulator stores the deterministic response components introduced below.

Before the threshold is reached, the simulator samples a uniform $\beta_i \in \mathcal{R}_q$ for each newly queried uncorrupted party i and stores it for that ciphertext. Every response from this party is $\beta_i + cg_i \bmod q$, with a fresh flooding sample g_i. Repeated queries use the same β_i, so only the flooding noise is resampled. For a corrupted party, the deterministic component is computed from the known keys as

$$\beta_i = c^2\big(a(s_i \bmod q) + H(\mathsf{ct})z_i + \eta_i\big) \pmod q.$$

On the first query reaching t distinct parties, the queried index i is new and uncorrupted. The simulator fixes $T = P[\mathsf{ct}] \setminus \{i\}$, so $|T| = t - 1$, obtains $\mathbf{m}$, and computes m_0. For $i \in T$, the response remains $\beta_i + cg_i$. For $i \notin T$, the simulator returns

$$\widetilde{d}_i = c^2 \ell_0(\alpha_i)(m_0 - b) + \sum_{j \in T} \ell_j(\alpha_i)\beta_j + cg_i \pmod q.$$

Subsequent responses use the same T and the stored components on T, with fresh flooding for every invocation. For $t = 1$, the first query releases the output, T is empty, and the masking polynomial is identically zero.

Statistical hybrid. We first use the simulator's interpolation formula for responses generated after the threshold is reached, while retaining the real deterministic components β_j on T. Write d_i^{H} for a response in this hybrid experiment. For $i \in T$, the response distribution is unchanged. For $i \notin T$, Shamir interpolation gives

$$s_i = \ell_0(\alpha_i)s + \sum_{j \in T} \ell_j(\alpha_i)s_j \pmod q,$$

$$z_i = \sum_{j \in T} \ell_j(\alpha_i)z_j \pmod q.$$

Since $as = m_0 - b + e \pmod q$, the real deterministic component differs from its counterpart in the hybrid by

$$c^2 \left(\ell_0(\alpha_i)e + \eta_i - \sum_{j \in T} \ell_j(\alpha_i)\eta_j \right) \pmod q.$$

Conditional on the key material, random oracle answers, ciphertext, and common query history, the difference is fixed and only the fresh flooding remains random. Since c is invertible modulo q, we may divide both responses by c and apply Lemma C.1 to obtain

$$\mathsf{SD}(d_i, d_i^{\mathrm{H}}) \le \frac{\left\| c\ell_0(\alpha_i)e + c\eta_i - \sum_{j \in T} c\ell_j(\alpha_i)\eta_j \right\|_1}{2B_{\mathrm{sm}} + 1} \le \frac{D_*}{2B_{\mathrm{sm}} + 1}.$$

The bound holds uniformly over T, including sets determined by adaptive queries. We couple the experiments using the same keys, random oracles, and randomness except for flooding. As long as the transcripts agree, the adversary makes the same next query in both experiments. A maximal coupling of the fresh responses then disagrees with probability at most $D_*/(2B_{\mathrm{sm}}+1)$ whenever the error bounds hold. An error bound fails before the first disagreement with probability at most ϵ_{bad}. A union bound over all response queries therefore gives

$$\mathsf{SD}(\mathsf{View}_{\mathrm{real}}, \mathsf{View}_{\mathrm{H}}) \le \min \left\{ 1, \frac{q_{\mathrm{pd}} D_*}{2B_{\mathrm{sm}} + 1} + \epsilon_{\mathrm{bad}} \right\} \le \delta + \epsilon_{\mathrm{bad}}. \tag{30}$$

Computational hybrids. We next replace the deterministic components returned before the threshold is reached. For each ciphertext, at most $t - 1 - |C|$ distinct uncorrupted parties receive such responses. The zero shares at any fixed set of at most $t-1-|C|$ additional indices are jointly uniform conditional on the corrupted zero shares. The masks $H(\mathsf{ct})z_i + \eta_i$ and the affine terms $a(s_i \bmod q)$ in (14) are exactly those of the second KLLPS construction at modulus q. Moreover, c^2 is invertible modulo q, so $c^2\mathcal{R}_q = \mathcal{R}_q$. The KLLPS computational hybrid therefore replaces these components by uniform elements of $\mathcal{R}_q$ under the decisional $\mathrm{RLWE}_{q,\chi}$ assumption in the random oracle model [15]. The hybrid applies jointly across ciphertexts under the same zero sharing and consistent random oracle answers. Hence the replacement also covers later responses computed from these deterministic components.

The remaining change is to generate the corrupted CKKS shares from a sharing of zero. Extend C to a set C' of size $t-1$. For fixed s, the vector $(s_j)_{j \in C'}$ is an affine function of the independent uniform nonconstant coefficients of f, with matrix $V = (\alpha_j^k)_{j \in C',\, 1 \le k < t}$ and determinant

$$\det V = \left(\prod_{j \in C'} \alpha_j \right) \prod_{\substack{j,k \in C' \\ j<k}} (\alpha_k - \alpha_j).$$

All factors are units in $\mathcal{R}_Q$. Hence the shares indexed by C are uniform and independent of s, even conditional on the encryption and evaluation keys. A sharing of zero gives the same distribution, and the independent masking shares already have the simulator's distribution.

The adversary receives the corrupted CKKS shares at Q, whereas the KLLPS hybrid uses their reductions at q. For $q < Q$, condition on $f \bmod q$. The nonconstant coefficients of f are uniform over their compatible lifts to $\mathcal{R}_Q$. The same matrix is invertible modulo Q/q, so the corrupted shares are conditionally uniform over all lifts compatible with their reductions modulo q. Since responses use only these reductions, the additional information at modulus Q can be simulated by sampling those lifts. For $q = Q$, the reduction is the identity. These lifts preserve the KLLPS hybrid at q, so the final response distribution is exactly that of the ideal experiment.

Polynomial running time. Ring arithmetic, interpolation, and sampling take polynomial time, and the response tables contain at most polynomially many entries. The reference encoding m_0 is also computable in polynomial time. Indeed, each coefficient of $\Delta' M$ has a rational representation of polynomial bit length in the cyclotomic field of degree N. After clearing denominators, comparison with a half integer reduces to determining the sign of an algebraic integer β. Equality is decided in the integral basis and handled by the fixed rounding rule. Otherwise, the field norm is a nonzero integer. If $B \geq 1$ bounds the other conjugates, then $|\beta| \geq B^{-(N-1)}$. Such a bound has polynomial bit length, so interval approximation with polynomially many bits determines the sign. For positive ν, the numerical decoder can likewise use interval arithmetic to meet the prescribed error bound [21].

Together, (30) and the KLLPS computational hybrids give distinguishing advantage at most $\delta + \epsilon_{\text{bad}}$ plus a negligible term. The encryption algorithm and public key distribution are unchanged, so IND-CPA security with the required evaluation keys completes the proof of Theorem 4.1.

D Parameters and Additional Results

The extraction output determines the input scale and modulus for cleaning. In Table 3, leveled extraction gives scale 1125899943673858 with 20 ordinary primes, whereas functional bootstrapping gives scale 239294774892 with 21 ordinary primes, in both cases after conjugation. The error bounds below assume signed digit errors at most 2^{-14} and 2^{-15}, respectively, and evaluation key error coefficients of magnitude at most 39.

D.1 Cleaning Functions and Scales

For $R_1 \to R_1 \to R_1 \to R_2$, the evaluation scales are $S = 2\Delta^3$ in the cubic rounds and $S = 8\Delta^5$ in the final round. After rescaling, the decoding scale is the nearest positive integer to the rational output scale (Table 6).

Table 6. Scale and modulus schedules for the integer experiments. The final scale represents the centered integer $v = 2m - 255$.

Stage	Input primes	Removed	Output primes	$\log_2 \Delta'$
Leveled extraction				
First R_1	20	2	18	52.000000
Second R_1	18	1	17	107.000000
Third R_1	17	4	13	122.000001
Final R_2	13	3	10	463.000005
Functional bootstrapping				
First R_1	21	2	19	44.499988
Second R_1	19	2	17	72.496885
Third R_1	17	3	14	118.988890
Final R_2	14	6	8	325.944448

The complete error bound follows by propagating the switching bound in (28) and the rescaling bound of Lemma B.1 through the remaining rounds and weighted recombination. Under the stated input and evaluation key error assumptions, all three methods have conditional lower bounds on normalized precision of 246.207540 bits for leveled extraction and 156.417871 bits for functional bootstrapping. These bounds use uniform estimates of the switching and rounding errors, whereas the precisions above 300 bits in Table 3 are measured from the complete output. The modulus condition of Section 3.4 holds with final moduli of 501.00 and 336.50 bits, reduced from cleaning input moduli of approximately 1000.00 and 839.90 bits, respectively.

The final round of auxiliary evaluation uses the $r = 2$ family member

$$P_2(x, T) = \frac{15x - 7x^3 - 3T + 3x^2T}{8},$$
$$P_2(x, x^3 + \eta) - R_2(x) = \frac{3}{8}(x^2 - 1)\eta.$$

The choice $\Gamma = \Delta^3$ avoids rescaling the auxiliary ciphertext, so the auxiliary error comes from the two switches used to generate the cube. The factor $3(x^2 - 1)/8$ in the identity above determines its contribution to the final cleaning error for each digit.

D.2 Switching Keys and Stage Costs

The maximum bit lengths of the products of ciphertext and special moduli are 1721 and 1541 for extraction and cleaning in the leveled route, and 1747 and 1328 with functional bootstrapping.

Let g be the number of decomposition digits, K the number of ordinary primes, and β the number of special primes. A switching key contains g polynomial pairs over $K+\beta$ RNS primes, occupying $2g(K+\beta)N\cdot 8$ bytes at 64 bits per residue. Each application requires $2g(K+\beta)$ polynomial products, in addition to decomposition and basis conversion.

The final round of ordinary evaluation requires keys for $s^2, \ldots, s^5$, whereas auxiliary evaluation uses keys for s^2, s^3. However, the auxiliary construction requires a larger special modulus to control the cube generation error through (28). Consequently, the storage saving depends on both the number of keys and the number of special primes, as shown in Table 7.

Table 7. Key switching parameters for integer cleaning. A sequence of β values follows the round order. The first three rounds are common to ordinary and auxiliary evaluation.

Evaluation	Rounds	g	β	Keys (MiB)
Leveled extraction				
Ordinary / Auxiliary	1–3	2	$(10, 9, 9)$	166
Ordinary	4	1	7	80
Auxiliary	4	1	10	46
Fully delayed	1–4	2	$(10, 9, 9, 2)$	452
Functional bootstrapping				
Ordinary / Auxiliary	1–3	2	$(9, 8, 7)$	162
Ordinary	4	1	7	84
Auxiliary	4	1	10	48
Fully delayed	1–4	2	$(9, 8, 7, 1)$	444

All three methods rescale two component ciphertexts according to the same schedule within each route. The first three rounds rescale each digit separately, while the final round rescales one recombined ciphertext, giving $8(2+1+4)+3 = 59$ prime divisions for leveled extraction and $8(2+2+3)+6 = 62$ for functional bootstrapping. The differences in cleaning time therefore come mainly from relinearization and basis conversion (Table 8).

Table 8. Operation times in seconds for cleaning and recombination. Relinearization includes basis conversion. Other arithmetic includes tensor products, integer scalar multiplication and recombination.

Evaluation	Relinearization	Rescaling	Other arithmetic
Leveled extraction			
Ordinary	7.66	0.85	2.20
Fully delayed	5.82	0.92	1.77
Auxiliary	8.17	0.82	2.22
Functional bootstrapping			
Ordinary	6.44	0.57	2.22
Fully delayed	5.19	0.68	2.02
Auxiliary	7.66	0.59	2.25

D.3 Measured Rescaling Error

Let Z and Z' be the decryption polynomials immediately before and after a rescaling sequence with divisor product d and output scale Δ'. The rescaling error is $\max_\ell |\sigma_\ell(Z' - Z/d)|/\Delta'$, maximized over the digit branches in the first three rounds. For the recombined integer ciphertext, this error is divided by 512 under $v = 2m - 255$.

The contribution of rescaling to the complete error is measured separately in Table 9, using ordinary evaluation with parameters distinct from Table 3. The input has minimum signed digit precision 16.40 bits, scale 1166198535339, and 26 ordinary primes. The four rounds remove 2, 2, 3, and 8 primes, giving scale logarithms 45.25, 74.76, 126.28, and 362.40. The final rescaling contributes approximately $2^{-355.05}$ normalized error, whereas the complete output has 330.09 bits of normalized precision. Thus, final rescaling does not dominate the output error in this configuration.

Table 9. Measured rescaling error in ordinary evaluation. Values are negative base two logarithms of the errors.

Stage	Prime divisions	Error (bits)
First R_1	16	28.16
Second R_1	16	57.88
Third R_1	24	109.42
Final R_2	8	355.05

D.4 Parameters for Binary Noise Cleaning

The binary experiments of Section 5.2 use the same packed input and uniform ternary secret, with initial scale $\Delta_0 = 1166198535339$ and signed slot error bound $B_0 = 2^{-16}$. Under the evaluation key error bound $B_{\mathrm{key}} = 39$ in (28) and the deterministic rounding bound of Lemma B.1, the complete recurrence in Section 3.2 gives analytical precision $-\log_2(B_k/4)$. Every step satisfies the modulus condition of that section.

Table 10 gives additional compositions at the targets of Table 2, all with fully delayed evaluation. We write $(r_1, \ldots, r_k)$ for the Hermite orders in execution order, with R_{r_1} evaluated first.

Table 10. Additional binary cleaning compositions at the targets of Table 2.

Target	Orders	Precision (bits)		$\log_2 Q_0$	Time	Keys
(bits)		Analytical	Measured		(s)	(MiB)
128	$(1, 2, 1)$	131.20	165.74	550.01	0.570	158
	$(1, 1, 1, 1)$	147.22	165.81	685.02	0.866	196
200	$(3, 3)$	218.20	247.13	851.02	1.172	378
	$(1, 1, 1, 1)$	201.68	216.45	782.02	1.054	234
300	$(3, 4)$	302.65	311.64	1043.02	1.786	700
	$(2, 3, 1)$	302.93	317.64	1158.52	2.045	638
	$(1, 1, 1, 1, 1)$	300.14	329.09	1312.52	2.519	780

The initial modulus contains K_0 ordinary primes of 32 to 41 bits. At step i, key switching uses g_i decomposition digits and β_i special primes of 54 bits. The maximum bit length of $Q_{i-1}P_{\mathrm{ks},i}$ includes both ordinary and special primes (Table 11).

The scale choices in Table 12 use $S_i = L_{r_i}\Delta_{i-1}^{2r_i+1}$ and $\Delta_i = \lfloor S_i/d_i \rceil$, where d_i is the product of the removed primes. Without rescaling, $d_i = 1$ and $\Delta_i = S_i$.

Table 11. Modulus and key switching parameters for binary cleaning. The sequences list the values at successive steps. Removed denotes the number of ordinary primes removed after each polynomial evaluation.

Target (bits)	Orders	K_0	Removed	(g_i)	(β_i)	Max. QP_{ks} (bits)
128	(8)	21	(0)	(1)	(4)	935
	(2, 2)	18	(3, 0)	(1, 1)	(10, 4)	1159
	(1, 2, 1)	16	(2, 5, 0)	(1, 1, 1)	(10, 6, 4)	1091
	(1, 1, 1, 1)	20	(2, 2, 5, 0)	(1, 1, 1, 1)	(12, 10, 7, 4)	1334
200	(13)	32	(0)	(1)	(5)	1391
	(3, 3)	25	(6, 0)	(1, 1)	(13, 6)	1554
	(2, 2, 1)	23	(4, 7, 0)	(1, 1, 1)	(13, 8, 5)	1485
	(1, 1, 1, 1)	23	(2, 2, 5, 0)	(1, 1, 1, 1)	(14, 12, 9, 5)	1539
300	(19)	45	(0)	(2)	(1)	1675
	(3, 4)	30	(5, 0)	(2, 1)	(7, 7)	1422
	(2, 3, 1)	33	(3, 12, 0)	(2, 1, 1)	(9, 12, 7)	1692
	(1, 1, 1, 2)	31	(1, 4, 4, 0)	(2, 2, 2, 1)	(10, 7, 6, 7)	1622
	(1, 1, 1, 1, 1)	37	(1, 1, 14, 5, 0)	(3, 3, 2, 1, 1)	(8, 6, 2, 11, 7)	1745

Table 12. Output scale after each binary cleaning step. Values are $\log_2 \Delta_i$.

Target (bits)	Orders	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
128	(8)	696.444				
	(2, 2)	101.418	510.091			
	(1, 2, 1)	53.251	99.247	298.741		
	(1, 1, 1, 1)	54.252	95.752	118.247	355.742	
200	(13)	1105.294				
	(3, 3)	85.591	603.136			
	(2, 2, 1)	73.423	132.101	397.303		
	(1, 1, 1, 1)	57.253	106.760	151.269	454.808	
300	(19)	1598.313				
	(3, 4)	92.595	840.352			
	(2, 3, 1)	87.925	194.467	584.402		
	(1, 1, 1, 2)	82.755	95.265	148.791	746.955	
	(1, 1, 1, 1, 1)	82.755	210.765	115.791	180.365	542.096