A New CRT-based Fully Homomorphic Encryption

Anil Kumar Pradhan[1], Abhraneel Dutta[2], Hansraj Jangir[3], and Dipayan Das[3]

[1] Vaultree Ltd, Ireland
anil@vaultree.com
[2] Florida International University, USA
abdutta@fiu.edu
[3] Florida Atlantic University, USA
{hjangir2020, dasd}@fau.edu

Abstract. The idea of computing on encrypted data without decryption dates back to the notion of *privacy homomorphisms* introduced by Rivest, Adleman, and Dertouzos (1978). Their proposals built using the elegant structure of Chinese Remainder Theorem (CRT), were later shown to be insecure under simple known-plaintext attacks. Subsequent CRT-based fully homomorphic encryption (FHE) over the integers addresses this algebraic transparency by injecting noise and basing security on approximate common divisor–type assumptions, but the resulting designs are burdened by large public keys and costly ciphertext refresh procedures.

In this work, we develop a new CRT-based FHE scheme whose security relies on the Ring-LWE (RLWE) hardness assumption. For this purpose, we introduce the *CRT-RLWE* problem. We show that the problem is at least as hard as the RLWE, thereby positioning our construction within the established post-quantum security landscape of RLWE-based cryptography. Our scheme retains an explicit CRT embedding, separating a message component modulo a prime-power plaintext modulus and an auxiliary CRT component, while using RLWE-style key and ring arithmetic for compactness and efficiency.

Finally, we make it fully homomorphic by using a new bootstrapping procedure, that adopts the recryption paradigm for BGV/BFV schemes utilizing the linear transformation and digit extraction techniques.

Keywords: Fully Homomorphic Encryption (FHE)· Chinese Remainder Theorem· Chosen Plaintext Attack

1 Introduction

Fully Homomorphic Encryption (FHE) is a transformative cryptographic paradigm that enables computations on encrypted data without requiring decryption. While the scheme was introduced in 1978 by Rivest [28], the first scheme was proposed by Gentry in 2009 [18,19]. Later, it was improved in several works, for

example, [2,4,5,10,20,29]. FHE has opened new avenues for secure data processing, allowing operations to be performed on ciphertexts while maintaining the confidentiality of the underlying plaintexts. This capability is particularly valuable in scenarios where data privacy is paramount, such as in cloud computing, medical data processing, and financial transactions.

FHE has been approached through multiple mathematical frameworks, with CRT-based constructions representing one of the earliest and most influential directions. The concept of utilizing the Chinese Remainder Theorem (CRT) to construct a FHE was first introduced as a "privacy homomorphism" in 1978 by Rivest, Adleman, and Dertouzos [28]. However, the concrete instantiations proposed in [28] were vulnerable to even simple leakage patterns; in particular, the algebraic transparency of ring-homomorphic constructions known-plaintext attacks [7].

To address the vulnerability, [30] introduced a CRT-based homomorphic encryption scheme over the integers whose security relies on new assumptions, namely the approximate GCD (AGCD) assumption together with an additional sparse subset-sum–type assumption [12]. Nevertheless, the introduction of new assumptions has, in some cases, resulted in subsequent cryptanalytic breaks of homomorphic encryption schemes, as illustrated by the example in [16]. In that work, the authors showed that a homomorphic encryption scheme relying on the newly proposed Finite Field Isomorphism Problem is insecure by presenting a successful attack against it.

While integer/CRT constructions are conceptually appealing, they operate over familiar integer arithmetic and expose a natural CRT structure. Their path to *fully* homomorphic encryption has historically been dominated by bootstrapping cost and parameter bloat. Coron–Mandal–Naccache–Tibouchi [13] report benchmark datapoints illustrating this challenge: in a "large" parameter regime, a public key on the order of gigabytes (e.g., $\sim$2.3 GB) with multi-hour key generation (e.g., $\sim$2.2 h), minutes-level encryption (e.g., $\sim$3 min), and tens-of-minutes ciphertext refresh (e.g., $\sim$30 min). They also report AGCD-style parameterizations with public keys on the order of hundreds of megabytes (e.g., $\sim$800 MB) and refresh times on the order of minutes (e.g., $\sim$14 min) [13]. Subsequent work significantly reduced key size: Coron–Naccache–Tibouchi [14] describe compression and modulus switching techniques yielding an implementation with $\sim$10.1 MB public keys compared to $\sim$802 MB under similar parameters, albeit with security proven in the random oracle model. Nevertheless, these results underscore that (i) bootstrapping/refresh remains a central cost driver for integer/CRT-based FHE, and (ii) nontrivial auxiliary assumptions and engineering are often required to obtain practical parameters.

In contrast, fully homomorphic encryption based on Ring-LWE represents a fundamentally different and now dominant approach, replacing integer- and CRT-based constructions with a ring-lattice framework endowed with strong reduction-based security guarantees. Following the seminal work of Gentry [18, 19], which introduced the first fully homomorphic encryption scheme, a sequence of constructions grounded in Ring-LWE has emerged as the foundation of modern

FHE. In particular, schemes such as Brakerski–Gentry–Vaikuntanathan (BGV) [3], Fan–Vercauteren (FV) [17], and subsequent variants build homomorphic encryption directly on the hardness of Ring-LWE, enjoying security reductions to the worst-case hardness of standard ideal lattice problems [23].

From a systems perspective, RLWE-based schemes admit compact public keys consisting of only a constant number of ring elements. For standard parameter choices at the 128-bit security level, this results in public keys on the order of hundreds of kilobytes to a few megabytes in practice [6, 25], in stark contrast to the hundreds-of-megabytes to gigabytes required by earlier integer- and CRT-based bootstrapping schemes [13, 14].

In summary, prior CRT-based fully homomorphic encryption schemes over the integers preserve an appealing and transparent CRT structure, but suffer from two fundamental limitations. First, their security relies on AGCD-style assumptions, which have been shown to admit efficient lattice-based attacks under homomorphic parameter regimes [9]. Second, achieving full homomorphism in this setting has historically required expensive bootstrapping or ciphertext refresh procedures, often accompanied by large public keys and additional auxiliary assumptions [12–14]. These shortcomings motivate the search for alternatives that retain the conceptual advantages of CRT while overcoming both security and efficiency bottlenecks.

Our Contribution. In this work, we propose a CRT-based construction over rings rather than over the integers. By moving to an appropriate ring setting, we obtain a CRT-based design that avoids known vulnerabilities of integer-based AGCD assumptions and the security is based on the Ring-LWE assumption. This approach preserves the structural intuition of CRT while providing stronger theoretical guarantees and improved asymptotic efficiency. We summarize our main contributions below.

- We introduce a new hardness assumption, which we call *CRT-RLWE*. We first show that the new CRT-RLWE assumption is at least as hard as Ring LWE. As a result, the new assumption inherits the hardness of the standard RLWE problem. Leveraging our CRT-RLWE problem, we propose a new *CRT-based fully homomorphic encryption* scheme equipped with an explicit *bootstrapping* mechanism supporting *full* homomorphism.
- Our design aims to retain the structural advantages that motivated CRT-based homomorphic encryption while replacing AGCD-centric foundations with RLWE-hardness through CRT-RLWE, thereby aligning security with the modern lattice-based paradigm and enabling substantially smaller key material than historical integer/CRT FHE implementations.
- We quantify the practical implications of our design by comparing against established integer/CRT FHE benchmarks and key-compression techniques [13, 14], as well as standard RLWE parameter regimes used in practice [6, 25]. This comparison highlights that RLWE-style ring-element key material scales substantially more favorably than the historically large public keys and expensive refresh costs reported for integer/CRT bootstrapping-based constructions, while retaining a CRT-centric interface.

Related Work. We note that the Chinese Remainder Theorem (CRT) has previously been used in FHE constructions [22], albeit in a different context; the role of CRT in their work is fundamentally different from ours. In [22], CRT is used as an implementation-level packing technique to amortize multiple bootstrapping instances by combining accumulators and keys within a composite-modulus NTT framework and subsequently splitting the results back (for more details, we refer Algorithm 5 of [22]). In contrast, our construction CRT-FHE incorporates CRT at the scheme level: ciphertexts are defined to embed a CRT-encoded payload $\mathrm{CRT}_{p_1^r,p_2}(m,e)$, and our bootstrapping procedure is specifically designed to restore this native CRT representation after recryption.

Organization. The rest of the paper is organized as follows: we begin with the preliminaries in Section 2, followed by the definition of essential notations and parameters. Next, we describe the proposed fully homomorphic encryption (FHE) algorithm in detail in Section 3. We then provide the noise analysis of the proposed scheme in Section 4. Following this, we provide a proof of the algorithm's correctness and conduct a thorough security analysis in Section 5. We describe the bootstrapping procedure for the proposed scheme in Section 6. Finally, we present a performance analysis and benchmarking results in Section 7.

2 Preliminary

In this section, we provide an overview of the basic concepts that underpin our Fully Homomorphic Encryption (FHE) scheme, focusing on the Chinese Remainder Theorem (CRT).

2.1 Notation

Let n be an integer of the form power-of-two and define $f(X) = X^n + 1$. In this paper, we fix the ring to be $\mathcal{R} := \mathbb{Z}[X]/\langle f \rangle$. We write $\mathcal{R}_q$ to denote the ring $R/q\mathcal{R}$ for some prime q of sufficiently large size and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} \cong \mathbb{Z}_q[X]/(X^n+1)$. For $a \in \mathbb{Z}$, we write $[a]_q$ to denote the centered lifted representation of the reduction modulo q, i.e., the unique representation in the interval $[-(q-1)/2, (q+1)/2]$. Similarly, we can extend it for $a \in \mathcal{R}$ by taking the coefficients of a as an element in $\mathbb{Z}$ in the polynomial basis representations. For coprime integers p_1, p_2, we define $\Delta_1 := p_2\,[p_2^{-1}]_{p_1}$ and $\Delta_2 := p_1\,[p_1^{-1}]_{p_2}$, and $CRT_{p_1,p_2}(x,y) := (\Delta_1 x + \Delta_2 y) \bmod p_1p_2$, which satisfies $CRT_{p_1,p_2}(x,y) \bmod p_1 = x$ and $CRT_{p_1,p_2}(x,y) \bmod p_2 = y$. When $x, y \in \mathcal{R}$, CRT_{p_1,p_2} acts coefficientwise. In this paper, we use the Euclidean norm $\|\cdot\|$, which is defined by $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$ for some vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$.

Let g be a function where $g : \mathbb{N} \to (0,1]$. We say f is negligible if for all positive polynomials $g(\cdot)$ there exists a positive integer N such that $g(n) < \frac{1}{f(n)}$ for all $n > N$. We say a function h is overwhelming if $1 - h(n)$ is negligible. These functions are usually defined w.r.t the security parameter λ.

2.2 Probability Distribution

Given a distribution D, we write $\mathrm{Supp}(D)$ to denote its support. Let D be a distribution over S. We write $x \leftarrow D$ to sample x according to the distribution D. When S is a (finite) set, we write $x \xleftarrow{\$} S$ to denote that x is sampled uniformly from the set S.

In this work, we often consider sampling the coefficients of a polynomial f from a certain distribution D. For example, we write $f \leftarrow D$ to denote that the coefficients of f are sampled independently from D. We say a distribution D is B-bounded for a real number $B > 0$ if the $\Pr_{s \leftarrow D}[\|\mathbf{x}\| \leq B]$ is overwhelming. Let $f : X \to \mathbb{R}$ be a non-negative function, then for all countable $Y \subseteq X$, we define $f(Y) = \sum_{y \in Y} f(y) \in [0, +\infty]$.

We will use several standard distributions in this work. The centered binomial distribution with parameter $\eta \in \mathbb{Z}$ is defined as $\mathcal{B}_\eta = \{\sum_{i=0}^{\eta-1}(a_i - b_i), \forall a_i, b_i \leftarrow \{0, 1\}\}$. Its density is $\mathcal{B}_\eta(x) = \binom{2\eta}{\eta+x}/2^{2\eta}$ where $x \in [-\eta, \eta]$. The ternary distribution $\mathcal{T}_\sigma$ where $\sigma \in (0, 1/2)$ has support $\{-1, 0, 1\}$, and density $\Pr[X = -1] = \Pr[X = 1] = \sigma$ and $\Pr[X = 0] = 1 - 2\sigma$. For any vector $\mathbf{c} \in \mathbb{R}^n$ and any real $\sigma > 0$, the spherical Gaussian function with deviation parameter σ and center $\mathbf{c}$ is $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/\sigma^2)$. The spherical Gaussian distribution has density $D_{\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\sigma^n$. When $\mathbf{c} = \mathbf{0}$, we may omit the subscript $\mathbf{c}$.

2.3 Background on Chinese Remainder Theorem

Theorem 1 (Chinese Remainder Theorem). *Let $n_1, n_2, \ldots, n_k$ be pairwise coprime integers (i.e., $\gcd(n_i, n_j) = 1$ for $i \neq j$). For any given integers $a_1, a_2, \ldots, a_k$, there exists an integer x that simultaneously satisfies the system of congruences:*

$$x \equiv a_i \mod n_i, \quad i = 1, 2, ..., k$$

Moreover, the solution x is unique modulo $N = n_1 n_2 \cdots n_k$.

The CRT is useful in cryptographic applications because it allows computations to be performed independently in smaller, modular arithmetic spaces and then recombined to form the final result. This can lead to efficiencies in both computation and storage.

Definition 1 (CRT Function). *For pairwise coprime integers $p_1, \ldots, p_k$, we define the CRT function $CRT_{(p_1,\ldots,p_k)}$ for inputs $(m_1, \ldots, m_k)$ as a number in $\mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_k}$ which is congruent to m_i modulo p_i for all $i \in \{1, \ldots, k\}$, where $p = \prod_{i=1}^{k} p_i$.*

Formally, we have:

$$CRT_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k) \equiv \sum_{i=1}^{k} m_i \hat{p}_i (\hat{p}_i^{-1} \mod p_i) \mod p,$$

where

$$\hat{p}_i = \frac{p}{p_i} = \prod_{\substack{j=1 \\ j \neq i}}^{k} p_j.$$

Further for $\mathbf{a}, \mathbf{b} \in \mathbb{Z}[X]/\langle X^n+1\rangle$ wherever $\mathbf{a} = a_0+a_1x+...+a_{n-1}x^{n-1}, \mathbf{b} = b_0 + b_1x + ... + b_{n-1}x^{n-1}$ and consequently we denote

$$CRT_{p_1,p_2}(\mathbf{a}, \mathbf{b}) = CRT_{p_1,p_2}(a_0, b_0) + CRT_{p_1,p_2}(a_1, b_1)x + \dots + CRT_{p_1,p_2}(a_{n-1}, b_{n-1})x^{n-1}.$$

The ring learning with errors (RLWE) problem was introduced by Lyubashevsky, Peikert, and Regev in [23].

Definition 2 (Decision Ring-LWE Problem). *Let $f(X) = X^n + 1$ (with n a power-of-two) and define the cyclotomic ring $\mathcal{R}_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$. Fix a noise distribution χ over $\mathcal{R}_q$ and draw a secret $s \xleftarrow{\$} \mathcal{R}_q$. The* Ring-LWE distribution *$\mathcal{A}_{s,\chi}$ on $\mathcal{R}_q \times \mathcal{R}_q$ is*

$$\mathcal{A}_{s,\chi} := \Big\{ \big(a,\ a \cdot s + e\big) \ \Big|\ a \xleftarrow{\$} \mathcal{R}_q,\ e \xleftarrow{\$} \chi \Big\}.$$

For any probabilistic polynomial-time (PPT) distinguisher $\mathcal{D}$ we define the advantage *of the decision Ring-LWE problem as*

$$Adv_{\mathcal{D}}^{\mathrm{RLWE}_{n,q,\chi}} := \Big| \Pr_{a,b \leftarrow \mathcal{A}_{s,\chi}} \big[\mathcal{D}\big(a, b\big) = 1\big] - \Pr_{a,b \xleftarrow{\$} \mathcal{R}_q \times \mathcal{R}_q} \big[\mathcal{D}(a, b) = 1\big] \Big|.$$

Theorem 2 (Lyubashevsky-Peikert-Regev [23]). *For any d that is a power of 2, define the ring $\mathcal{R} = \mathbb{Z}[x]/\langle x^d + 1\rangle$. Let $q = q(d) \equiv 1 \mod d$ be a prime integer, and let $B = \omega(\sqrt{d}\log d)$. There exists an efficiently samplable distribution χ that outputs elements of $\mathcal{R}$ of length at most B with overwhelming probability. If there exists an efficient algorithm that solves $RLWE_{d,q,\chi}$, then there is an efficient quantum algorithm for solving $d^{\omega(1)} \cdot (q/B)$-approximate worst-case SVP for ideal lattices over $\mathcal{R}$.*

3 Construction

In this section, we outline the construction of our Fully Homomorphic Encryption (FHE) scheme, focusing on its key components: key generation, encryption, homomorphic operations, and decryption. Our construction builds on the principles of lattice-based cryptography, incorporating enhancements to achieve improved performance and security. First we describe the required parameters for the scheme along with their sizes, then the actual construction of the scheme and finally we prove the correctness of the scheme.

The scheme consists of the following algorithms:
CRT-FHE.Setup(1^λ): Takes as input the security parameter λ and sets up global parameters for the scheme.

- **Input:** Security Parameters λ
- **Output:** Public Parameters PP

– Define the following public parameters:
  - $q = q(\lambda)$: Ciphertext modulus.
  - $f(X) = X^n + 1$: Cyclotomic polynomial of degree $n = n(\lambda)$.
  - $\mathcal{R}_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$: Polynomial ring modulo $f(X)$ with coefficients in $\mathbb{Z}_q$.
  - $\chi = \chi(\lambda)$: Discrete noise distribution, e.g., a Gaussian or binomial distribution.
– Define auxiliary CRT moduli p_1 and p_2 satisfying:
  - $p_1 \cdot p_2 < q/2$: Ensures correctness under bounded noise growth.
  - $\gcd(p_1, p_2) = \gcd(p_1, q) = \gcd(p_2, q) = 1$: All parameters are pairwise coprime.
– Output public parameters:

$$PP = (\lambda, n, q, \chi, p_1, p_2, \mathcal{R}_q)$$

CRT-FHE.KeyGen(PP): Given the public parameters, generates keys for encryption and evaluation.

– **Input:** Public parameters PP
– **Output:** Public key $pk = (\mathbf{pk}_0, \mathbf{pk}_1)$, secret key $sk = \mathbf{s}$
– **Procedure:**
  1. Sample secret key: $\mathbf{s} \leftarrow \chi$
  2. Sample $\mathbf{a} \in \mathcal{R}_q$ uniformly at random
  3. Sample noise $\mathbf{e} \leftarrow \chi$
  4. Compute:
$$\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + CRT_{p_1,p_2}(0, \mathbf{e}) \mod q$$
  5. Set:
$$pk = (\mathbf{pk}_0, \mathbf{pk}_1) = (\mathbf{b}, -\mathbf{a}), \quad sk = \mathbf{s}$$

CRT-FHE.RelinKeyGen(sk): Generates the relinearization key used to reduce ciphertext degree after multiplication.

– Let $w = 2^r$ be the decomposition base, and let $\ell = \lfloor \log_w(q) \rfloor$ denote the number of base-w digits needed to represent ciphertext coefficients.
– For each $i = 0, 1, \ldots, \ell$, perform:
  1. Sample noise polynomials: $\mathbf{e}_i, \mathbf{e}'_i \leftarrow \chi$
  2. Sample uniform polynomial: $\mathbf{a}_i \leftarrow \mathcal{R}_q$
  3. Compute:
$$r_{i,0} = [\mathbf{a}_i \cdot \mathbf{s} + w^i \cdot \mathbf{s}^2 + \mathrm{CRT}_{p_1,p_2}(0, \mathbf{e}_i)]_q$$
$$r_{i,1} = [-\mathbf{a}_i + \mathrm{CRT}_{p_1,p_2}(0, \mathbf{e}'_i)]_q$$
– The evaluation key is:
$$\mathsf{evk} = \{(r_{i,0}, r_{i,1})\}_{i=0}^{\ell}$$

CRT-FHE.Enc(pk, $\mathbf{m} \in \mathbb{Z}_{p_1}^n$): Encrypts a plaintext vector into a degree-1 ciphertext.

– **Input:** Public key $pk = (\mathbf{pk}_0, \mathbf{pk}_1)$, message vector $\mathbf{m} = (m_0, \ldots, m_{n-1}) \in \mathbb{Z}_{p_1}^n$
– **Output:** Ciphertext $\mathbf{c} = (c_0, c_1) \in \mathcal{R}_q^2$
– **Procedure:**
  1. Encode plaintext as polynomial in R_{p_1}:
  $$\mathbf{m}(x) = m_0 + m_1 x + \cdots + m_{n-1} x^{n-1}$$
  2. Sample $\mathbf{u}, \mathbf{e}', \mathbf{e}'' \leftarrow \chi$
  3. Compute:
  $$c_0 = [\mathbf{pk}_0 \cdot \mathbf{u} + \mathrm{CRT}_{p_1,p_2}(\mathbf{m}, \mathbf{e}')]_q$$
  $$c_1 = [\mathbf{pk}_1 \cdot \mathbf{u} + \mathrm{CRT}_{p_1,p_2}(0, \mathbf{e}'')]_q$$
  4. Output ciphertext $\mathbf{c} = (c_0, c_1)$

CRT-FHE.Dec(sk, $\mathbf{c} = (c_0, c_1)$): Decrypts a ciphertext to recover the original message.

– **Input:** Secret key $\mathbf{s}$, ciphertext $\mathbf{c} = (c_0, c_1) \in \mathcal{R}_q^2$
– **Output:** Plaintext $\mathbf{m} \in R_{p_1}$
– **Procedure:**
  1. Compute:
  $$\mathbf{m} = [(c_0 + c_1 \cdot \mathbf{s}) \bmod q] \bmod p_1$$
  2. Output $\mathbf{m}$

We refer to Section 4 for additional details on the noise bound required to ensure correct decryption.

Homomorphic Operations: With the encryption, decryption, and relinearization procedures defined, the CRT-FHE scheme supports essential homomorphic operations on ciphertexts. These include homomorphic addition and multiplication, allowing computations to be performed directly over encrypted data without requiring decryption. The following subsections describe these operations in detail.

Homomorphic Addition. Let $\mathbf{c}_1 = (c_0^1, c_1^1)$ and $\mathbf{c}_2 = (c_0^2, c_1^2)$ be two ciphertexts in $\mathcal{R}_q^2$. The homomorphic addition is performed component-wise:

$$\mathbf{c}_{\text{add}} = (c_0^1 + c_0^2 \bmod q,\ c_1^1 + c_1^2 \bmod q).$$

This results in a ciphertext $\mathbf{c}_{\text{add}}$ that encrypts the sum of the plaintexts.

Homomorphic Multiplication Ciphertext multiplication follows the standard RLWE pattern, producing a degree-2 ciphertext that is subsequently relinearized. When batch encoding is used, multiplication is performed slot-wise on the packed plaintexts; correctness follows from the ring-homomorphic property of the encoding map. We refer to Section 3.1 for more details. Let $\mathbf{c}_1 = (c_0^1, c_1^1)$ and $\mathbf{c}_2 = (c_0^2, c_1^2)$ be ciphertexts encrypting plaintexts $\mathbf{m}_1$ and $\mathbf{m}_2$ respectively. Their product is computed as:

$$\mathbf{c}_{\text{mult}} = (c_0, c_1, c_2) \in \mathcal{R}_q^3, \text{ where}$$

$$c_0 = c_0^1 \cdot c_0^2, \quad c_1 = c_0^1 \cdot c_1^2 + c_1^1 \cdot c_0^2, \quad c_2 = c_1^1 \cdot c_1^2 \tag{1}$$

The resulting ciphertext $\mathbf{c}^*_{\text{mult}}$ encrypts the product $\mathbf{m}_1 \cdot \mathbf{m}_2$, but is of degree 2 in the secret key. To reduce its size and enable further operations, relinearization is applied:

Relinearization Decompose c_2 in base w:

$$c_2 = \sum_{i=0}^{\ell} c_i^2 \cdot w^i$$

Let $\mathsf{evk} = \{(\mathbf{r}_0^i, \mathbf{r}_1^i)\}_{i=0}^{\ell}$ be the evaluation key containing encryption of s^2 scaled by w^i. Then relinearization is computed as:

$$c_0' = c_0 + \sum_{i=0}^{\ell} c_i^2 \cdot \mathbf{r}_0^i, \quad c_1' = c_1 + \sum_{i=0}^{\ell} c_i^2 \cdot \mathbf{r}_1^i$$

The relinearized ciphertext $\mathbf{c}_{\text{mult}} = (c_0', c_1') \in \mathcal{R}_q^2$ is now of degree 1, encrypting $\mathbf{m}_1 \cdot \mathbf{m}_2$ and compatible with further homomorphic operations.

3.1 Batch Encoding and SIMD Operations

Batching is a powerful technique in FHE that allows multiple plaintexts to be encrypted and processed simultaneously within a single ciphertext. This approach significantly enhances the efficiency of homomorphic computations, making FHE more practical for real-world applications. The concept of batching leverages the structure of certain algebraic objects, such as rings and polynomials, to pack multiple messages into one.

Mathematical Foundation To enable coordinate-wise multiplication over plaintext vectors, the underlying polynomial multiplication must be transformed. This is achieved using the Number Theoretic Transform (NTT) and its inverse (INTT), which map the plaintexts into a domain where polynomial multiplication corresponds to element-wise vector multiplication. The following section outlines the mathematical principles supporting this transformation. The current form of the ciphertext:

$$c_0 + c_1 s = \mathbf{m}^* = CRT_{p_1,p_2}(m_0, e_0) + CRT_{p_1,p_2}(m_1, e_1)x + ... + CRT_{p_1,p_2}(m_{n-1}, e_{n-1})x^{n-1}$$

We rewrite it to express the plaintext and noise separately as follows

$$\begin{aligned}
&\mathrm{CRT}_{p_1,p_2}(m_0,e_0) + \mathrm{CRT}_{p_1,p_2}(m_1,e_1)\cdot x + \ldots + \mathrm{CRT}_{p_1,p_2}(m_{n-1},e_{n-1})\cdot x^{n-1} \\
&= \sum_{i=0}^{n-1} \mathrm{CRT}_{p_1,p_2}(m_i,0)\cdot x^i + \sum_{i=0}^{n-1} \mathrm{CRT}_{p_1,p_2}(0,e_i)\cdot x^i \\
&= \sum_{i=0}^{n-1} p_2\cdot[p_2^{-1}]_{p_1}\cdot m_i\cdot x^i + \sum_{i=0}^{n-1} p_1\cdot[p_1^{-1}]_{p_2}\cdot e_i\cdot x^i \\
&= \big(p_2\cdot[p_2^{-1}]_{p_1}\big)\cdot\sum_{i=0}^{n-1} m_i\cdot x^i + \big(p_1\cdot[p_1^{-1}]_{p_2}\big)\cdot\sum_{i=0}^{n-1} e_i\cdot x^i \\
&= \Delta_1\cdot\mathbf{m} + \Delta_2\cdot\mathbf{e}
\end{aligned}$$

Let $\mathbf{c}_1 = (c_0^1, c_1^1)$ and $\mathbf{c}_2 = (c_0^2, c_1^2)$ be two ciphertexts that encrypt plaintexts μ_1 and μ_2, respectively.

$$\begin{aligned}
c_0^j + c_1^j s = \mathbf{m}_i^* = CRT_{p_1,p_2}(m_0^j, e_0^j) + CRT_{p_1,p_2}(m_1^j, e_1^j)x + ... \\
+ CRT_{p_1,p_2}(m_{n-1}^j, e_{n-1}^j)x^{n-1}
\end{aligned}$$

Now when two ciphertexts are multiplied the corresponding polynomials $\sum_{i=0}^{n-1} m_i^1 x^i$ and $\sum_{i=0}^{n-1} m_i^2 x^i$ were multiplied.

$$\mathrm{Dec}(\mathbf{c}_{\mathrm{mult}}, sk) = \mathbf{m}_1^*.\mathbf{m}_2^* = (\sum_{i=0}^{n-1} m_i^1 x^i)(\sum_{i=0}^{n-1} m_i^2 x^i)$$

This does not give use the coordinate-wise multiplication of the two vectors i.e.

$$(m_0^1, m_1^1, ..., m_{n-1}^1) * (m_0^2, m_1^2, ..., m_{n-1}^2) \rightarrow (m_0^1\cdot m_0^2, m_1^1\cdot m_1^2, \ldots, m_{n-1}^1\cdot m_{n-1}^2)$$

This can be achieved using the concept of NTT as follows:

$$\begin{aligned}
&\mathrm{INTT}(m_0^1, m_1^1, \ldots, m_{n-1}^1) = \boldsymbol{\mu}_1 = \mu_{1,0} + \mu_{1,1}x + \cdots + \mu_{1,n-1}x^{n-1} \\
&\mathrm{INTT}(m_0^2, m_1^2, \ldots, m_{n-1}^2) = \boldsymbol{\mu}_2 = \mu_{2,0} + \mu_{2,1}x + \cdots + \mu_{2,n-1}x^{n-1} \\
&\boldsymbol{\mu}_1\cdot\boldsymbol{\mu}_2 = \mathrm{INTT}(m_0^1, m_1^1, \ldots, m_{n-1}^1)\cdot\mathrm{INTT}(m_0^2, m_1^2, \ldots, m_{n-1}^2) \\
&\qquad = \mathrm{INTT}(m_0^1\cdot m_0^2, m_1^1\cdot m_1^2, \ldots, m_{n-1}^1\cdot m_{n-1}^2)
\end{aligned}$$

In order to achieve this instead of directly putting plaintext values in coefficient, we need to compute the INTT of the plaintext vector and get a polynomial and we encrypt the new polynomial.

Encoding To batch multiple plaintext messages $\mathbf{m} = (m_0, m_1, \ldots, m_{n-1})$, each $m_i \in \mathbb{Z}_{p_1}$, into a single polynomial, we perform the following steps:

– Compute the Inverse NTT of the plaintext as

$$\mu = INTT(\mathbf{m}) = \mu_0 + \mu_1 x + \ldots + \mu_{n-1}x^{n-1}$$

The $INTT()$ function is performed in the space Z_{p_1} so that the result $\mu \in Z_{p_1}[X]$.

– The resulting polynomial is then encoded as

$$\mu^* = CRT_{p_1,p_2}(\mu_0, 0) + CRT_{p_1,p_2}(\mu_1, 0)x + ... + CRT_{p_1,p_2}(\mu_{n-1}, 0)x^{n-1}$$

Decoding To decode the batched ciphertext back into individual plaintexts:

- Apply $\mod p_1$ to separate plaintext and noise.

$$\mu^* \mod p_1$$

$$= CRT_{p_1,p_2}(\mu_0, e_0) + CRT_{p_1,p_2}(\mu_1, e_1)x + ... + CRT_{p_1,p_2}(\mu_{n-1}, e_{n-1})x^{n-1} \mod p_1$$

$$= \mu_0 + \mu_1 x + \ldots + \mu_{n-1}x^{n-1} = \mu$$

- Use the NTT function to transform the polynomial in $\mathbb{Z}_{p_1}[X]$ back into a tuple of elements in $\mathbb{Z}_{p_1}$.

$$NTT(\mu) = \mathbf{m} = (m_0, m_1, ..., m_{n-1})$$

- Extract the individual plaintext messages from the resulting polynomial.

Homomorphic Operations on Batches The primary advantage of batching is that homomorphic operations on the ciphertexts can be performed in parallel on the individual plaintexts. For instance:

Addition:

- Given two batched ciphertexts c_1 and c_2 encoding plaintexts $\mathbf{m_1} = m_1^1, m_2^1, \ldots, m_k^1)$ and $\mathbf{m_2} = (m_1^2, m_2^2, \ldots, m_k^2)$, respectively, their homomorphic addition results in a ciphertext encoding $\mathbf{m_1} + \mathbf{m_2} = (m_1^1 + m_1^2, m_2^1 + m_2^2, \ldots, m_k^1 + m_k^2)$.

Multiplication:

- Similarly, homomorphic multiplication of two batched ciphertexts results in a ciphertext encoding the product of the corresponding plaintexts:

$$\mathbf{m_1} \times \mathbf{m_2} = (m_1^1 \times m_1^2, m_2^1 \times m_2^2, \ldots, m_k^1 \times m_k^2)$$

Batching in FHE schemes offers a powerful means to enhance the efficiency of homomorphic computations by leveraging the parallelism in polynomial rings. By packing multiple plaintexts into a single ciphertext, batching allows for significant improvements in computational and memory efficiency, making FHE more practical for a wider range of applications.

3.2 Noise Growth Behavior in CRT Ciphertexts

We analyze the growth of decryption noise for ciphertexts that use the CRT embedding

$$CRT_{p_1,p_2}(m, e) = \Delta_1 m + \Delta_2 e \bmod p, \qquad p = p_1 p_2, \;\; \Delta_1 = p_2[p_2^{-1}]_{p_1}, \;\; \Delta_2 = p_1[p_1^{-1}]_{p_2}.$$

Our analysis follows the methodology of [3,11,17], adapted to account for the two-modulus CRT structure.

Rings, norms, and tails. Let $R = \mathbb{Z}[X]/(X^n + 1)$ with n a power of two, and $R_q = R/qR$. Denote by $\sigma_1, \ldots, \sigma_n : R \to \mathbb{C}$ the canonical embeddings, and write

$$\|x\|_{\mathrm{can}} = \Big(\sum_{i=1}^{n} |\sigma_i(x)|^2 \Big)^{1/2}, \qquad \|x\|_\infty = \max_j |x_j| \text{ (coefficient } \ell_\infty).$$

There exist fully explicit ring-dependent constants $\kappa_{\infty \leftarrow \mathrm{can}}, \kappa_{\mathrm{can} \leftarrow \infty} > 0$ such that for all $x \in R$ (see, e.g., [24]):

$$\|x\|_\infty \le \kappa_{\infty \leftarrow \mathrm{can}} \cdot \|x\|_{\mathrm{can}} \quad \text{and} \quad \|x\|_{\mathrm{can}} \le \kappa_{\mathrm{can} \leftarrow \infty} \cdot \sqrt{n}\, \|x\|_\infty. \tag{2}$$

Throughout we assume that all error terms are drawn independently from a sub-gaussian distribution χ over R with parameter σ (e.g., discrete Gaussian), and that s, u are either sampled from χ or from a bounded small-secret distribution. For $e \leftarrow \chi$ and any $\alpha \in R$,

$$\Pr[\, |\langle \alpha, e \rangle| > t\, \sigma \|\alpha\|_2 \,] \le 2e^{-\pi t^2} \qquad (\forall t > 0). \tag{3}$$

In particular, for any fixed failure probability $\varepsilon \in (0, 1)$, with probability at least $1 - \varepsilon$ we have

$$\|e\|_{\mathrm{can}} \le \tau(\varepsilon)\, \sigma \sqrt{n} \quad \text{with} \quad \tau(\varepsilon) = \sqrt{2 \log\big(2n/\varepsilon\big)}. \tag{4}$$

Critical quantity and centered CRT-error. Given a ciphertext $c = (c_0, c_1) \in R_q^2$, define its *critical quantity*

$$v(c) = \big[c_0 + c_1 s\big]_q \in R_q,$$

where $[\cdot]_q$ denotes coefficient-wise centered reduction to $(-\frac{q}{2}, \frac{q}{2}]$. If c encrypts $m \in R_{p_1}$ under the CRT embedding, then there exists an (integral) *centered CRT-error* $\eta(c) \in R$ such that, as integers,

$$v(c) = \Delta_1 m + \Delta_2 \eta(c) \pmod q. \tag{5}$$

We will measure noise via $\|\eta(c)\|_{\mathrm{can}}$, which isolates the contribution that can cause decryption failure.

Lemma 1 (Correctness threshold). *Let $M_\infty := \max_{m \in R_{p_1}} \|\Delta_1 m\|_\infty$. If*

$$\kappa_{\infty \leftarrow \mathrm{can}} \cdot \|\Delta_2 \eta(c)\|_{\mathrm{can}} \;<\; \frac{q}{2} - M_\infty,$$

then decryption of c returns m.

Proof. By (2), $\|x\|_\infty \le \kappa_{\infty \leftarrow \mathrm{can}} \|x\|_{\mathrm{can}}$ for all $x \in R$. If $\|\Delta_1 m + \Delta_2 \eta(c)\|_\infty < q/2$, then the centered lift of $v(c)$ equals the integer polynomial $\Delta_1 m + \Delta_2 \eta(c)$, hence $v(c) \bmod p = \Delta_1 m + \Delta_2 \eta(c) \bmod p$. Reducing modulo p_1 eliminates the Δ_2 component and yields m, because $\Delta_1 \equiv 1 \pmod{p_1}$ and $\Delta_2 \equiv 0 \pmod{p_1}$. The stated bound suffices to ensure $\|\Delta_1 m + \Delta_2 \eta(c)\|_\infty < q/2$ for all m.

Theorem 3 (Correctness of Decryption). *Let $m \in R_{p_1}$ be a plaintext and let $(c_0, c_1) = \mathit{CRT\text{-}FHE.Enc}(pk, m)$ be a fresh ciphertext generated as in Section 3. Let $M_\infty := \max_{m' \in R_{p_1}} \|\Delta_1 m'\|_\infty$. Then decryption recovers m, i.e.,*

$$\mathit{CRT\text{-}FHE.Dec}(sk, (c_0, c_1)) = m$$

provided the accumulated noise $\eta(c)$ satisfies

$$\kappa_{\infty \leftarrow can} \cdot |\Delta_2| \cdot \|\eta(c)\|_{can} < \frac{q}{2} - M_\infty.$$

Proof. From the encryption equations, we have:

$$\begin{aligned} c_0 &= [pk_0 \cdot u + \mathrm{CRT}_{p_1,p_2}(m, e')]_q \\ &= [(a \cdot s + \mathrm{CRT}_{p_1,p_2}(0, e)) \cdot u + \mathrm{CRT}_{p_1,p_2}(m, e')]_q \\ &= [a \cdot u \cdot s + \mathrm{CRT}_{p_1,p_2}(m, e \cdot u + e')]_q, \end{aligned}$$

and

$$c_1 = [pk_1 \cdot u + \mathrm{CRT}_{p_1,p_2}(0, e'')]_q = [-a \cdot u + \mathrm{CRT}_{p_1,p_2}(0, e'')]_q.$$

The decryption algorithm computes the critical quantity:

$$v(c) = [c_0 + c_1 \cdot s]_q.$$

Substituting the expressions for c_0 and c_1:

$$\begin{aligned} c_0 + c_1 \cdot s &= [a \cdot u \cdot s + \mathrm{CRT}_{p_1,p_2}(m, e \cdot u + e')]_q + [-a \cdot u \cdot s + \mathrm{CRT}_{p_1,p_2}(0, e'') \cdot s]_q \\ &= [\mathrm{CRT}_{p_1,p_2}(m, e \cdot u + e') + \mathrm{CRT}_{p_1,p_2}(0, e'' \cdot s)]_q \\ &= [\mathrm{CRT}_{p_1,p_2}(m, e \cdot u + e' + e'' \cdot s)]_q. \end{aligned}$$

Define the accumulated noise polynomial:

$$\eta(c) := e \cdot u + e' + e'' \cdot s \in R.$$

By the CRT structure, we have:

$$\mathrm{CRT}_{p_1,p_2}(m, \eta(c)) = \Delta_1 \cdot m + \Delta_2 \cdot \eta(c) \in R,$$

where $\Delta_1 = p_2[p_2^{-1}]_{p_1}$ and $\Delta_2 = p_1[p_1^{-1}]_{p_2}$ satisfy:

$$\Delta_1 \equiv 1 \pmod{p_1}, \quad \Delta_1 \equiv 0 \pmod{p_2}, \quad \Delta_2 \equiv 0 \pmod{p_1}, \quad \Delta_2 \equiv 1 \pmod{p_2}.$$

Therefore, as an integer polynomial before modular reduction:

$$c_0 + c_1 \cdot s = \Delta_1 \cdot m + \Delta_2 \cdot \eta(c) + q \cdot k$$

for some $k \in R$.

By Lemma 1, if

$$\kappa_{\infty \leftarrow \mathrm{can}} \cdot \|\Delta_2 \cdot \eta(c)\|_{\mathrm{can}} < \frac{q}{2} - M_\infty,$$

then $\|\Delta_1 \cdot m + \Delta_2 \cdot \eta(c)\|_\infty < q/2$, which ensures that the centered reduction $[c_0 + c_1 \cdot s]_q$ equals the integer polynomial $\Delta_1 \cdot m + \Delta_2 \cdot \eta(c)$.

Consequently, the decryption computation yields:

$$\mathrm{Dec}(\mathrm{sk}, (c_0, c_1)) = [[c_0 + c_1 \cdot s]_q]_{p_1} = [\Delta_1 \cdot m + \Delta_2 \cdot \eta(c)]_{p_1} = m,$$

where the last equality follows from $\Delta_1 \equiv 1 \pmod{p_1}$ and $\Delta_2 \equiv 0 \pmod{p_1}$.

Remark 1 (Fresh Encryption Noise). For a fresh ciphertext, the noise components e, e', e'' are drawn independently from χ. If χ is subgaussian with parameter σ, then with probability at least $1 - 3\varepsilon$:

$$\|\eta_{\mathrm{fresh}}\|_{\mathrm{can}} \leq \tau(\varepsilon)^2 \sigma^2 n + 2\tau(\varepsilon)\sigma\sqrt{n},$$

where $\tau(\varepsilon) = \sqrt{2\log(2n/\varepsilon)}$. This bound, combined with Theorem 3, determines the initial noise budget available for homomorphic operations.

For decryption, Lemma 1 gives a *uniform* condition that is convenient for parameter setting:

$$\kappa_{\infty\leftarrow\mathrm{can}} \cdot |\Delta_2| \cdot \rho(c) \;<\; \frac{q}{2} - M_\infty. \tag{6}$$

This motivates the following:

Definition 3 (CRT noise budget). *For a ciphertext c at modulus q, define*

$$\mathit{Budget}(c) \;:=\; \log_2\!\left(\frac{q/2 - M_\infty}{\kappa_{\infty\leftarrow\mathrm{can}}\,|\Delta_2|\,\rho(c)}\right).$$

Correctness holds whenever $\mathit{Budget}(c) > 0$. Each homomorphic operation updates $\rho(\cdot)$ according to the recurrences above, and modulus switching contracts both message scale and $\rho(\cdot)$ by a factor $\approx q'/q$ up to an additive rounding term.

Remarks on instantiation. (i) The constants $\kappa_{\infty\leftarrow\mathrm{can}}, \kappa_{\mathrm{can}\leftarrow\infty}$ depend only on R and can be bounded tightly for power-of-two cyclotomics [24]. (ii) If s, u are bounded small secrets, replace the $\tau(\varepsilon)\sigma\sqrt{n}$ factors by the corresponding high-probability bounds for $\|s\|_{\mathrm{can}}, \|u\|_{\mathrm{can}}$; this typically improves the fresh noise bound. (iii) Choosing the modulus chain so that each multiplication is followed by modulus switching with factor $s \approx \Delta_1$ (and relinearization base B with moderate Γ_B) yields the same asymptotic noise behavior as in BGV/BFV: additive growth under addition, multiplicative/cross-term growth under multiplication, an additive $\tilde{O}(\sigma\sqrt{n\ell})$ from relinearization, and contraction by q'/q under modulus switching.

4 Noise Growth and Global Correctness Bound

In this section we analyze how the decryption error evolves under the CRT-FHE operations and derive a global bound that prevents wrap-around modulo the ciphertext modulus. Throughout, we work with a prime-power plaintext modulus p_1^r and an auxiliary CRT modulus p_2 coprime to p_1, and set

$$p := p_1^r p_2.$$

Let $R = \mathbb{Z}[X]/(X^n+1)$ and $R_q = \mathbb{Z}_q[X]/(X^n+1)$. We use the canonical embedding norm $\|\cdot\|_{\mathrm{can}}$ and the coefficient-wise $\|\cdot\|_\infty$, with constants

$$\|x\|_\infty \;\le\; \kappa_{\infty\leftarrow\mathrm{can}}\,\|x\|_{\mathrm{can}}, \qquad \|x\|_{\mathrm{can}} \;\le\; \kappa_{\mathrm{can}\leftarrow\infty}\,\sqrt{n}\,\|x\|_\infty, \tag{7}$$

for all $x \in R$.

4.1 CRT decomposition and global error

For coprime p_1^r and p_2, define

$$\Delta_1 := p_2[p_2^{-1}]_{p_1^r}, \qquad \Delta_2 := p_1^r[(p_1^r)^{-1}]_{p_2},$$

and

$$\mathrm{CRT}_{p_1^r,p_2}(m,e) := (\Delta_1 m + \Delta_2 e) \bmod p,$$

so that

$$\mathrm{CRT}_{p_1^r,p_2}(m,e) \bmod p_1^r = m, \qquad \mathrm{CRT}_{p_1^r,p_2}(m,e) \bmod p_2 = e.$$

Let $c = (c_0, c_1) \in R_q^2$ be a ciphertext under secret key $s \in R_q$, and define its centered decryption value

$$v(c) := \langle c_0 + c_1 s \rangle_q \in R,$$

where $\langle \cdot \rangle_q$ denotes coefficient-wise reduction to $(-q/2, q/2]$ followed by canonical lifting to R. In the no-wrap regime, there exist *plaintext* $m(c) \in R_{p_1^r}$, *CRT error* $\eta(c) \in R_{p_2}$, and a coarse integer component $k(c) \in R$ such that

$$v(c) \;=\; p\,k(c) + \mathrm{CRT}_{p_1^r, p_2}(m(c), \eta(c)) \;=\; p\,k(c) + \Delta_1 m(c) + \Delta_2 \eta(c). \tag{8}$$

Since p and Δ_2 are multiples of p_1^r, reducing modulo p_1^r eliminates both the coarse term and the CRT error:

$$v(c) \bmod p_1^r = m(c),$$

so decryption recovers $m(c)$ as long as no wrap-around occurs modulo q.

It is convenient to factor out the *global decryption error*

$$E(c) \;:=\; \Delta_2 \eta(c) + p\,k(c) \in R,$$

so that

$$v(c) \;=\; \Delta_1 m(c) + E(c), \qquad E(c) \equiv 0 \pmod{p_1^r}. \tag{9}$$

We will track the canonical norm

$$\Lambda(c) := \|E(c)\|_{\mathrm{can}}$$

as our global noise measure. Note that $E(c)$ contains both the "fine" CRT error and the coarse $p_1^r p_2$ multiple.

4.2 Global correctness (no-wrap) condition

Assume that messages are a priori bounded in canonical norm by a fixed constant

$$\|m(c)\|_{\mathrm{can}} \le M_{\mathrm{msg}} \qquad \text{for all ciphertexts } c \text{ in the supported computation.}$$

(This bound depends on the chosen plaintext space and the maximum arithmetic depth; it is fixed at parameter selection.)

From (9) and (7),

$$\|v(c)\|_{\mathrm{can}} \;\le\; |\Delta_1|\, M_{\mathrm{msg}} + \Lambda(c),$$

and hence

$$\|v(c)\|_\infty \;\le\; \kappa_{\infty \leftarrow \mathrm{can}}\big(|\Delta_1|\, M_{\mathrm{msg}} + \Lambda(c)\big). \tag{10}$$

Decryption correctness (no wrap-around modulo q) is guaranteed whenever $|v(c)_i| < q/2$ coefficient-wise. Combining with (10), we obtain:

Lemma 2 (Global correctness bound). *Let c be any ciphertext at modulus q, with message bound M_{msg} and global error $\Lambda(c) = \|E(c)\|_{\mathrm{can}}$. If*

$$\kappa_{\infty \leftarrow \mathrm{can}}\big(|\Delta_1|\, M_{\mathrm{msg}} + \Lambda(c)\big) < \frac{q}{2}, \tag{11}$$

then decryption of c returns $m(c) \in R_{p_1^r}$.

This motivates the following definition.

Definition 4 (Global noise budget). *For a ciphertext c at modulus q, define*

$$\mathrm{Budget}(c) \;:=\; \log_2\left(\frac{q/2}{\kappa_{\infty\leftarrow\mathrm{can}}\big(|\Delta_1|\,M_{\mathrm{msg}} + \Lambda(c)\big)}\right).$$

Correctness holds as long as $\mathrm{Budget}(c) > 0$. *Homomorphic operations update $\Lambda(c)$ according to the recurrences below, and modulus switching shrinks both $|\Delta_1|\,M_{\mathrm{msg}}$ and $\Lambda(c)$ by a factor q'/q up to a rounding term.*

4.3 Noise growth under basic operations

We now bound $\Lambda(c)$ for fresh encryptions, additions, and multiplications. The bounds are in canonical norm and therefore interact well with the multiplicative property of the canonical embedding,

$$\|xy\|_{\mathrm{can}} \le \|x\|_{\mathrm{can}}\,\|y\|_{\mathrm{can}} \qquad (\forall x, y \in R). \tag{12}$$

Fresh encryption. Let the public key be $(b, -a)$ with $b = as + \mathrm{CRT}_{p_1^r,p_2}(0, e)$ and let a fresh ciphertext use $u, e', e'' \leftarrow \chi$:

$$c_0 = bu + \mathrm{CRT}_{p_1^r,p_2}(m, e'), \qquad c_1 = -au + \mathrm{CRT}_{p_1^r,p_2}(0, e'').$$

A direct cancellation gives

$$v(c) = c_0 + c_1 s = \mathrm{CRT}_{p_1^r,p_2}\left(m, \eta_{\mathrm{fresh}}\right) = \Delta_1 m + \Delta_2 \eta_{\mathrm{fresh}},$$

where

$$\eta_{\mathrm{fresh}} = eu + e' + e''s. \tag{13}$$

Thus $k(c) = 0$ and $E(c) = \Delta_2\eta_{\mathrm{fresh}}$, so

$$\Lambda(c) = \|E(c)\|_{\mathrm{can}} = |\Delta_2|\,\|\eta_{\mathrm{fresh}}\|_{\mathrm{can}}.$$

Using (12) and the triangle inequality,

$$\|\eta_{\mathrm{fresh}}\|_{\mathrm{can}} \le \|e\|_{\mathrm{can}}\|u\|_{\mathrm{can}} + \|e'\|_{\mathrm{can}} + \|e''\|_{\mathrm{can}}\|s\|_{\mathrm{can}}. \tag{14}$$

If e, e', e'', s, u are sampled from a subgaussian distribution with parameter σ (or from a small-secret distribution) and we denote by $\tau(\varepsilon)\sigma\sqrt{n}$ a high-probability bound on their canonical norms, then with probability at least $1 - \varepsilon$,

$$\Lambda_{\mathrm{fresh}} \;\le\; |\Delta_2|\Big(\tau(\varepsilon)^2\sigma^2 n + 2\tau(\varepsilon)\sigma\sqrt{n}\Big), \tag{15}$$

or better if s, u are chosen from a more concentrated distribution.

Addition. Let $c^{(1)}, c^{(2)}$ encrypt messages m_1, m_2 with global errors E_1, E_2 and

$$v\big(c^{(j)}\big) = \Delta_1 m_j + E_j, \qquad \Lambda_j = \|E_j\|_{\mathrm{can}}, \quad j \in \{1, 2\}.$$

The sum $c^{(+)} = c^{(1)} + c^{(2)}$ encrypts $m_+ = m_1 + m_2$ with

$$v(c^{(+)}) = v\big(c^{(1)}\big) + v\big(c^{(2)}\big) = \Delta_1(m_1 + m_2) + (E_1 + E_2),$$

so $E_+ = E_1 + E_2$ and

$$\Lambda(c^{(+)}) = \|E_1 + E_2\|_{\mathrm{can}} \;\le\; \Lambda_1 + \Lambda_2. \tag{16}$$

Multiplication (pre-relinearization). Let $c^{(1)}, c^{(2)}$ encrypt m_1, m_2 with

$$v\big(c^{(j)}\big) = \Delta_1 m_j + E_j, \qquad \Lambda_j = \|E_j\|_{\mathrm{can}}.$$

Before relinearization, the degree-2 ciphertext has critical polynomial

$$g(X) = \big(c_0^{(1)} + c_1^{(1)} s\big)\big(c_0^{(2)} + c_1^{(2)} s\big),$$

and hence

$$g(s) = v\big(c^{(1)}\big)\, v\big(c^{(2)}\big) = (\Delta_1 m_1 + E_1)(\Delta_1 m_2 + E_2).$$

We want to rewrite this in the form

$$g(s) = \Delta_1 m_{\mathrm{mult}} + E_{\mathrm{mult}}, \qquad m_{\mathrm{mult}} = m_1 m_2,$$

to match the semantics of ciphertext multiplication.

Expanding and using the CRT identity

$$\Delta_1^2 = k' p + \Delta_1 \quad \text{for some fixed } k' \in R,$$

we obtain

$$\begin{aligned} g(s) &= \Delta_1^2 m_1 m_2 + \Delta_1(m_1 E_2 + m_2 E_1) + E_1 E_2 \\ &= \Delta_1 m_1 m_2 + \underbrace{p\, k' m_1 m_2 + \Delta_1(m_1 E_2 + m_2 E_1) + E_1 E_2}_{E_{\mathrm{mult}}}. \end{aligned} \tag{17}$$

Thus the *global error* after multiplication (before relinearization) is

$$E_{\mathrm{mult}} = p\, k' m_1 m_2 + \Delta_1(m_1 E_2 + m_2 E_1) + E_1 E_2,$$

and by (12),

$$\begin{aligned} \Lambda_{\mathrm{mult}}^{\mathrm{pre}} &:= \|E_{\mathrm{mult}}\|_{\mathrm{can}} \\ &\le |pk'|_{\mathrm{can}} \cdot \|m_1\|_{\mathrm{can}} \|m_2\|_{\mathrm{can}} + |\Delta_1|\big(\|m_1\|_{\mathrm{can}} \Lambda_2 + \|m_2\|_{\mathrm{can}} \Lambda_1\big) + \Lambda_1 \Lambda_2 \\ &\le C_p\, M_{\mathrm{msg}}^2 + |\Delta_1| M_{\mathrm{msg}}(\Lambda_1 + \Lambda_2) + \Lambda_1 \Lambda_2, \end{aligned} \tag{18}$$

where $|pk'|_{\mathrm{can}}$ is the canonical norm of the fixed ring element pk' and we set $C_p := |pk'|_{\mathrm{can}}$. This bound already incorporates the contribution of the coarse p-multiple (which was previously tracked via k_{mult}) into the single global error term E_{mult}.

4.4 Relinearization

Let $g(s) = d_0 + d_1 s + d_2 s^2$ be the degree-2 ciphertext before key switching, and assume a standard gadget-based relinearization with base w, length $\ell = \lceil \log_w q \rceil$, and evaluation key

$$\mathrm{EK}_i = (a_i,\ b_i = a_i s + e_i - w^i s^2) \in R_q^2, \qquad e_i \leftarrow \chi.$$

Writing $g_w^{-1}(d_2) = (\delta_0, \ldots, \delta_{\ell-1})$ for the gadget decomposition, the relinearized ciphertext replaces $d_2 s^2$ by $\sum_i \delta_i \mathrm{EK}_i$, injecting additional error $\sum_i \delta_i e_i$ into the decryption expression. Define the gadget constant

$$\Gamma_w := \sup_{x \neq 0} \frac{\|g_w^{-1}(x)\|_{2,\mathrm{can}}}{\|x\|_\infty},$$

which depends only on w, ℓ, R. A standard Cauchy–Schwarz argument (as in RLWE-FHE analyses) gives, with probability $\geq 1 - \varepsilon$,

$$\Big\| \sum_{i=0}^{\ell-1} \delta_i e_i \Big\|_{\mathrm{can}} \;\leq\; \Gamma_w \, \|d_2\|_\infty \cdot \tau(\varepsilon)\, \sigma \sqrt{n\ell}. \tag{19}$$

Thus, if $\Lambda_{\mathrm{mult}}^{\mathrm{pre}}$ is the pre-relinearization global error, the *post-relinearization* error satisfies

$$\Lambda_{\mathrm{mult}}^{\mathrm{post}} \;\leq\; \Lambda_{\mathrm{mult}}^{\mathrm{pre}} + \Gamma_w \, \|d_2\|_\infty \cdot \tau(\varepsilon)\, \sigma \sqrt{n\ell}. \tag{20}$$

4.5 Summary of global noise recurrences

Let $\Lambda(c) = \|E(c)\|_{\mathrm{can}}$ be the global decryption error defined in (9), and assume $\|m(c)\|_{\mathrm{can}} \leq M_{\mathrm{msg}}$ for all ciphertexts in the supported computation. Then:

- **Fresh encryption:** with probability $\geq 1 - \varepsilon$,

$$\Lambda_{\mathrm{fresh}} \;\leq\; |\Delta_2| \Big(\tau(\varepsilon)^2 \sigma^2 n + 2\tau(\varepsilon) \sigma \sqrt{n} \Big).$$

- **Addition:**

$$\Lambda(c^{(1)} + c^{(2)}) \;\leq\; \Lambda(c^{(1)}) + \Lambda(c^{(2)}).$$

- **Multiplication (pre-relinearization):**

$$\Lambda_{\mathrm{mult}}^{\mathrm{pre}} \;\leq\; C_p \, M_{\mathrm{msg}}^2 + |\Delta_1| M_{\mathrm{msg}} \big(\Lambda(c^{(1)}) + \Lambda(c^{(2)}) \big) + \Lambda(c^{(1)})\, \Lambda(c^{(2)}),$$

  where $C_p = |pk'|_{\mathrm{can}}$ with $\Delta_1^2 = k'p + \Delta_1$.
- **Relinearization:** incurs an additive error bounded (with probability $\geq 1 - \varepsilon$) by (19), added to $\Lambda_{\mathrm{mult}}^{\mathrm{pre}}$ as in (20).

Combined with the global correctness condition (11), these recurrences give a uniform way to choose the modulus chain and noise parameters so that all intermediate ciphertexts remain in the no-wrap regime. In particular, the analysis naturally accounts for both the CRT error and the accumulated $p_1^r p_2$-multiples via the single global error term $E(c)$, ensuring that contributions such as k_{mult} are fully captured in the noise budget.

4.6 Experimental noise budget vs. theoretical bound

Table 1 compares the *experimental* noise budget (mean over trials) with the *theoretical* budget obtained by composing our noise recurrences for the same parameter set as in Section 7 ($n = 8192$ and the matched two-prime modulus ladder Q). Following the presentation style used in prior BFV noise-growth studies, we report all values in *bits*: x denotes the observed mean remaining budget, while E denotes the predicted (worst-case) lower bound from the analysis. Larger values indicate a larger correctness margin; in particular, the analysis is conservative when $x \geq E$.

n	Enc		Add		Mult		$\min(x - E)$
	E	x	E	x	E	x	
8192	49.10	51.42	48.25	51.34	6.88	7.50	0.62

Table 1. Noise budget (bits) for $n = 8192$: Theoretical bounds (E) vs. experimental results (x). Mult includes relinearization.

5 Security Analysis

In this section, we present the security analysis of our scheme. To achieve this, we start by introducing a new hardness assumption: the *Chinese Remainder Theorem Ring Learning With Errors* (CRT-RLWE) problem. The new problem extends the definition of the standard RLWE problem by incorporating structured noise via CRT encoding. This adaptation supports improved batching and efficiency while retaining the hardness of RLWE. For establishing the hardness of the new assumption, we show that CRT-RLWE problem is at least as hard as the RLWE problem.

Definition 5 (CRT-RLWE Distribution). *Let s be a secret in the ring $\mathcal{R}_q$, and let χ be a noise distribution over $\mathcal{R}_q$. A sample from the Chinese Remainder Theorem-based Ring-LWE (CRT-RLWE) Distribution, denoted as $C_{s,\chi}$, is a pair:*

$$(\mathbf{a}, \mathbf{b}), \quad \textit{where} \quad \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e})$$

Here, $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$ is chosen uniformly at random, and $\mathbf{e} \leftarrow \chi$. The CRT-RLWE distribution is formally defined as:

$$C_{s,\chi} = \left\{ (\mathbf{a}, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e})) \mid \mathbf{a} \xleftarrow{\$} \mathcal{R}_q, \quad \mathbf{e} \leftarrow \chi \right\}$$

Definition 6 (Decision CRT-RLWE Problem). *The* Decision CRT-RLWE problem*, denoted* CRT-DLWE$_{q,\chi}$*, is defined as follows:*

For any probabilistic polynomial-time (PPT) distinguisher $\mathcal{D}$, the advantage in solving the decision CRT-RLWE problem is:

$$Adv_{\mathcal{D}}^{\text{CRT-DLWE}_{q,\chi}} := \left| \Pr_{\mathbf{a},\mathbf{b} \leftarrow C_{s,\chi}} [\mathcal{D}(\mathbf{a}, \mathbf{b}) = 1] - \Pr_{\mathbf{a},\mathbf{b} \xleftarrow{\$} \mathcal{R}_q^2} [\mathcal{D}(\mathbf{a}, \mathbf{b}) = 1] \right|$$

Theorem 4. *Let $s \in \mathcal{R}_q$ be a secret and χ a noise distribution over $\mathcal{R}_q$. Then any sample $(\mathbf{a}, \mathbf{b})$ from the standard RLWE distribution $A_{s,\chi}$*

can be efficiently transformed into a sample $(\mathbf{a}_{CRT}, \mathbf{b}_{CRT})$ from the CRT-RLWE distribution $C_{s,\chi}$.

Proof. Given an RLWE sample $(\mathbf{a}, \mathbf{b})$, where $\mathbf{a} \in \mathcal{R}_q, \mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e} \mod q$, we define a CRT embedding $\mathbf{b}_{\text{CRT}} \in \mathcal{R}_q$ as follows:

$$\mathbf{b}_{\text{CRT}} = \Delta_2 \cdot \mathbf{b} \mod q$$

Substituting $\mathbf{b} = \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$, we obtain:

$$\mathbf{b}_{\text{CRT}} = (\Delta_2 \cdot \mathbf{a}) \cdot \mathbf{s} + \Delta_2 \cdot \mathbf{e} = (\Delta_2 \cdot \mathbf{a}) \cdot \mathbf{s} + CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e}).$$

If $\mathbf{e}$ is small enough such that $\|\Delta_2 \cdot \mathbf{e}\| < q/2$, then we can express:

$$\Delta_2 \cdot \mathbf{e} \mod q = CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e}) \mod q.$$

So we can write

$$\mathbf{b}_{\text{CRT}} = (\Delta_2 \cdot \mathbf{a}) \cdot \mathbf{s} + CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e}).$$

Further, we define

$$\mathbf{a}_{\text{CRT}} = \Delta_2 \cdot \mathbf{a}$$

Thus, this transformation correctly simulates the CRT-RLWE problem instance:

$$(\mathbf{a}_{\text{CRT}}, \mathbf{b}_{\text{CRT}}) = (\Delta_2 \cdot \mathbf{a}, \mathbf{b}_{\text{CRT}}).$$

The transformation $\mathbf{b} \to \mathbf{b}_{\text{CRT}}$ preserves the RLWE problem's structure within the CRT-RLWE framework:

$$\mathbf{b}_{\text{CRT}} = \mathbf{a}_{\text{CRT}} \cdot \mathbf{s} + CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e}) \mod q \iff \mathbf{b} - \mathbf{a} \cdot \mathbf{s} = \mathbf{e} \mod q.$$

Specifically, we show the following:

- The CRT embedding maps $\mathbf{b}$ to $\mathbf{b}_{\text{CRT}}$, embedding the RLWE noise $\mathbf{e}$ into the CRT representation $CRT_{p_1,p_2}(\mathbf{0}, \mathbf{e})$, which matches the CRT-RLWE formulation.
- The secret $\mathbf{s}$ remains identical between the RLWE and CRT-RLWE formulations.

Theorem 5 (IND-CPA Security). *The proposed CRT-FHE cryptosystem is IND-CPA secure under the hardness of the Decision CRT-RLWE problem.*

Proof. We prove this theorem using a sequence of games $G_0 \to G_1 \to G_2$ as follows,

Game G_0: Real IND-CPA Game. In this game, the adversary $\mathcal{A}$ receives:

- a challenger generates public key $\mathsf{pk} = (\mathbf{pk}_0, \mathbf{pk}_1) = (\mathbf{a} \cdot \mathbf{s} + \text{CRT}_{p_1,p_2}(0, \mathbf{e}), -\mathbf{a}) \in \mathcal{R}_q^2$, where $\mathbf{a}, \mathbf{s} \xleftarrow{\$} \mathcal{R}_q$, $\mathbf{e} \xleftarrow{\$} \chi$
- adversary $\mathcal{A}$ receives pk and generate two messages $\mathsf{m_0}, \mathsf{m_1}$
- challenger chooses $b \xleftarrow{\$} \{0, 1\}$ and sends $\mathsf{c} = \mathsf{Enc}(\mathsf{pk}, \mathsf{m_b})$ to challenger.
- a challenge ciphertext encrypting a plaintext $\mathbf{m}_b \in R_{p_1}$, where $b \in \{0, 1\}$ is chosen uniformly at random.

The adversary outputs a guess b'.
Define:

$$\text{Adv}_{\mathcal{A}}^{G_0} := \left| \Pr[\mathcal{A} \text{ outputs } b' = b] - \frac{1}{2} \right|$$

Game G_1: Replace Public Key with Uniform. In this game, we sample $(\mathbf{pk}_0, \mathbf{pk}_1) \xleftarrow{\$} \mathcal{R}_q^2$ uniformly at random. The challenge ciphertext is still computed using the same encryption procedure.

We want to prove that Game G_1 is indistinguishable from Game G_0. Suppose there exists any PPT adversary $\mathcal{A}$ which aims to distinguish Game G_0 and Game G_1 with non-negligible advantage. Then we can construct a distinguisher $\mathcal{D}$ aims to break CRT-DRLWE$_{n,q,\chi}$. $\mathcal{D}$ receives a sample $(pk_1, pk_2) \in \mathcal{R}_q^2$, then runs $\mathcal{A}$ with (pk_1, pk_2) as the public key. Here $\mathcal{D}$ can perfectly simulate Game G_0 or Game G_1 depending on whether its input samples are CRT-RLWE or uniform (respectively). Therefore,$\mathcal{D}$ and

$\mathcal{A}$ have equal distinguishing advantages. Because $\mathcal{D}$'s advantage must be negligible by CRT-RLWE assumption (Definition 6), so is $\mathcal{A}$'s.

In other words, we can express the advantage of $\mathcal{A}$ winning the Game G_0 and Game G_1 using the advantage of $\mathcal{D}$ which is negligible in the security parameter λ.

$$\mathrm{Adv}_{\mathcal{D}}^{\text{CRT-RLWE}_{n,q,\chi}} = |\Pr[\mathcal{A} \text{ wins in } G_0] - \Pr[\mathcal{A} \text{ wins in } G_1]|$$

By assumption (hardness of Decision CRT-RLWE), for all PPT $\mathcal{D}$,

$$\mathrm{Adv}_{\mathcal{D}_1}^{\text{CRT-RLWE}} \leq \epsilon_1(\lambda)$$

Thus:

$$|\Pr[\mathcal{A} \text{ wins in } G_0] - \Pr[\mathcal{A} \text{ wins in } G_1]| \leq \epsilon_{CRT-RLWE}(\lambda)$$

Game G_2: Replace Ciphertext with Uniform. In this game, we again use a public key sampled uniformly from $\mathcal{R}_q^2$, as in G_1. We now also replace the ciphertext with uniformly random elements $(\mathbf{c}_0, \mathbf{c}_1) \xleftarrow{\$} \mathcal{R}_q^2$, independently of the public key.

We want to prove that Game G_1 and Game G_2 are indistinguishable from $\mathcal{A}$. The ciphertext generated by a public key which is chosen uniformly random from $\mathcal{R}_q^2$ has a distribution like the following:

$$\mathcal{E}_{s,\chi}\{(\mathbf{a}, \mathbf{a}\cdot\mathbf{s} + \mathrm{CRT}_{p_1,p_2}(\mathbf{m}, \mathbf{e})) = ((\mathbf{a}, \mathbf{a}\cdot\mathbf{s} + \mathrm{CRT}_{p_1,p_2}(\mathbf{0}, \mathbf{e}) + \Delta m)\},$$

where $|\ \mathbf{a} \xleftarrow{\$} \mathcal{R}_q, \mathbf{e} \xleftarrow{\$} \chi, \mathbf{m} \xleftarrow{\$} R_{p_1}, \Delta = p_2 \cdot [p_2^{-1}]_{p_1}$

Then the $\mathcal{E}_{s,\chi}$ is indistinguishable from uniform distribution since all the elements are CRT-RLWE samples adding any fixed message $\Delta m \in \mathcal{R}_q$, which preserves the uniformity. Moreover, Game G_1 and Game G_2 are indistinguishable for any fixed m and Game G_2 does not depend on m at all, therefore, the two games for $m = 0, 1$ are indistinguishable.

In other words, we can express the advantage of $\mathcal{A}$ winning the Game G_1 and Game G_2 using the advantage of $\mathcal{D}_{\in}$ a PPT adversary that distinguishes encryptions under $\mathcal{E}_{s,\chi}$ (defined below) from uniform.

The advantage of $\mathcal{D}_2$ is the same as $\mathcal{D}_1$ which is CRT-RLWE distinguisher.

$$|\Pr[\mathcal{A} \text{ wins in } G_1] - \Pr[\mathcal{A} \text{ wins in } G_2]| = \mathrm{Adv}_{\mathcal{D}_2}^{\mathcal{E}_{s,\chi}} = \mathrm{Adv}_{\mathcal{D}_1}^{\text{CRT-RLWE}} \leq \epsilon_{CRT-RLWE}(\lambda)$$

Game G_2 Analysis. In Game G_2, the ciphertext is sampled uniformly and independently of the plaintext bit b. Therefore, regardless of $\mathcal{A}$'s strategy,

$$\Pr[\mathcal{A} \text{ wins in } G_2] = \frac{1}{2}$$

Final Bound. Combining all bounds via triangle inequality:

$$\begin{aligned}\left|\Pr[\mathcal{A} \text{ wins in } G_0] - \frac{1}{2}\right| &= |\Pr[\mathcal{A} \text{ wins in } G_0] - \Pr[\mathcal{A} \text{ wins in } G_2]| \\ &\leq |\Pr[\mathcal{A} \text{ wins in } G_0] - \Pr[\mathcal{A} \text{ wins in } G_1]| \\ &\quad + |\Pr[\mathcal{A} \text{ wins in } G_1] - \Pr[\mathcal{A} \text{ wins in } G_2]| \\ &\leq \epsilon_{CRT-RLWE}(\lambda) + \epsilon_{CRT-RLWE}(\lambda)\end{aligned}$$

Hence, the advantage of any PPT adversary $\mathcal{A}$ in the IND-CPA game is negligible under the Decision CRT-RLWE assumption, and the CRT-FHE cryptosystem is IND-CPA secure.

6 Bootstrapping CRT-FHE

Goal and invariant. Given a valid CRT-FHE ciphertext $\mathrm{ct} = (c_0, c_1) \in \mathcal{R}_q^2$ under secret key $s \in \mathcal{R}_q$, bootstrapping should output a refreshed ciphertext $\mathrm{ct}^\star$ such that

$$\langle c_0 + c_1 s \rangle_q = p_1^r p_2 \cdot k + \mathrm{CRT}_{p_1^r, p_2}(m, e) \;\rightarrow\; \langle c_0^\star + c_1^\star s \rangle_q = p_1^r p_2 \cdot k^\star + \mathrm{CRT}_{p_1^r, p_2}(m, e_{\mathrm{fresh}})$$

for the *same* message $m \in \mathcal{R}_{p_1^r}$, but with "fresh" noise e_{fresh} distributed as in a new encryption. That is, bootstrapping preserves the CRT structure and the plaintext while resetting the noise.

Prime-power plaintext and decryption form. To support digit extraction, we work with a prime-power message modulus p_1^r (with $r \geq 1$) and an auxiliary CRT modulus p_2 coprime to p_1. The native plaintext ring is $\mathcal{R}_{p_1^r}$; the scheme of Section 3 corresponds to the special case $r = 1$.

Let $\mathrm{ct} = (c_0, c_1)$ be as above and define the coefficient-wise centered decryption value

$$w \;:=\; \langle c_0 + c_1 s \rangle_q \in \mathcal{R},$$

where $\langle \cdot \rangle_q$ denotes representatives in $(-q/2, q/2]$. In the no-wrap regime,

$$w \;=\; p \cdot k \;+\; \mathrm{CRT}_{p_1^r, p_2}(m, e), \qquad p = p_1^r p_2, \tag{21}$$

for some $k \in \mathcal{R}$, $m \in \mathcal{R}_{p_1^r}$, and $e \in \mathcal{R}_{p_2}$. Writing

$$\Delta_1 := p_2 [p_2^{-1}]_{p_1^r}, \qquad \Delta_2 := p_1^r [(p_1^r)^{-1}]_{p_2},$$

we have $\mathrm{CRT}_{p_1^r, p_2}(m, e) \equiv \Delta_1 m + \Delta_2 e \pmod{p}$, with $\Delta_1 \equiv 1 \pmod{p_1^r}$ and $\Delta_2 \equiv 0 \pmod{p_1^r}$. Thus decryption is

$$m \;=\; w \bmod p_1^r.$$

Bootstrapping-oriented (lifted) decryption. Following the RLWE-based recryption framework of Halevi–Shoup [21, §3.1], we rewrite decryption in a lifted modulus p_1^e with $e > r$.

We consider the scale-and-round map

$$c_i' \leftarrow [p_1^{e-r} c_i]_q, \qquad w' \leftarrow [c_0' + c_1' s]_{p_1^e}, \qquad m \leftarrow \left[\!\left[w' / p_1^{e-r} \right]\!\right]_{p_1^r}. \tag{22}$$

Here $[\cdot]_{p_1^e}$ is reduction modulo p_1^e and $\left[\!\left[\cdot \right]\!\right]_{p_1^r}$ denotes coefficient-wise rounding followed by reduction modulo p_1^r.

Substituting (21) into the scaled expression yields

$$c_0' + c_1' s \;=\; p_1^{e-r} w \;=\; p_1^{e-r} \big(p_1^r p_2 k + \Delta_1 m + \Delta_2 e \big).$$

Modulo p_1^e, the terms $p_1^{e-r} \cdot p_1^r p_2 k$ and $p_1^{e-r} \Delta_2 e$ vanish (being multiples of p_1^e), while $\Delta_1 \equiv 1 \pmod{p_1^r}$ implies

$$w' \equiv p_1^{e-r} m \pmod{p_1^e}.$$

Hence $\left[\!\left[w' / p_1^{e-r} \right]\!\right]_{p_1^r} = m$ in the no-wrap regime. Bootstrapping homomorphically evaluates (22) (via base-p_1 digit extraction) to obtain an encryption of m, and then restores the native CRT ciphertext structure by CRT re-embedding.

High-level pipeline. Given $\mathrm{ct} = (c_0, c_1)$, bootstrapping proceeds in four phases:

1. **Lifted homomorphic decryption.** Using recryption material as in [21, §3.1], homomorphically evaluate the lifted decryption form (22) to obtain a ciphertext encrypting w' modulo p_1^e.
2. **Coefficient isolation.** Apply the linear evaluation map and, if needed, unpacking maps to move the relevant coefficients/blocks of w' into independent slots, as in [21, §4.2–4.3].
3. **Digit extraction.** Apply base-p_1 digit extraction (Halevi–Shoup [21, §5.3, Fig. 1] or Chen–Han [8, Alg. 1, Fig. 2]) to implement the division-and-rounding step in (22), recovering m modulo p_1^r in each slot.
4. **CRT restoration.** Repack and invert the linear transforms to obtain a ciphertext encrypting m in coefficient form, multiply by Δ_1 to obtain $\mathrm{CRT}_{p_1^r,p_2}(m, 0)$, and rerandomize by adding a fresh encryption of 0, thereby restoring the full CRT ciphertext shape.

After digit extraction, we have an encryption of m modulo p_1^r. Multiplying by Δ_1 embeds m into the CRT structure as $\mathrm{CRT}_{p_1^r,p_2}(m, 0)$ Rerandomization is required because the digit extraction process may leak information about the noise; adding a fresh encryption of 0 ensures the output distribution is indistinguishable from a fresh encryption.

The output is a ciphertext $\mathrm{ct}^\star$ encrypting the same m modulo p_1^r with refreshed noise and the same CRT structure as a fresh CRT-FHE encryption.

6.1 Digit extraction for CRT-FHE

Digit extraction is the nonlinear core of recryption: it implements the map $w' \mapsto [\![w'/p_1^{e-r}]\!]_{p_1^r}$ via homomorphic polynomial evaluation. As in prior RLWE bootstrapping schemes, digit extraction is applied coefficient-wise and is therefore preceded by the linear transform and unpacking layers that move the relevant coefficients into independent slots [21, §4.2–4.3] (cf. the pipeline in [8, Fig. 2]).

Balanced base-p_1 digits and functionality. Fix a prime p_1 and an exponent $e \geq 1$. For any $w \in \mathbb{Z}_{p_1^e}$, write its balanced base-p_1 expansion

$$w \;=\; \sum_{i=0}^{e-1} w_i p_1^i, \qquad w_i \in \left\{ -\frac{p_1-1}{2}, \ldots, \frac{p_1-1}{2} \right\}.$$

For $v \in \{0, \ldots, e\}$, define the "remove-v-lowest-digits" map

$$\mathrm{DigitExtract}_{p_1,e,v}(w) \;:=\; \left\lfloor \frac{w}{p_1^v} \right\rceil \;=\; \sum_{i=v}^{e-1} w_i \, p_1^{i-v} \in \mathbb{Z}_{p_1^{e-v}}, \tag{23}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest balanced integer. In particular, for $v = e - r$ this implements the divide-and-round step of (22).

Instantiation in CRT-FHE. With native plaintext modulus p_1^r and lifting parameter $e > r$, the lifted value

$$w' \;=\; [c_0' + c_1' s]_{p_1^e} \in \mathcal{R}_{p_1^e}$$

satisfies $w' \equiv p_1^{e-r} m \pmod{p_1^e}$ in the no-wrap regime. Thus, coefficient-wise,

$$m \;=\; \left[\!\left[w'/p_1^{e-r} \right]\!\right]_{p_1^r} \;=\; \mathrm{DigitExtract}_{p_1,e,e-r}(w') \bmod p_1^r.$$

In our CRT encoding, the (unscaled) centered decryption expression has the form $w = p_1^r p_2 \cdot k + \mathrm{CRT}_{p_1^r,p_2}(m, e_2)$, so reduction modulo p_1^r eliminates the multiple of p_1^r and returns the message component; digit extraction implements this reduction homomorphically.

Concrete procedures. We instantiate $\mathrm{DigitExtract}_{p_1,e,v}$ via existing polynomial-evaluation circuits. Halevi–Shoup [21, §5.3, Fig. 1] give a procedure based on *lifting polynomials* that iteratively removes upper digits while maintaining correctness modulo increasing powers of p_1. Chen–Han [8, Alg. 1] give an improved procedure ("lowest-digits removal") that applies a low-degree lowest-digit retain polynomial to reduce multiplicative depth; the placement of digit extraction within the bootstrapping pipeline is summarized in [8, Fig. 2]. In our setting, these procedures are applied after the linear transform/unpacking stage [21, §4.2–4.3], so that digit extraction is evaluated coefficient-wise on independent slots.

6.2 Complete bootstrapping procedure for CRT-FHE

We now summarize the full bootstrapping procedure in Algorithm 1, combining the lifted decryption map (22), the digit-extraction primitive of Section 6.1, and the CRT re-embedding described above.

Theorem 6 (Correctness of CRT-FHE.Bootstrap). *Let* ct *be a valid CRT-FHE ciphertext encrypting $m \in \mathcal{R}_{p_1^r}$ under secret key s, and suppose its centered decryption value satisfies*

$$w = \langle c_0 + c_1 s \rangle_q = p_1^r p_2 \cdot k + \mathrm{CRT}_{p_1^r,p_2}(m, e)$$

with coefficients in the no-wrap range. Then Algorithm 1 outputs a ciphertext $\mathrm{ct}^\star$ *such that*

$$\mathrm{Dec}(\mathrm{ct}^\star) \equiv m \pmod{p_1^r},$$

and its centered decryption value has the form

$$\langle c_0^\star + c_1^\star s \rangle_q = p_1^r p_2 \cdot k^\star + \mathrm{CRT}_{p_1^r,p_2}(m, e_{\mathrm{fresh}})$$

for some $k^\star \in \mathcal{R}$ and some e_{fresh} distributed as the noise component of a fresh CRT-FHE encryption.

Proof. By correctness of the lifted decryption step in [21, §3.1], Step 1 yields a ciphertext ct_w whose decryption w' satisfies

$$w' \equiv p_1^{e-r} m \pmod{p_1^e}$$

in each coefficient, under the no-wrap assumption. The linear transforms and (un)packing in Steps 2–3 and their inverses in Steps 5–6 are homomorphic realizations of invertible linear maps [21, §4.2–4.3], so they preserve the plaintext values in the slots.

Algorithm 1 CRT-FHE.Bootstrap

Input: Ciphertext $\mathrm{ct} = (c_0, c_1) \in \mathcal{R}_q^2$ encrypting $m \in \mathcal{R}_{p_1^r}$; recryption material as in [21, §3.1]; public key pk

Output: Refreshed ciphertext $\mathrm{ct}^\star \in \mathcal{R}_q^2$ encrypting the same m

1: **Lifted homomorphic decryption.** Modulus-switch ct from q to a special modulus $\hat{q}$ and homomorphically evaluate the lifted linear form:

$$\mathrm{ct}_w \leftarrow \mathrm{Enc}_{\hat{q}}\big([c_0' + c_1' s]_{p_1^e}\big),$$

where $c_i' = [p_1^{e-r} c_i]_{\hat{q}}$, following [21, §3.1].

2: **Transforms to slots.** $\mathrm{ct}_{\mathsf{slots}} \leftarrow \mathrm{CoeffToSlot}(\mathrm{ct}_w)$ [21, §4.2]

3: **(Un)packing.** $(\mathrm{ct}^{(0)}, \ldots, \mathrm{ct}^{(d-1)}) \leftarrow \mathrm{Unpack}(\mathrm{ct}_{\mathsf{slots}})$ [21, §4.3]

4: **for** $i = 0$ **to** $d - 1$ **do**

5: **Digit extraction.** $\mathrm{ct}_m^{(i)} \leftarrow \mathrm{DigitExtract}_{p_1,e,e-r}\big(\mathrm{ct}^{(i)}\big)$ [21, §5.3], [8, Alg. 1]

6: **end for**

7: **Inverse transforms.** $\mathrm{ct}'_{\mathsf{slots}} \leftarrow \mathrm{Repack}(\mathrm{ct}_m^{(0)}, \ldots, \mathrm{ct}_m^{(d-1)})$ [21, §4.3]

8: $\mathrm{ct}_m \leftarrow \mathrm{SlotToCoeff}(\mathrm{ct}'_{\mathsf{slots}})$ [21, §4.2]

9: **CRT re-embedding.** $\mathrm{ct}_{\mathrm{proj}} \leftarrow \Delta_1 \cdot \mathrm{ct}_m$

10: **Rerandomization.** $\mathrm{ct}^\star \leftarrow \mathrm{ct}_{\mathrm{proj}} + \mathrm{CRT}\text{-}\mathsf{FHE.Enc}(pk, 0)$

11: **return** $\mathrm{ct}^\star$ (optionally modulus-switch/key-switch back as in [21, §3.1])

By correctness of digit extraction in [21, §5.3] (Halevi–Shoup) or [8, Alg. 1, Thm. 5] (Chen–Han), Step 4 computes, in each slot,

$$w_m = \mathrm{DigitExtract}_{p_1,e,e-r}(w') = \left\lfloor \frac{w'}{p_1^{e-r}} \right\rceil,$$

which equals m modulo p_1^r. Hence the ciphertext ct_m after inverse transforms decrypts to m modulo p_1^r.

In Step 7, multiplication by Δ_1 is a public constant multiplication in $\mathcal{R}$. If $z \equiv m \pmod{p_1^r}$ then

$$\Delta_1 z \equiv \mathrm{CRT}_{p_1^r,p_2}(m, 0) \pmod{p_1^r p_2},$$

so $\mathrm{ct}_{\mathrm{proj}}$ decrypts to $\mathrm{CRT}_{p_1^r,p_2}(m, 0)$ plus an integral multiple of $p_1^r p_2$.

In Step 8, $\mathrm{CRT}\text{-}\mathsf{FHE.Enc}(pk, 0)$ is distributed as a fresh encryption of 0, whose centered decryption value has the form

$$p_1^r p_2 \cdot k_0 + \mathrm{CRT}_{p_1^r,p_2}(0, e_{\mathrm{fresh}})$$

for some k_0 and noise e_{fresh}. Adding this to $\mathrm{ct}_{\mathrm{proj}}$ yields a ciphertext $\mathrm{ct}^\star$ whose centered decryption value is

$$p_1^r p_2 \cdot k^\star + \mathrm{CRT}_{p_1^r,p_2}(m, e_{\mathrm{fresh}})$$

for some $k^\star$, and in particular reduces to m modulo p_1^r. All operations preserve the no-wrap condition by parameter choice, so the centered representatives are well defined throughout. This proves the claim.

6.3 Comparison with CRT-based FHE over the integers

Cheon–Kim–Lee–Yun (CKLY) revisit the third privacy–homomorphism of Rivest–Adleman–Dertouzos and obtain a fully homomorphic scheme over the integers whose

security relies on the decisional approximate-GCD problem together with a sparse subset-sum assumption. [12] Their construction can be viewed as a CRT generalization of the DGHV scheme, and they show that the ciphertext expansion overhead improves from $\tilde{O}(\lambda^8)$ for DGHV to $\tilde{O}(\lambda^5)$ for their CRT variant, while preserving the same $\tilde{O}(\lambda^5)$ ciphertext-size order as DGHV with compressed public key. [12, 14, Sec. 1] Moreover, CKLY support large plaintext spaces $\mathbb{Z}_Q$ with $\log Q = O(\lambda^4)$ and SIMD-style plaintexts $\mathbb{Z}_Q^k$ with $\log Q = O(\lambda)$ and $k = O(\lambda^3)$. [12, Abstract]

Our scheme shares with CKLY the goal of incorporating a CRT structure into FHE, but departs in two fundamental ways:

- **Security foundation.** CKLY work in the integer setting and base security on decisional approximate-GCD plus a sparse subset-sum assumption, for which no worst-case hardness reductions are known. [12] In contrast, our CRT-FHE scheme is built on the Ring-LWE problem via the CRT-RLWE reduction, and hence enjoys a (quantum) reduction from worst-case ideal-lattice problems in the sense of Brakerski–Vaikuntanathan. [6]
- **Key and ciphertext size.** Integer-based schemes have public keys of size $\tilde{O}(\lambda^{10})$ in the original DGHV construction, [30] reduced to $\tilde{O}(\lambda^7)$ and then to $\tilde{O}(\lambda^5)$ (about 10.1 MB instead of 802 MB in a reported implementation). [13, 14] CKLY inherit the $\tilde{O}(\lambda^5)$ ciphertext-size order. [12] Our scheme, like standard RLWE-based FHE, uses public keys consisting of two ring elements $(a, b = as + 2e)$ in $\mathcal{R}_q$. [6] For typical 128-bit-secure parameters such as $n = 8192$, $\log_2 q \approx 218$ bits or $n = 16384$, $\log_2 q \approx 438$ bits, [26, 27] the public key has size

$$2n \log_2 q \approx 0.4 \text{ MB and } \approx 1.7 \text{ MB},$$

  respectively, one order of magnitude smaller than the compressed DGHV/CKLY public keys at comparable security levels.

In addition, our bootstrapping design reuses the RLWE-based BGV/BFV bootstrapping framework (Halevi–Shoup and Chen–Han), [8, 21] applied to the p_1-message component of the CRT encoding and followed by a simple CRT re-embedding. Thus, the asymptotic depth and gate complexity of bootstrapping match those of state-of-the-art RLWE schemes, while the CRT layer provides an integer-like decomposition of plaintext and noise without incurring the large integer public keys characteristic of AGCD-based FHE.

In summary, relative to the earlier CRT-based integer FHE of CKLY, our construction (i) replaces AGCD-type assumptions by a reduction to Ring-LWE, (ii) reduces public-key and ciphertext sizes to the RLWE regime (hundreds of kilobytes to a few megabytes rather than tens or hundreds of megabytes), and (iii) reuses the mature RLWE bootstrapping toolkit while retaining a CRT structure on the plaintext space.

7 Performance Analysis

7.1 Experimental Setup

All experiments were run on a MacBook Pro (2022) with an Apple M2 SoC (4 performance + 4 efficiency cores), 24 GB RAM, macOS Sequoia 15.2. The CRT-FHE codebase was compiled with `rustc 1.85.1`; OpenFHE (BFV/CKKS) was compiled with `clang 17.0.0`. We used Criterion.rs for CRT-FHE and Google Benchmark for

OpenFHE, both configured for single-threaded runs. Each benchmark performed one warm-up followed by 50 sampling iterations; we report point estimates together with 95% confidence intervals as produced by the respective harnesses. Unless otherwise noted, all measurements are wall-clock latencies; throughput is computed as *slots per millisecond* (slots/ms).

Implementation notes. All schemes use a residual number system (RNS) representation for large moduli. In our CRT-FHE prototype, the hottest modular-arithmetic paths are division-free: we use Barrett-style reduction specialized to the Apple M2 to avoid 128-bit divisions. The NTT kernels use precomputed twiddle tables, lazy modular reduction, and cache-aware butterfly scheduling to reduce L1/L2 pressure. We additionally tune core-local instruction-level parallelism via loop unrolling for M2's out-of-order cores (without relying on explicit SIMD). Concretely, our CRT-FHE modulus is represented using two ≈ 45-bit RNS residues together with an ≈ 60-bit special prime used for preconditioning, which reduces carry propagation and mixed-radix overhead in base-conversion steps.

Fair-comparison notes. Different harnesses (Criterion vs. Google Benchmark) can yield slightly different variance models; we therefore focus on (i) medians and (ii) relative effects that are large compared to reported CIs. We also report per-slot throughput to normalize across schemes with different effective slot counts or polynomial degrees.

Security Level of the Benchmark Parameters. We begin by recalling that the security of the CRT-FHE reduces to the RLWE problem (see Section 5). Consequently, any parameter set (n, Q, χ) that achieves λ-bit security for CRT-FHE also provides an equivalent level of security for RLWE-based schemes such as BFV/BGV.

Therefore in our benchmarks, we evaluate CRT-FHE against BFV utilizing identical underlying ring and modulus parameters. Both implementations employ a polynomial degree of $n = 8192$ (yielding equivalent lattice dimensions and SIMD slot capacities), an identical coefficient-modulus ladder Q, and a plaintext modulus of $t = 65537$. The modulus Q evaluated herein comprises two RNS primes of approximately 45 bits, resulting in an aggregate bit-length of $\log_2 Q \approx 90$ bits. This configuration falls well within the established 128-bit RLWE security guidelines for $n = 8192$, which permit a total coefficient-modulus bit-length of up to approximately 218 bits as described in [1, 15].

7.2 Cryptographic Parameters and results

For CRT-FHE and BFV we used polynomial degree n=8192 (yielding 8192 SIMD slots) and matched ciphertext modulus ladders

$$Q = \{35{,}175{,}245{,}135{,}873,\ 35{,}156{,}991{,}246{,}337\}$$

The plaintext modulus was p_1=65537; the CRT embedding used an auxiliary parameter p_2=3. For CKKS in OpenFHE we followed a common configuration with n=16384 (effective 8192 complex slots) and

$$Q = \{1, 152, 921, 504, 606, 748, 673,\ 281, 474, 977, 595, 393\}.$$

We report per-slot throughput to partially normalize across these degree choices; a full security/precision alignment (e.g., equal BKZ cost or CKKS scale/precision targets) is left to future work.

Table 2. Operation benchmarks with latency (ms), throughput (slots/ms).

Operation	Latency (ms) ↓			Throughput (slots/ms) ↑		
	CRT-FHE	**BFV**	**CKKS**	**CRT-FHE**	**BFV**	**CKKS**
Encrypt	**1.912**	1.984	2.979	**4,283**	4,128	2,750
Add	**0.0130**	0.0249	0.0545	**630,154**	328,915	150,312
Add (in-place)	**0.0070**	0.0186	0.0400	**1,170,286**	440,430	204,800
Multiply + Relin	2.616	**2.193**	2.211	3,130	**3,733**	3,705

Relative effects (point estimates).

- **Encryption.** CRT-FHE is ∼3.6% faster than BFV (1.912 vs. 1.984 ms; 1.98/1.91 ≈ 1.04×) and ∼1.56× faster than CKKS (2.979 ms).
- **Addition.** CRT-FHE outperforms BFV by ∼1.9× (0.0249/0.0130) and CKKS by ∼4.2× (0.0545/0.0130).
- **In-place addition.** CRT-FHE outperforms BFV by ∼2.66× (0.0186/0.0070) and CKKS by ∼5.7× (0.0400/0.0070).
- **Multiplication + relinearization.** CRT-FHE trails BFV by ∼19% (2.616 vs. 2.193 ms) and CKKS by ∼18% (2.616 vs. 2.211 ms).

Throughput (slots/ms) shows the same qualitative picture and partially compensates for CKKS's larger polynomial degree.

7.3 Interpretation and Discussion on Performance Analysis

Additive path. CRT-FHE achieves large gains on additions (both standard and in-place). These gains are consistent with (i) a leaner ciphertext layout and (ii) optimized modular add/sub primitives that avoid wide divisions in the inner loop. Per-slot throughput indicates the advantage persists even when normalizing for slot count.
Multiplicative path. On Multiply+Relin, OpenFHE's BFV/CKKS are faster on this hardware/parameter point. Two likely drivers are (a) highly tuned key-switch/relinearization kernels (including gadget decomposition and RNS base conversions) and (b) CKKS/BFV's mature NTT pipelines. Our CRT-FHE implementation has room to close this gap with deeper tuning of key-switch primitives and memory layout.
Encryption. The small advantage over BFV and the larger advantage over CKKS align with the degree difference (CKKS at n=16384) and with CRT-FHE's lightweight public-key structure. The per-slot normalization corroborates this trend.

7.4 Ciphertext Size and Computational Complexity

This section compares CRT-FHE with BFV and BGV under a fixed λ-bit security setting. As discussed earlier, any parameter set (n, Q, χ) that achieves λ-bit security for CRT-FHE provides an equivalent level of security for RLWE-based schemes such as BFV and BGV. Accordingly, for a given security level (e.g., 128 bits), parameters are selected following the guidelines of the HomomorphicEncryption.org standard [1, 15].

Since all three schemes use the same RLWE ciphertext modulo $(c_0, c_1) \in R_Q^2$, the leading-order ciphertext size and the dominant ring-arithmetic cost for Add and Mult coincide; differences arise from scheme semantics (e.g., plaintext encoding/scaling, and in BGV optional modulus switching), not from the ciphertext representation.

Scheme	Ciphertext form	Size (bytes)
CRT-FHE	$(c_0, c_1) \in R_Q^2$	$\approx \frac{2n \log_2 Q}{8}$
BFV	$(c_0, c_1) \in R_Q^2$	$\approx \frac{2n \log_2 Q}{8}$
BGV	$(c_0, c_1) \in R_Q^2$	$\approx \frac{2n \log_2 Q}{8}$
Example: (n=8192, $\log_2 Q$=218)		$\approx 4.46 \times 10^5$ B ($\approx$ 436 KiB)

Table 3. Ciphertext size comparison at a fixed security level.

Ciphertext size A ciphertext consists of two polynomials in R_Q, hence its leading-order size is

$$|c|_{\text{bytes}} \approx \frac{2\, n\, \log_2 Q}{8},$$

ignoring small serialization headers and optional compression. After a homomorphic multiplication, ciphertexts are relinearized back to two components, so the ciphertext size remains the same. See Table 3

Time Complexity of Homomorphic Operations We report the dominant arithmetic cost in terms of the number of polynomial additions and polynomial multiplications in R_Q. Each polynomial addition costs $O(n)$ coefficient operations, while each polynomial multiplication costs $\tilde{O}(n)$ using NTT-based multiplication, depending on the implementation.

Addition. For ciphertexts $\mathbf{c}_1 = (c_0^1, c_1^1)$ and $\mathbf{c}_2 = (c_0^2, c_1^2)$, homomorphic addition is component-wise:

$$\mathbf{c}_{\text{add}} = (c_0^1 + c_0^2 \bmod Q,\ c_1^1 + c_1^2 \bmod Q).$$

thus requiring exactly 2 polynomial additions.

Multiplication (with relinearization). Ciphertext multiplication first forms a degree-2 ciphertext and then relinearizes it back to two components. The degree-2 product costs 3 polynomial multiplications and 1 polynomial addition:

$$\mathbf{c}_{\text{mult}} = (c_0, c_1, c_2) \in R_Q^3, \text{ where}$$

$$c_0 = c_0^1 \cdot c_0^2, \quad c_1 = c_0^1 \cdot c_1^2 + c_1^1 \cdot c_0^2, \quad c_2 = c_1^1 \cdot c_1^2 \tag{24}$$

Relinearization uses a gadget base $w \geq 2$ with $\ell = \lceil \log_w Q \rceil$ digits; it contributes 2ℓ polynomial multiplications and 2ℓ polynomial additions (plus coefficient decomposition, typically $O(n\ell)$ digit operations). Therefore,

$$\textsf{Mult+Relin}: \quad (3 + 2\ell) \text{ poly-muls} \quad \text{and} \quad (1 + 2\ell) \text{ poly-adds}.$$

This leading-order cost is the same for CRT-FHE, BFV, and BGV (BGV may additionally apply modulus switching after multiplication, implemented as coefficient-wise scale-and-round operations).

Tables 3–4 isolate the *common* arithmetic backbone of RLWE-style FHE schemes at a fixed security level. In particular, CRT-FHE does not incur ciphertext expansion beyond the standard RLWE multiplication-and-relinearization pattern; the CRT layer affects plaintext embedding and bootstrapping restoration, but not the ciphertext container size or the leading-order count of polynomial multiplications/additions for basic Add and Mult. Concrete latencies depend on implementation choices (RNS layout, NTT scheduling, memory traffic, and key-switching costs), which we report separately in Section 7.

Operation	CRT-FHE	BFV	BGV
Add	2 poly-add	2 poly-add	2 poly-add
Mult+Relin	$(3+2\ell)$ poly-mul, $(1+2\ell)$ poly-add	same	same

Table 4. Time Complexity of Homomorphic Operations.

8 Conclusion

We propose a fully homomorphic encryption (FHE) scheme based on the CRT-RLWE problem. We show that the hardness of CRT-FHE is at least that of the Ring-LWE problem. The proposed scheme CRT-FHE preserves the design advantages that originally motivated CRT in homomorphic encryption, namely modular decomposition and a packing-friendly structure, while retaining compact RLWE-style key material and efficient polynomial-ring arithmetic. Finally, to achieve *fully* homomorphic evaluation, we presented a bootstrapping procedure specialized to the CRT ciphertext form: the procedure homomorphically evaluates a lifted decryption map using established recryption components and then restores the native CRT embedding via a CRT re-embedding and rerandomization step. Experimental results show that CRT-FHE outperforms both CKKS and BFV in encryption and addition latency.

Several directions remain open and practically relevant. First, while we establish CRT-RLWE hardness via reduction to RLWE, tightening the quantitative loss in the reduction and translating it into sharper parameter guidance, including conservative post-quantum margins is an important next step. Second, improving the concrete efficiency of bootstrapping remains central: the present design follows the canonical structure (linear transforms, digit extraction, inverse transforms), and future work should focus on reducing multiplicative depth and runtime via lower-depth digit-extraction instantiations, better transform scheduling, and implementation-level optimizations.

References

1. M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan. Homomorphic encryption standard. Cryptology ePrint Archive, Paper 2019/939, 2019.
2. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. *Advances in Cryptology – CRYPTO 2012*, 7417:868–886, 2012.
3. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*, pages 309–325, 2012.
4. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. *Advances in Cryptology – CRYPTO 2011*, 6841:505–524, 2011.
5. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. *P. Rogaway, editor, Advances in Cryptology – CRYPTO 2011*, 6841:505–524, 2011.
6. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In P. Rogaway, editor, *Advances in*

Cryptology – CRYPTO 2011, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.
7. E. Brickell and Y. Yacobi. On privacy homomorphisms (extended abstract). *D. Chaum and W. Price, editors, Advances in Cryptology EUROCRYPT 87*, 304:117–125, 1988.
8. H. Chen and K. Han. Homomorphic Lower Digits Removal and Improved FHE Bootstrapping. *Advances in Cryptology – EUROCRYPT 2018, LNCS 10820*, 10820:315–337, 2018.
9. Y. Chen and P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. Cryptology ePrint Archive, Paper 2011/436, 2011.
10. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer, 2013.
11. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, 2017.
12. J. H. Cheon, J. Kim, M. S. Lee, and A. Yun. Crt-based fully homomorphic encryption over the integers. *Information Sciences*, 310:149–162, 2015.
13. J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2011.
14. J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer, 2012.
15. Cryptography and Privacy Research Group, Microsoft. Microsoft SEAL (release 4.1). `https://github.com/microsoft/SEAL`, 2024. GitHub repository.
16. D. Das and A. Joux. On the hardness of the finite field isomorphism problem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 343–359. Springer, 2023.
17. J. Fan and F. Vercauteren. Somewhat Practical Fully Homomorphic Encryption. In *IACR Cryptology ePrint Archive, Report 2012/144*, 2012.
18. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, 05 2009.
19. C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 116–137, 08 2010.
20. C. Gentry, S. Halevi, and N. Smart. Fully homomorphic encryption with polylog overhead. *Advances in Cryptology – EUROCRYPT 2012*, 7237:465–482, 2012.
21. S. Halevi and V. Shoup. Bootstrapping for HElib. *Advances in Cryptology – EUROCRYPT 2015, LNCS 9056*, 9056:641–670, 2015.
22. Z. Li, Y. Liu, X. Lu, R. Wang, C. Wei, B.and Chen, and K. Wang. Faster bootstrapping via modulus raising and composite ntt. In *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 563–591, 2023.
23. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-lwe cryptography. *Lecture Notes in Computer Science (LNCS)*, 7881:35–54, 2013.

24. D. Micciancio and O. Regev. Lattice-based Cryptography. *Post-Quantum Cryptography*, PQCrypto:147–191, 2009.
25. Microsoft SEAL contributors. Issues with defining coeffmodulus for ckks (github issue #128). GitHub issue tracker, 2020. Includes discussion of 128-bit security coefficient-modulus bit-count bounds for polynomial degrees N=8192 and N=16384. Opened Jan 28, 2020.
26. Microsoft SEAL Team. Microsoft SEAL homomorphic encryption library: Parameter selection (example with $n = 16384$). `https://github.com/microsoft/SEAL`. Example parameter set with polynomial modulus degree $N = 16384$ and total coefficient-modulus bit-length about 438 bits at 128-bit security.
27. Microsoft SEAL Team. Microsoft SEAL homomorphic encryption library: Parameter selection (example with $n = 8192$). `https://github.com/microsoft/SEAL`. Example parameter set with polynomial modulus degree $N = 8192$ and total coefficient-modulus bit-length about 218 bits at 128-bit security.
28. R. L. Rivest and M. L. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–180. Academic Press, 1978.
29. M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. *Advances in Cryptology – EUROCRYPT 2010*, 6110:24–43, 2010.
30. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *International Conference on the Theory and Application of Cryptographic Techniques*, 2010.