Butterfly Effect: Multi-Key FHE from Ring-LWR

Mansi Goyal[1], Ali Raya[2*], Mohakjot Dhiman[3],
Aditi Kar Gangopadhyay[1*]

[1*]Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, 247667, Uttarakhand, India.
[2*]Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, 247667, Uttarakhand, India.
[3]Department of Mathematics, Indian Institute of Technology Ropar, Rupnagar, 140001, Punjab, India.

*Corresponding author(s). E-mail(s): ali_r@cs.iitr.ac.in; aditi.gangopadhyay@ma.iitr.ac.in;
Contributing authors: mansi@ma.iitr.ac.in; mohakjot@gmail.com;

Abstract

The Learning with Errors (LWE) problem is a fundamental hardness assumption underlying most fully homomorphic encryption (FHE) schemes. Given the close relationship between the Learning with Rounding (LWR) and LWE problems, several cryptographic constructions have also been developed based on LWR. In particular, LWR-based schemes often benefit from simpler and more efficient implementations, as they eliminate the need for explicit Gaussian error sampling. Despite these advantages, relatively few FHE constructions in the literature are based on the LWR assumption.

At *AsiaCCS 2025*, Goyal and Gangopadhyay proposed a multi-key FHE (MKFHE) scheme based on LWR that adopts a public-key extension mechanism rather than the conventional ciphertext-extension paradigm. The authors also identified the development of a ring-based analogue as an open future direction.

In this work, we present two Ring-LWR-based MKFHE constructions that can be viewed as ring analogues of the LWR-based MKFHE scheme of Goyal and Gangopadhyay. To the best of our knowledge, these are the first MKFHE schemes based on the Ring-LWR assumption. Compared with existing Ring-LWE-based multi-key constructions, our schemes achieve improved compactness in terms of storage and communication costs. We provide concrete parameterizations supporting circuits of varying multiplicative depths and present a proof-of-concept implementation to validate our claims.

Keywords: MKFHE, LWR, LWE, BFV, Key extension

1 Introduction

Fully Homomorphic Encryption (FHE) is an advanced cryptographic primitive that enables the evaluation of arbitrary functions directly on encrypted data, without requiring decryption. In his groundbreaking work [1], Gentry provided a solution to a long-standing open problem in cryptography by introducing a *Somewhat Homomorphic Encryption* (SHE) scheme that can evaluate circuits up to a certain depth, and then modifying it to evaluate its own decryption circuit. Such a scheme, known as a bootstrappable SHE, is sufficient to construct a FHE scheme.

Following Gentry's pioneering contribution, numerous works have proposed improvements, simplifying his original construction and achieving more efficient implementations [2–7]. FHE enables a wide range of privacy-preserving applications, from healthcare and finance to secure cloud computing. While the original FHE schemes are designed for single-user settings, many real-world applications require collaborative computation over data encrypted under different keys. To address this need, López-Alt *et al.* [8] extended the FHE framework to the multiparty setting, introducing *Multi-Key Fully Homomorphic Encryption* (MKFHE).

In the MKFHE framework, each party independently encrypts its private data under its own public key and sends the resulting ciphertext to an untrusted service provider. The provider performs the requested computation directly on the encrypted inputs without accessing any plaintext, and the result is recovered through a collaborative decryption protocol requiring participation of all parties. As the data is encrypted with different keys, the ciphertexts are not directly compatible for evaluation. To address this, the MKFHE schemes implement either a ciphertext extension or a key extension algorithm. Before evaluation, the ciphertext extension algorithm extends the ciphertexts to make them suitable for evaluation, while in the key extension, each participant public key is combined to make an extended key, and parties encrypt their private data using that extended key, eliminating the need for ciphertext extension. The primary advantage of ciphertext extension is that each party can encrypt its data independently, without any prior coordination or knowledge of the other participants. This makes the approach particularly attractive in dynamic or asynchronous settings where the set of parties is not fixed in advance. The trade-off is an increase in the size of each ciphertext and additional per-party communication cost at the time of evaluation, both of which grow with the number of parties. Key extension schemes generally achieve lower per-operation evaluation costs than ciphertext extension alternatives, since ciphertexts retain the same dimensionality as in the single-key setting. However, they require an interactive key-aggregation phase in which all parties must coordinate before any data can be encrypted. This introduces additional communication rounds and assumes that the participant set is known and stable at the outset.

1.1 Learning with Rounding

Learning with Rounding (LWR), introduced by Banerjee *et al.* [9], as a deterministic alternative to Learning with Errors (LWE). It replaces random noise with a simple scaled rounding operation, making it faster and more efficient in practice. They provided the first formal reduction between the two problems. However, this early reduction required a superpolynomially large modulus, limiting its practical scope, a shortcoming that was later bridged by Alwen *et al.* [10] who demonstrated a tighter reduction valid for polynomial moduli. Bogdanov *et al.* [11] further strengthened it by proving that LWR remains hard even when an adversary is given multiple samples, which is essential for real-world cryptographic security. These results collectively indicate that LWR inherits its hardness from LWE. The latter enjoys a strong theoretical foundation through Regev's quantum worst-case to average-case reduction, which relates the average-case hardness of LWE to the worst-case hardness of approximating lattice problems such as GapSVP and SIVP [12]. The LWR and its ring variant have been widely adopted for the construction of efficient quantum-safe cryptographic primitives, including pseudorandom generators [9], public-key encryption schemes [13, 14], key encapsulation mechanisms [15], and digital signatures due to their advantages and ease of implementation in hardware setup [15]. In this work, we focus on the Ring Learning with Rounding (RLWR) problem, which inherits both the strong security guarantees of RLWE and the computational efficiency of the rounding-based approach.

Informally, the Ring Learning with Errors (RLWE) assumption states that, for a suitable polynomial ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(f(x))$, it is hard to distinguish samples $(a,\ a \cdot s + e) \in \mathcal{R}_q^2$ from uniformly random elements, where e is drawn from a small error distribution.

The RLWR assumption replaces the explicit error with deterministic rounding: for $q > p$, it is hard to distinguish samples $(a,\ \lfloor \frac{p}{q}\, a \cdot s \rceil)$ from uniformly random elements in $\mathcal{R}_q \times \mathcal{R}_p$. Thus RLWR is a derandomized variant of RLWE in which rounding replaces explicit error sampling.

1.2 Related Works

Following the pioneering MKFHE construction based on the NTRU assumption by López-Alt *et al.* [8], several MKFHE schemes based on the Learning With Errors (LWE) assumption have been proposed.

Clear and McGoldrick [16] introduced an LWE-based MKFHE scheme derived from the GSW homomorphic encryption scheme [4]. Their construction was subsequently optimized by Mukherjee and Wichs [17], who employed it to obtain an optimal two-round general Multi-Party Computation (MPC) protocol. Later works [18, 19] further extended these constructions to dynamic settings, allowing additional parties to join the homomorphic evaluation process at any stage.

The majority of existing MKFHE constructions [16–19] follow the *ciphertext extension* paradigm. In this approach, ciphertexts encrypting messages under different secret keys are transformed into higher-dimensional ciphertexts that can be viewed as encryptions under a combined secret key obtained by concatenating the secret

keys of all participating parties. This enables homomorphic operations on ciphertexts originating from different users while preserving correctness and security.

An alternative approach avoids ciphertext extension by employing *public-key extension*. In this paradigm, the public keys of all parties participating in the computation are combined into a single extended public key that can be used for encryption. Decryption then requires each participating party to contribute a partial decryption share. This approach has been adopted in works such as [20, 21].

To improve efficiency and reduce ciphertext expansion, several ring-based constructions have subsequently been proposed. For example, Chen *et al.* [22] introduced an MKFHE scheme based on the BGV homomorphic encryption scheme [2]. Their construction follows the ciphertext extension paradigm and achieves a significantly improved ciphertext-to-plaintext ratio compared to GSW-based schemes, whose ciphertexts inherently encode only a single bit. In contrast, Mouchet *et al.* [20] proposed a RLWE-based multiparty homomorphic encryption scheme derived from the BFV scheme [3]. Their construction adopts the public-key extension paradigm rather than ciphertext extension and serves as the foundation for an efficient MPC protocol.

1.3 Our Work

In this work, we present the first MKFHE construction based on the RLWR assumption instead of RLWE assumption on which most existing works are based. In particular, we provide two constructions:

- **Construction I (MKFHE-I):** extends the SHE scheme of [23] to the multi-key setting.
- **Construction II (MKFHE-II):** introduces Simplified-SHE scheme and extends it to the multi-key setting.

Our constructions adopt the *public-key extension* paradigm of [20, 21], in which a joint public key is constructed from the individual public keys of the participating parties. Consequently, any ciphertext encrypted under the extended public key can only be decrypted through the cooperation of all parties, each contributing a partial decryption share. The plaintext is then recovered by combining these shares (see Figure 1). Consequently, the communication complexity of the partial decryption poses the central efficiency bottleneck.

We demonstrate that the modulus-reduction advantages of RLWR observed in [23] become even more pronounced in the multi-key setting. While proposed MKFHE-II attains the lowest communication cost for distributed decryption, the MKFHE-I provides more compact public keys and ciphertexts compared to RLWE-MKFHE construction in [20]. A summary of the comparison is presented in Table 1. This exhibits a butterfly-effect-like behavior: a small reduction in modulus size leads to a noticeable decrease in communication cost.

Although the reduction in sizes seems modest in absolute terms, it represents a consistent improvement in communication-intensive deployments. To illustrate, consider a practical setting with multiplicative depth $d = 7$ and $k = 128$ participating parties.

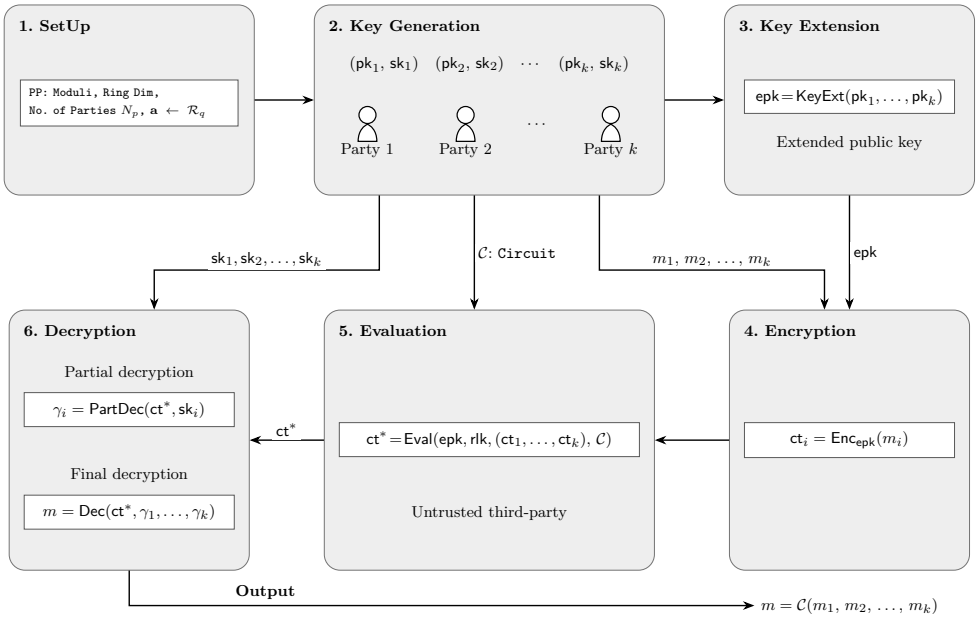


Figure 1: Overview of MKFHE protocol with public key extension paradigm.

At this depth, the partial decryption share size is 824 kB for the RLWE-MKFHE, compared to 816 kB and 814 kB for MKFHE-I and II, respectively. This yields per-party communication savings of 8 kB and 10 kB for each distributed decryption. Aggregated across all 128 parties, these savings amount to approximately 1 MB and 1.25 MB per decryption, respectively. Over 1,000 decryptions, the cumulative communication reduction reaches approximately 1 GB and 1.25 GB, respectively.

Importantly, these communication savings require no additional computational cost. Moreover, unlike RLWE-MKFHE, the proposed schemes eliminate the need for explicit Gaussian error sampling, thereby reducing implementation complexity and improving efficiency, particularly in hardware-oriented settings. The reduction in partial decryption communication is therefore only one aspect of a broader efficiency advantage that is consistently observed across all considered parameter sets. In communication-intensive applications, such as secure multi-party computation protocols involving repeated homomorphic evaluations and distributed decryptions, these cumulative savings can become practically significant as both the number of participating parties and the number of decryption queries increase.

To enable a direct comparison with the reference RLWE-MKFHE, we revisit the error-growth analysis for circuits of multiplicative depth d following [20, 24]. We subsequently compare these bounds with those obtained for the proposed RLWR-based constructions.

In summary, our contribution can be summarized in the following points:

Table 1: Comparison of the proposed MKFHE-I and II with the Ring-LWE-based MKFHE scheme of [20]. For the considered parameter sets, $q \approx q''$ and $q' \approx 8q$.

Parameter	MKFHE-**I**	MKFHE-**II**	MKFHE [20]
$\lvert\mathsf{pk}\rvert$	$n\log(qp_1)$	$n\log(q'p_1')$	$2n\log q''$
$\lvert\mathsf{rlk}\rvert$	$n\log_w q \cdot \log(p_1p_2)$	$n\log_w p_1' \cdot \log(p_1'p_2')$	$2n\log_w q''\log q''$
$\lvert\mathsf{epk}\rvert$	$n\log(qp_1)$	$n\log(q'p_1')$	$2n\log q''$
$\lvert\mathsf{ct}\rvert$	$n\log(p_1p_2)$	$n\log(p_1'p_2')$	$2n\log q''$
$\lvert\gamma_i\rvert$	$n\log p_1$	$n\log p_2'$	$n\log q''$
Security Assumption	RLWR_{n,q,p_1}	$\mathsf{RLWR}_{n,q',p_1'}$	$\mathsf{RLWE}_{n,\chi,q''}$
Constraints	$p_1 \mid q,\ p_2q = p_1^2$	$p_1' \mid q',\ p_2'q' = (p_1')^2$	—

- **First MKFHE from RLWR:** Two MKFHE constructions in the public-key extension setting, one based on [23] which shows overall compactness compared to RLWE construction, and MKFHE-II further saves in the distributed decryption.
- **Parameter selection and error analysis:** Detailed error-growth analysis for depth-d circuits, with concrete 128-bit-secure parameter sets for $k \in \{2, 4, 8, 128\}$ parties.
- **Proof-of-concept implementation:** Both constructions are implemented and publicly available at: https://github.com/The-Isogeniest/butterfly_effect. The implementation confirms correct decryption across all proposed parameter sets, validating the error analysis.

1.4 Technical Overview

Construction I: MKFHE-**I.**

Our first construction extends [23] directly to the multi-key setting and consists of the following algorithms:

$$\mathsf{MKFHE\text{-}I} = \{\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}\}.$$

Each party independently runs KeyGen to produce a key pair $\mathsf{pk}_i = (b_i = \lfloor a \cdot s_i \rceil_{p_1,q},\ a) \in \mathcal{R}_{p_1} \times \mathcal{R}_q$, $\mathsf{sk}_i = s_i$, where $a \leftarrow \mathcal{R}_q$ is a common reference polynomial shared among all parties and $s_i \leftarrow \chi$ is the secret. The KeyExt algorithm aggregates all individual public keys $\{\mathsf{pk}_i\}$ into a single extended public key

$$\mathsf{pk} = (\bar{b},\ a), \qquad \bar{b} = \sum_i b_i.$$

Despite using their public key, each party uses the extended key to encrypt their messages, ensuring all ciphertexts are compatible for joint evaluation. For a message

$m \in \mathcal{R}_t$, the resulting ciphertext is

$$\mathsf{ct} = (c_0,\ c_1) \in \mathcal{R}_{p_1} \times \mathcal{R}_{p_2},$$

where

$$c_0 = \Delta m + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1} \bmod p_2, \qquad c_1 = \lfloor a \cdot r \rceil_{p_1,q}.$$

The decryption is a multi-party protocol, as the ciphertext contains the public key share of each party. To decrypt a ciphertext $\mathsf{ct} = (c_0,\ c_1)$, each party computes and shares $\gamma_i = c_1 \cdot s_i + e_i^{sm}$, where e_i^{sm} is smudging noise.

Subsequently, the final decryption is given as $m' = \left\lfloor \frac{t}{p_2} \left(c_0 - \frac{p_2}{p_1} \sum_{i \in [k]} \gamma_i \right)_{p_2} \right\rceil_t$. The decryption is correct if $m = m'$.

***A Simplified*-SHE *Scheme:* Simplified-SHE.**

As an intermediate contribution, we introduce a simplified variant of the SHE scheme of Bolboceanu *et al.* [23], defined as

$$\text{Simplified-SHE} = \{\mathsf{SetUp},\ \mathsf{KeyGen},\ \mathsf{Enc},\ \mathsf{Eval},\ \mathsf{Dec}\}.$$

The SetUp and KeyGen algorithms follow [23] with one modification: the public key is computed as

$$\mathsf{pk} = (b = \lfloor -a \cdot s \rceil_{p_1,q},\ a) \in \mathcal{R}_{p_1} \times \mathcal{R}_q.$$

For a message $m \in \mathcal{R}_t$, the ciphertext is

$$\mathsf{ct} = (c_0,\ c_1) \in \mathcal{R}_{p_2}^2,$$

where

$$c_0 = \Delta m + \lfloor b \cdot r \rceil_{p_2,p_1} \bmod p_2, \qquad c_1 = \lfloor a \cdot r \rceil_{p_2,q}.$$

The key structural improvement over [23] is that *both* ciphertext components now reside in the same ring $\mathcal{R}_{p_2}$, in contrast to the original scheme where $c_0 \in \mathcal{R}_{p_1}$ and $c_1 \in \mathcal{R}_{p_2}$ (that is, they belong to distinct rings). This uniform ring structure simplifies both the algebraic analysis and the implementation.

***Construction II:* MKFHE-II.**

Our second multi-key construction, MKFHE-II, lifts Simplified-SHE to the multi-party setting using the same key extension mechanism as MKFHE-I. In particular, the extended public key $\mathsf{pk} = (\bar{b},\ a)$ is formed identicall. For a message $m \in \mathcal{R}_t$, the resulting ciphertext is

$$\mathsf{ct} = (c_0,\ c_1) \in \mathcal{R}_{p_2}^2,$$

where

$$c_0 = \Delta m + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1} \bmod p_2, \qquad c_1 = \lfloor a \cdot r \rceil_{p_2,q}.$$

The ciphertext obtained here has both components in $\mathcal{R}_{p_2}$, inheriting the uniform ring structure of the simplified single-key scheme and enabling more savings in communications while sharing the partial decryption shares.

To decrypt a ciphertext $\mathsf{ct} = (c_0,\ c_1)$, each party computes and shares $\gamma_i = c_1 \cdot s_i + e_i^{sm}$, where e_i^{sm} is smudging noise. Subsequently, the final decryption is given as $m' = \left\lfloor \frac{t}{p_2} \left(c_0 + \sum_{i \in [k]} \gamma_i \right)_{p_2} \right\rceil_t$. The decryption is correct if $m = m'$.

2 Preliminaries

Notations

Throughout the paper, we use the notation summarized below.

Notation	Meaning
λ	Security parameter
$q,\ p_1,\ p_2$	Moduli
$f(x)$	Cyclotomic polynomial of degree n
$\mathcal{R},\ \mathcal{R}_q$	$\mathbb{Z}[x]/(f(x)), \quad \mathbb{Z}_q[x]/(f(x))$
$\delta_{\mathcal{R}}$	$\max\{\|a \cdot b\|_\infty / (\|a\|_\infty \|b\|_\infty) \ :\ a,\ b \in \mathcal{R}\}$
$\|a\|_\infty$	$\max\{\|a_i\| : i = 0, \ldots, n-1\}$ for $a = \sum_i a_i x^i$
$\lfloor u \rfloor,\ \lceil u \rceil,\ \lfloor u \rceil$	Floor, ceiling, and nearest integer (ties rounded up)
$(u)_p$	$u \bmod p$
$\lfloor u \rceil_p$	$\lfloor u \rceil \bmod p$
$\lfloor a \rceil_{p,q}$	$\lfloor \frac{p}{q} a \rceil \bmod p$, for integers $q > p \geq 2$ and $a \in \mathcal{R}_q$
Δ	$\lfloor p_2/t \rfloor$
$r_t(p_2)$	$p_2 - t \lfloor p_2/t \rfloor$
$\approx_c$	Computationally indistinguishable
$\mathcal{U}(X)$	Uniform distribution over X
$a \leftarrow \mathcal{R}_q$	Sample a uniformly at random from $\mathcal{R}_q$
$[m]$	$\{1, 2, \ldots, m\}$
$\Pr[X]$	Probability of event X

2.1 Hard Assumptions

Lattice-based cryptography relies on hardness assumptions believed to be secure against quantum attacks. Here, we mention the relevant assumptions used in this work.

Ring Learning with Errors (RLWE)

Definition 1 (RLWE Distribution) For modulus q, ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(f(x))$, and error distribution χ bounded by β_χ:

$$\mathsf{RLWE}_{q,n,\chi} = \{(a,\ b = a \cdot s + e) \mid a \leftarrow \mathcal{R}_q,\ s \leftarrow \mathcal{R}_q,\ e \leftarrow \chi\}.$$

Definition 2 (Decisional RLWE Problem) For security parameter λ, $n = n(\lambda)$ degree of the ring $\mathcal{R}$. Let $q = q(\lambda)$ be a large modulus and $\chi = \chi(\lambda)$ be an error distribution on $\mathcal{R}$ bounded by β_χ. The Decisional RLWE problem is to distinguish between

- $\mathcal{F}_0 = \{(a,\ b) \in \mathcal{R}_q \times \mathcal{R}_q \mid b = a \cdot s + e\}$, where $a,\ s \in \mathcal{R}_q$ are sampled uniformly and $e \leftarrow \chi$.

- $\mathcal{F}_1 = \{(a,\ b) \in \mathcal{R}_q \times \mathcal{R}_q\}$, where $a,\ b$ are sampled uniformly from $\mathcal{R}_q$.

Decisional RLWE Assumption. For any Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$,

$$|\Pr[\mathcal{A}(a,\ b) = 1 \mid (a,\ b) \in \mathcal{F}_0] - \Pr[\mathcal{A}(a,\ b) = 1 \mid (a,\ b) \in \mathcal{F}_1]| = \text{negl}(\lambda).$$

It is worth noting that the hardness of the average-case RLWE problem is supported by a quantum worst-case-to-average-case reduction from certain lattice problems (such as approximate SIVP and related problems) on ideal lattices [25]. Sampling s from χ instead of uniformly does not weaken security [25, 26], and q need not be prime [27].

Ring Learning with Rounding (RLWR)

Definition 3 (RLWR Distribution) For $q > p \geq 2$:

$$\mathsf{RLWR}_{q,p,n} = \{(a,\ \lfloor a \cdot s \rceil_{p,q}) \mid a,\ s \leftarrow \mathcal{R}_q\}.$$

Definition 4 (Decisional RLWR Problem) For a security parameter λ, $n = n(\lambda)$ degree of the ring $\mathcal{R}$. Let $q = q(\lambda)$, $p = p(\lambda)$ be two large moduli. The RLWR problem is to distinguish between

- $\mathcal{D}_0 = \{(a,\ b) \in \mathcal{R}_q \times \mathcal{R}_p \mid b = \lfloor a \cdot s \rceil_{p,q}\}$, where $a,\ s \in \mathcal{R}_q$ are uniformly chosen.
- $\mathcal{D}_1 = \{(a,\ b) \in \mathcal{R}_q \times \mathcal{R}_p\}$, where $a,\ b$ are sampled uniformly from $\mathcal{R}_q$ and $\mathcal{R}_p$, respectively.

Decisional RLWR Assumption. For any PPT adversary $\mathcal{A}$,

$$|\Pr[\mathcal{A}(a,\ b) = 1 \mid (a,\ b) \in \mathcal{D}_0] - \Pr[\mathcal{A}(a,\ b) = 1 \mid (a,\ b) \in \mathcal{D}_1]| = \text{negl}(\lambda).$$

3-Moduli RLWR

The security of our schemes also relies on the following assumption from [23].

Definition 5 (3-Moduli RLWR Distribution) For $q > p_1 > p_2 \geq 2$ and secret s with coefficients in $[-\gamma,\ \gamma]$:

$$\mathsf{RLWR}_{n,q,p_1,p_2} = \{(a,\ b,\ c,\ d) \mid b = \lfloor a \cdot s \rceil_{p_1,q},\ d = \lfloor c \cdot s \rceil_{p_2,p_1}\},$$

where $a \leftarrow \mathcal{R}_q$ and $c \leftarrow \mathcal{R}_{p_1}$.

Definition 6 (Decisional 3-Moduli RLWR Problem) For a security parameter λ, $n = n(\lambda)$ degree of the ring $\mathcal{R}$. Let $q = q(\lambda)$, $p_1 = p_1(\lambda)$, and $p_2 = p_2(\lambda)$ be three large moduli. The decisional 3-moduli ring learning with rounding problem (denoted by Decisional 3-moduli $\mathsf{RLWR}_{n,q,p_1,p_2}$) is to distinguish between

- $\mathcal{I}_0 = \{(a,\ b,\ c,\ d) \in \mathcal{R}_q \times \mathcal{R}_{p_1} \times \mathcal{R}_{p_1} \times \mathcal{R}_{p_2}\}$, where $a,\ b,\ c$, and d are sampled uniformly from the respective rings.

- $\mathcal{I}_1 = \{(a,\ b,\ c,\ d) \in \mathcal{R}_q \times \mathcal{R}_{p_1} \times \mathcal{R}_{p_1} \times \mathcal{R}_{p_2} \mid b = \lfloor a \cdot s \rceil_{p_1,q},\ d = \lfloor c \cdot s \rceil_{p_2,p_1}\}$, where a, and c are sampled uniformly.

Lemma 1 ([23, Lemma 1]) The map $(u,\ v) \mapsto (u \bmod p_1,\ v \bmod p_2)$ sends $\mathsf{RLWR}_{n,\eta p_1,\eta p_2}$ samples to $\mathsf{RLWR}_{n,p_1,p_2}$ samples, and uniform samples to uniform samples.

Theorem 1 ([23, Theorem 1]) *If $p_1 \mid q$ and $qp_2 = p_1^2$, then the decisional 3-moduli $\mathsf{RLWR}_{n,q,p_1,p_2}$ problem reduces to RLWR_{n,q,p_1} with the same advantage.*

Lemma 2 (Smudging [28]) Let $a_1 \in [-D_1,\ D_1]$ be fixed and $a_2 \leftarrow [-D_2,\ D_2]$ uniform. If $D_1/D_2 = \mathrm{negl}(\lambda)$, then a_2 and $a_2 + a_1$ are statistically indistinguishable.

2.2 Threat Model and Security

2.2.1 Adversarial Model

We consider a *passive* (honest-but-curious) adversary model: adversaries follow the protocol faithfully but attempt to extract private information of other participants.

Adversary Types.

- *External:* operates outside the protocol; passively observes public keys, ciphertexts, and partial decryption shares over public channels.
- *Internal:* a registered participant in key generation, key extension, and/or encryption; may also compute partial decryptions.

Capabilities.

- *Eavesdropping:* both adversary types may intercept all public-channel communications.
- *Corruption ($k-1$ model):* the adversary may corrupt up to all but one honest party, obtaining their secret keys, plaintext-ciphertext pairs, and partial decryption shares.
- *Encryption oracle:* the adversary may request encryptions of arbitrary plaintexts.

2.2.2 Security Goals

A robust MKFHE scheme must ensure:

– *Ciphertext Confidentiality.* Ciphertexts must be computationally indistinguishable from random, even to an adversary holding all but one secret key.
– *Simulation-Security of Partial Decryption.* Partial decryption shares must not leak information beyond the final output.

Since the evaluation algorithm is public, security rests on the IND-CPA security of the underlying base scheme.

1. **Setup.** Challenger Ch generates public parameters and its key pair $(\mathsf{pk}_r,\ \mathsf{sk}_r)$; publishes pk_r.
2. **Corruption.** $\mathcal{A}$ obtains $\{(\mathsf{pk}_j,\ \mathsf{sk}_j)\}_{j\neq r}$ for all $k-1$ other parties; submits two messages $m_0,\ m_1 \in \mathcal{R}_t$ to Ch.
3. **Challenge.** Ch computes $\mathsf{epk} \leftarrow \mathsf{KeyExt}(\{\mathsf{pk}_i\})$, samples $\alpha \leftarrow \{0,1\}$, and sends $\mathsf{ct}_\alpha \leftarrow \mathsf{Enc}(m_\alpha,\ \mathsf{epk})$ to $\mathcal{A}$.
4. **Guess.** $\mathcal{A}$ outputs α'; wins if $\alpha' = \alpha$.

Figure 2: The IND-CPA security game for $\mathsf{MKFHE}_{\mathsf{Basic}}$.

Definition 7 (IND-CPA Security) A scheme $\mathsf{MKFHE}_{\mathsf{Basic}} = \{\mathsf{SetUp},\ \mathsf{KeyGen},\ \mathsf{KeyExt},\ \mathsf{Enc},\ \mathsf{Dec}\}$ is IND-CPA *secure* if for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}CPA}} = \left|\Pr\left[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{MKFHE}_{\mathsf{Basic}}}(1^\lambda,\ \alpha) = 1\right] - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda),$$

where the security game is described in Figure 2.

2.3 Underlying SHE Scheme: SHE [23]

In this section, we restate the SHE scheme of [23].

$$\mathsf{SHE} = \{\mathsf{SetUp},\ \mathsf{KeyGen},\ \mathsf{Enc},\ \mathsf{Eval},\ \mathsf{Dec}\}.$$

$\mathsf{PP} \leftarrow \mathsf{SetUp}(1^\lambda,\ 1^d)$: Takes an input the security parameter λ and circuit depth d. Output $\mathsf{PP} = (n,\ q,\ p_1,\ p_2,\ t,\ w,\ \chi)$, where n the degree of polynomial $f(x) \in \mathbb{Z}[x]$, $\mathcal{R} = \mathbb{Z}[x]/(f(x))$, $q > p_1 > p_2 \geq 2$ are large moduli, $t \geq 2$ is an integer, $w \geq 2$ is a radix, and χ is a probability distribution on $\mathcal{R}$.

$(\mathsf{pk},\ \mathsf{sk},\ \mathsf{rlk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$: Takes as input PP. Sample $s \leftarrow \chi$, $a \leftarrow \mathcal{R}_q$. Compute $b = \lfloor a \cdot s \rceil_{p_1,q}$. Set $\mathsf{pk} = (b,\ a) \in \mathcal{R}_{p_1} \times \mathcal{R}_q$ and $\mathsf{sk} = s$. For $i = 0, \ldots, l-1$ (where $l = \lceil \log_w q \rceil$), sample $a_i \leftarrow \mathcal{R}_{p_1}$ and set

$$\mathsf{rlk}[i] = \left(\lfloor a_i \cdot s \rceil_{p_2,p_1} + \left\lfloor \frac{p_2^2}{p_1^2} w^i s^2 \right\rceil \bmod p_2,\ a_i\right).$$

$\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$: Takes as input the message m and pk. Sample $r \leftarrow \chi$. Output $\mathsf{ct} = (c_0,\ c_1) \in \mathcal{R}_{p_2} \times \mathcal{R}_{p_1}$ with

$$c_0 = \Delta m + \lfloor b \cdot r \rceil_{p_2,p_1} \bmod p_2, \qquad c_1 = \lfloor a \cdot r \rceil_{p_1,q}.$$

$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Takes as input sk, ct. Compute

$$m' = \left\lfloor \frac{t}{p_2}\left(c_0 - \frac{p_2}{p_1} c_1 \cdot s\right)_{p_2} \right\rceil_t.$$

The decryption is correct if $m = m'$.

$\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{pk},\ \mathsf{rlk},\ (\mathsf{ct}_1,\ \mathsf{ct}_2),\ \mathcal{C})$: It takes as input pk, rlk, ct_1, ct_2, and a circuit $\mathcal{C}$. For $\mathsf{ct}_j = (c_{j0},\ c_{j1})$, $j \in \{1,\ 2\}$:

- $\mathsf{ct}_{\mathsf{add}} = (c_{10} + c_{20} \bmod p_2,\ c_{11} + c_{21} \bmod p_1)$.
- For multiplication, first compute the product:

$$c_0 = \left\lfloor \frac{t}{p_2} c_{10} \cdot c_{20} \right\rceil_{p_2}, \quad c_1 = \left\lfloor \frac{t}{p_2} (c_{10} \cdot c_{21} + c_{11} \cdot c_{20}) \right\rceil_{p_1}, \quad c_2 = \left\lfloor \frac{t}{p_2} c_{11} \cdot c_{21} \right\rceil_{q}.$$

  Then *relinearise*: write $c_2 = \sum_{i=0}^{l-1} c_2^{(i)} w^i$ and set

$$c_0' = \left(c_0 + \sum_i \mathsf{rlk}[i][0] \cdot c_2^{(i)} \right)_{p_2}, \quad c_1' = \left(c_1 + \sum_i \mathsf{rlk}[i][1] \cdot c_2^{(i)} \right)_{p_1}.$$

Theorem 2 ([23, Theorem 2]) *Under $p_1 \mid q$ and $p_2 q = p_1^2$,* SHE *is* IND-CPA *secure assuming hardness of* RLWR_{n,q,p_1}.

3 Construction I: MKFHE scheme based on SHE

Our proposed MKFHE-I extends SHE [23] to the multi-key setting using the public-key extension paradigm of [20, 21]. The scheme consists of the following algorithms:

$$\text{MKFHE-I} = \{\mathsf{SetUp},\ \mathsf{KeyGen},\ \mathsf{KeyExt},\ \mathsf{Enc},\ \mathsf{Eval},\ \mathsf{Dec}\}.$$

$\mathsf{PP} \leftarrow \mathsf{MKFHE.SetUp}(1^\lambda,\ 1^d)$: Output $\mathsf{PP} = (n,\ q,\ p_1,\ p_2,\ t,\ w,\ N_p,\ a,\ \chi)$, where N_p is the total number of parties, $a \in \mathcal{R}_q$ is a common public and the other parameters are the same as of SHE (Section 2.3).

$(\mathsf{pk}_i,\ \mathsf{sk}_i) \leftarrow \mathsf{MKFHE.KeyGen}(\mathsf{PP})$: Party i samples $s_i \leftarrow \chi$ and computes $b_i = \lfloor a \cdot s_i \rceil_{p_1,q}$. Set $\mathsf{pk}_i = (b_i,\ a) \in \mathcal{R}_{p_1} \times \mathcal{R}_q$ and $\mathsf{sk}_i = s_i$.

Relinearisation Key. Let k be the number of the *participating* parties. Then, the k parties jointly compute rlk via Algorithm 1.

$\mathsf{epk} \leftarrow \mathsf{MKFHE.KeyExt}(\{\mathsf{pk}_i\}_{i\in[k]})$: Output $\mathsf{epk} = (\bar{b},\ a)$, where $\bar{b} = \sum_{i\in[k]} b_i \bmod p_1$.

$\mathsf{ct} \leftarrow \mathsf{MKFHE.Enc}(m,\ \mathsf{epk})$: Identical to SHE (Section 2.3) with b replaced by $\bar{b}$:

$$c_0 = \Delta m + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1} \bmod p_2, \qquad c_1 = \lfloor a \cdot r \rceil_{p_1,q}.$$

$\hat{\mathsf{ct}} \leftarrow \mathsf{MKFHE.Eval}(\mathsf{epk},\ \mathsf{rlk},\ (\mathsf{ct}_1,\ \mathsf{ct}_2),\ \mathcal{C})$: Identical to SHE (Section 2.3).

$m' \leftarrow \mathsf{MKFHE.Dec}(\{s_i\}_{i\in[k]},\ \mathsf{ct})$: It is a multi-party protocol. Each party partially decrypts the ciphertext using its secret key and discloses it.

Algorithm 1 Relinearisation Key Generation Protocol (MKFHE-I)

Input: Public: $\mathbf{a}_t = (a_1, \ldots, a_l) \in \mathcal{R}_q^l$, $\mathbf{w} = (1, w, \ldots, w^{l-1})$, $l = \lceil \log_w q \rceil$
Input: Private: $\{s_i\}_{i\in[k]}$
Output: $\mathsf{rlk} = (\mathbf{h}'_0 + \mathbf{h}'_1, \mathbf{h}_1)$

— *Round 1* —

1: **for** each party $i \in [k]$ **do**
2: Sample $u_i \leftarrow \chi$; compute and broadcast

$$(\mathbf{h}_{0,i}, \mathbf{h}_{1,i}) = \left(\lfloor u_i \mathbf{a}_t \rceil_{p_1,q} + \left\lfloor \frac{p_2}{p_1} s_i \mathbf{w} \right\rceil \bmod p_1,\ \lfloor s_i \mathbf{a}_t \rceil_{p_1,q} \right)$$

3: **end for**
4: All parties aggregate: $\mathbf{h}_0 = \sum_i \mathbf{h}_{0,i}$, $\mathbf{h}_1 = \sum_i \mathbf{h}_{1,i}$

— *Round 2* —

5: **for** each party $i \in [k]$ **do**
6: Compute and broadcast

$$(\mathbf{h}'_{0,i}, \mathbf{h}'_{1,i}) = (\lfloor s_i \mathbf{h}_0 \rceil_{p_2,p_1},\ \lfloor (s_i - u_i)\mathbf{h}_1 \rceil_{p_2,p_1})$$

7: **end for**
8: Aggregate: $\mathbf{h}'_0 = \sum_i \mathbf{h}'_{0,i}$, $\mathbf{h}'_1 = \sum_i \mathbf{h}'_{1,i}$; **return** $\mathsf{rlk} = (\mathbf{h}'_0 + \mathbf{h}'_1, \mathbf{h}_1)$

- *Partial decryption.* Each party i computes and broadcasts $\gamma_i = c_1 \cdot s_i + e_i^{sm}$, where e_i^{sm} is smudging noise whose norm depends on the norm of the error term in the ciphertext.
- *Final decryption.* $m' = \left\lfloor \frac{t}{p_2} \left(c_0 - \frac{p_2}{p_1} \sum_{i\in[k]} \gamma_i \right)_{p_2} \right\rceil_t$. The decryption is correct if $m = m'$.

3.1 Correctness of Decryption

In this section, we analyze the correctness of the decryption procedure for both fresh (which does not undergo evaluation) and evaluated ciphertexts. We write $s = \sum_{i\in[k]} s_i$ and denote the decryption inner product of any ciphertext $(c_0,\ c_1)$ as

$$D = c_0 - (p_2/p_1) \sum_{i\in[k]} \gamma_i.$$

Case 1: Fresh ciphertext. Expanding D using $\lfloor \bar{b} \cdot r \rceil_{p_2,p_1} = \frac{p_2}{p_1}\bar{b} \cdot r + e_0$ with $\|e_0\|_\infty \leq \frac{1}{2}$, $\bar{b} = \sum_{i\in[k]} b_i$, $b_i = (p_1/q)a \cdot s_i + e_i$ with $\|e_i\|_\infty \leq \frac{1}{2}$, and $c_1 = (p_1/q)a \cdot r + e'$

with $\|e'\|_\infty \le \frac{1}{2}$, the $a \cdot r \cdot s$ terms cancel and we obtain

$$D = \Delta m + y + p_2 h - \frac{p_2}{p_1} \sum_{i \in [k]} e_i^{sm},$$

where $h \in \mathcal{R}$ with

$$\|h\|_\infty \le \frac{\delta_{\mathcal{R}} \|s\|_\infty + 3}{2} \tag{1}$$

and $y = \frac{p_2}{p_1} \sum_i e_i \cdot r + e_0 - \frac{p_2}{p_1} e' \cdot s$ absorbs the rounding errors, with

$$\|y\|_\infty \le \frac{p_2}{p_1} \delta_{\mathcal{R}} \|s\|_\infty + \frac{1}{2}. \tag{2}$$

After multiplying by t/p_2 and rounding, decryption is correct provided

$$\left\| -\frac{r_t(p_2)}{p_2} m + \frac{t}{p_2} y - \frac{t}{p_1} \sum_{i \in [k]} e_i^{sm} \right\|_\infty < \frac{1}{2}. \tag{3}$$

Case 2: Evaluated ciphertext. Let $\mathsf{ct}_j = (c_{j0},\ c_{j1})$ satisfy $c_{j0} - (p_2/p_1) c_{j1} \cdot s = \Delta m_j + y_j + p_2 h_j$ for $j = 1, 2$.

Correctness of Addition. For $\mathsf{ct}_{\mathsf{add}} = (c_{10} + c_{20},\ c_{11} + c_{21})$, expanding the decryption inner product gives

$$\begin{aligned} D_{\mathsf{add}} &= \Delta m_1 + y_1 + \Delta m_2 + y_2 + p_2(\ldots) - \frac{p_2}{p_1} \sum_{i \in [k]} e_i^{sm} \\ &= \Delta((m_1 + m_2)_t + t r_3) + y_1 + y_2 + p_2(\ldots) - \frac{p_2}{p_1} \sum_{i \in [k]} e_i^{sm} \\ &= \Delta(m_1 + m_2)_t + y_{\mathsf{add}} + p_2(\ldots) - \frac{p_2}{p_1} \sum_{i \in [k]} e_i^{sm} \end{aligned}$$

where $y_{\mathsf{add}} = -r_t(p_2) r_3 + y_1 + y_2$ (with $\|r_3\|_\infty \le 1$), so

$$\|y_{\mathsf{add}}\|_\infty \le t + \|y_1\|_\infty + \|y_2\|_\infty. \tag{4}$$

Decryption is correct if $\| - \frac{r_t(p_2)}{p_2}(m_1 + m_2)_t + \frac{t}{p_2} y_{\mathsf{add}} - \frac{t}{p_1} \sum_{i \in [k]} e_i^{sm} \|_\infty < \frac{1}{2}$.

Correctness of Multiplication. For $\mathsf{ct}_{\mathsf{mult}} = (c_0',\ c_1')$, expanding $D_{\mathsf{mult}} = c_0' - (p_2/p_1) c_1' \cdot s - (p_2/p_1) \sum_{i \in [k]} e_i^{sm}$, after expanding, we get

$$\begin{aligned} D_{\mathsf{mult}} = \frac{t}{p_2}(c_{10} \cdot c_{20} - \frac{p_2}{p_1}(c_{10} \cdot c_{21} + c_{11} \cdot c_{20}) \cdot s + \frac{p_2^2}{p_1^2}(c_{11} \cdot c_{21}) \cdot s^2) - r_a + e'' \\ + p_2 g - \frac{p_2}{p_1} \sum_{i \in [k]} e_i^{sm}, \end{aligned}$$

where

- r_a is the rounding error from the tensor product, bounded by

$$\|r_a\|_\infty \leq \frac{1 + \frac{p_2}{p_1}\delta_{\mathcal{R}}\|s\|_\infty + \frac{p_2^2}{p_1^2}\delta_{\mathcal{R}}^2\|s\|_\infty^2}{2}; \tag{5}$$

- e'' is the relinearisation error, bounded by

$$\|e''\|_\infty \leq kwl\delta_{\mathcal{R}}^2\|s\|_\infty\frac{p_2}{p_1} + \frac{1}{2}kwl\delta_{\mathcal{R}}; \tag{6}$$

- $g \in \mathcal{R}$ is a bounded integer carry.

$$\begin{aligned}
D_{\mathsf{mult}} &= \frac{t}{p_2}(c_{10} - \frac{p_2}{p_1}c_{11}\cdot s)\cdot(c_{20} - \frac{p_2}{p_1}c_{21}\cdot s) - r_a + e'' + p_2(\ldots) - \frac{p_2}{p_1}\sum_{i\in[k]} e_i^{sm} \\
&= \frac{t}{p_2}(\Delta m_1 + y_1 + p_2 h_1)\cdot(\Delta m_2 + y_2 + p_2 h_2) - r_a + e'' + p_2(\ldots) - \frac{p_2}{p_1}\sum_{i\in[k]} e_i^{sm} \\
&= \Delta(1 - \frac{r_t(p_2)}{p_2})((m_1\cdot m_2)_t + tr_m) + (1 - \frac{r_t(p_2)}{p_2})(m_1\cdot y_2 + y_1\cdot m_2) \\
&\quad - r_t(p_2)(m_1\cdot h_2 + m_2\cdot h_1) + t(y_1\cdot h_2 + y_2\cdot h_1) + \frac{t}{p_2}y_1\cdot y_2 - r_a \\
&\quad + e'' + p_2(\ldots) - \frac{p_2}{p_1}\sum_{i\in[k]} e_i^{sm}, \text{ where } \|r_m\|_\infty = \frac{\delta_{\mathcal{R}}t}{2} \\
D_{\mathsf{mult}} &= \Delta(m_1\cdot m_2)_t + y_{\mathsf{mult}} + p_2(\cdots) - \frac{p_2}{p_1}\sum_{i\in[k]} e_i^{sm}.
\end{aligned}$$

Using the bound 1, 2, 5, and 6, we get

$$\|y_{\mathsf{mult}}\|_\infty \leq \delta_{\mathcal{R}}t\left(\frac{\delta_{\mathcal{R}}\|s\|_\infty}{2} + 2\right)(\|y_1\|_\infty + \|y_2\|_\infty) + \frac{\delta_{\mathcal{R}}}{2}\min\{\|y_1\|_\infty, \|y_2\|_\infty\} + C_{\mathsf{rk}}, \tag{7}$$

with constant

$$\begin{aligned}
C_{\mathsf{rk}} &= r_t(p_2)\delta_{\mathcal{R}}t\left(\frac{\delta_{\mathcal{R}}\|s\|_\infty}{2} + 2.5\right) + \frac{1 + \frac{p_2}{p_1}\delta_{\mathcal{R}}\|s\|_\infty + \frac{p_2^2}{p_1^2}\delta_{\mathcal{R}}^2\|s\|_\infty^2}{2} + lwk\delta_{\mathcal{R}}^2\|s\|_\infty\frac{p_2}{p_1} \\
&\quad + \frac{1}{2}klw\delta_{\mathcal{R}}.
\end{aligned}$$

Decryption is correct if $\|\frac{t}{p_2}y_{\mathsf{mult}} - \frac{r_t(p_2)}{p_2}(m_1\cdot m_2)_t - \frac{t}{p_1}\sum_{i\in[k]} e_i^{sm}\|_\infty < \frac{1}{2}$.

3.2 Error Analysis

Definition 8 (β-Noisy Ciphertext) A ciphertext $\mathsf{ct} = (c_0,\ c_1)$ encrypting m is *β-noisy* if $\|c_0 - (p_2/p_1)c_1 \cdot s - \Delta m\|_\infty \le \beta$.

Lemma 3 Let ct_1, ct_2 be β_1- and β_2-noisy ciphertexts of m_1, m_2 under the extended key. Then:

1. $\mathsf{ct}_{\mathsf{add}}$ is $(\beta_1 + \beta_2 + t)$-noisy for $(m_1 + m_2)_t$.
2. $\mathsf{ct}_{\mathsf{mult}}$ is β_{mult}-noisy for $(m_1 \cdot m_2)_t$, where β_{mult} is given by the right-hand side of (7) with $\|y_j\|_\infty = \beta_j,\ j \in \{1, 2\}$.

Proof Follows directly from inequalities (4) and (7). □

Error Growth Summary (MKFHE-I)

Fresh ciphertext: $y_{\text{init}} = \frac{1}{2} + \frac{p_2}{p_1}\delta_{\mathcal{R}}\|s\|_\infty$.

After one multiplication: The noise $\approx b_1 y_{\text{init}} + b_2$, with

$$b_1 = (\delta_{\mathcal{R}}\|s\|_\infty + 4)\delta_{\mathcal{R}} t + \frac{\delta_{\mathcal{R}}}{2},$$

$$b_2 = r_t(p_2)t\delta_{\mathcal{R}}\left(\frac{\delta_{\mathcal{R}}\|s\|_\infty}{2} + 2.5\right) + \frac{\frac{p_2^2}{p_1^2}\delta_{\mathcal{R}}^2\|s\|_\infty^2 + \frac{p_2}{p_1}\delta_{\mathcal{R}}\|s\|_\infty + 1}{2} + lwk\delta_{\mathcal{R}}^2\|s\|_\infty\frac{p_2}{p_1} + \frac{klw\delta_{\mathcal{R}}}{2}.$$

After d multiplications: $b_1^d y_{\text{init}} + b_2\sum_{i=0}^{d-1} b_1^i \approx b_1^{d-1}(b_1 y_{\text{init}} + b_2)$.

3.3 Security Analysis

In this section, we analyze the security of the proposed MKFHE-I scheme with respect to the threat model defined in Section 2.2. Specifically, we prove that the scheme achieves IND-CPA security under the hardness of the Decisional Ring Learning with Rounding (RLWR) assumption, via a sequence of hybrid games. We begin with an auxiliary lemma used in the proof of the main theorem.

Lemma 4 Let λ be a security parameter, and let $n,\ p,\ q$ be positive integers with $p \mid q$. Let $f(x)$ be a monic polynomial of degree n with integer coefficients, $\mathcal{R}_q = \mathbb{Z}_q[x]/(f(x))$, and let $a \leftarrow \mathcal{R}_q$ be sampled uniformly at random. For each $i = 1, \ldots, k$, sample $s_i \leftarrow \chi$ independently and define

$$b_i = \lfloor a \cdot s_i \rceil_{p,q}, \qquad \bar{b} = \sum_{i\in[k]} b_i \pmod p.$$

Under the $\mathsf{RLWR}_{n,q,p}$ assumption, it holds that $(\bar{b},\ a) \approx_c \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q)$.

Proof We proceed via a standard hybrid argument. For $j \in \{0, 1, \ldots, k\}$, define the hybrid distribution $\mathbf{H}_j$ as follows:

- Sample $a \leftarrow \mathcal{R}_q$ uniformly at random.
- For each $i = 1, \ldots, j$, sample $u_i \leftarrow \mathcal{R}_p$ uniformly at random.
- For each $i = j+1, \ldots, k$, sample $s_i \leftarrow \chi$ and set $b_i = \lfloor a \cdot s_i \rceil_{p,q}$.
- Output the pair

$$\left(\sum_{i=1}^{j} u_i \; + \; \sum_{i=j+1}^{k} b_i \pmod{p},\; a \right).$$

Observe that $\mathbf{H}_0$ corresponds to the real distribution $(\sum_{i\in[k]} b_i \bmod p,\, a)$, and $\mathbf{H}_k$ corresponds to the idealized distribution $(\sum_{i\in[k]} u_i \bmod p,\, a)$.

Claim 1 $\mathbf{H}_k \equiv \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q)$.

Proof Let $\bar{u} = \sum_{i=1\in[k]} u_i \pmod{p}$. Fix arbitrary values $u_2 = c_2, \ldots, u_k = c_k \in \mathcal{R}_p$, and set $c = \sum_{i=2}^{k} c_i \pmod{p}$. The map

$$\phi : \mathcal{R}_p \to \mathcal{R}_p, \quad u_1 \mapsto u_1 + c \pmod{p}$$

is a bijection. Since $u_1 \leftarrow \mathcal{R}_p$ is uniform, it follows that $\phi(u_1) = u_1 + c$ is also uniform over $\mathcal{R}_p$. Therefore, for every $t \in \mathcal{R}_p$ and every fixing of $u_2, \ldots, u_k$,

$$\Pr\left[\bar{u} = t \;\middle|\; u_2 = c_2, \ldots, u_k = c_k\right] = \Pr[u_1 = t - c \pmod{p}] = \frac{1}{|\mathcal{R}_p|}.$$

By the law of total probability, $\Pr[\bar{u} = t] = 1/|\mathcal{R}_p|$ for every $t \in \mathcal{R}_p$, so $\bar{u}$ is uniformly distributed over $\mathcal{R}_p$. Moreover, since $u_1, \ldots, u_k$ are sampled independently of a, the random variable $\bar{u}$ is independent of a. Hence,

$$\mathbf{H}_k \equiv \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q). \qquad \square$$

Claim 2 For each $j \in \{1, \ldots, k\}$, we have $\mathbf{H}_{j-1} \approx_c \mathbf{H}_j$ under the $\mathsf{RLWR}_{n,q,p}$ assumption.

Proof Suppose, for contradiction, that there exists a PPT distinguisher $\mathcal{D}$ such that

$$\left|\Pr[\mathcal{D}(\mathbf{H}_{j-1}) = 1] - \Pr[\mathcal{D}(\mathbf{H}_j) = 1]\right| \geq \varepsilon$$

for some non-negligible function $\varepsilon = \varepsilon(\lambda)$. We construct a PPT adversary $\mathcal{B}$ that breaks the single-instance $\mathsf{RLWR}_{n,q,p}$ assumption.

Given a challenge $(b^*,\ a^*)$, where either $b^* = \lfloor a^* \cdot s \rceil_{p,q}$ for $s \leftarrow \chi$ or $b^* \leftarrow \mathcal{R}_p$ uniformly, the adversary $\mathcal{B}$ proceeds as follows:

- For $i = 1, \ldots, j-1$: sample $u_i \leftarrow \mathcal{R}_p$ uniformly.
- For $i = j+1, \ldots, k$: sample $s_i \leftarrow \chi$ and compute $b_i = \lfloor a^* \cdot s_i \rceil_{p,q}$.
- Compute $b' = \sum_{i=1}^{j-1} u_i + \sum_{i=j+1}^{k} b_i \pmod{p}$ and forward $(b' + b^* \bmod p,\, a^*)$ to $\mathcal{D}$, outputting whatever $\mathcal{D}$ outputs.

When $b^* = \lfloor a^* \cdot s \rceil_{p,q}$, the input to $\mathcal{D}$ is distributed identically to $\mathbf{H}_{j-1}$; when $b^* \leftarrow \mathcal{R}_p$, it is distributed identically to $\mathbf{H}_j$. Therefore, $\mathcal{B}$ distinguishes the two cases with advantage ε, contradicting the $\mathsf{RLWR}_{n,q,p}$ assumption. □

Conclusion. By Claim 1, $\mathbf{H}_k \equiv \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q)$. By Claim 2, each adjacent pair of hybrids is computationally indistinguishable. Applying the hybrid triangle inequality yields

$$\mathbf{H}_0 \approx_c \mathbf{H}_1 \approx_c \cdots \approx_c \mathbf{H}_k \equiv \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q),$$

and therefore $(\bar{b},\ a) \approx_c \mathcal{U}(\mathcal{R}_p \times \mathcal{R}_q)$. □

Theorem 3 *Let q, p_1, p_2 be positive integers satisfying $p_1 \mid q$ and $p_2 q = p_1^2$. Then the* RLWR*-based* $\mathsf{MKFHE}_{\mathsf{Basic}}$ *scheme described in Section 3 is* IND-CPA *secure (Definition 7), assuming the hardness of the decisional* RLWR_{n,q,p_1} *problem.*

Proof We show that no PPT adversary $\mathcal{A}$ can win the IND-CPA experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{MKFHE}_{\mathsf{Basic}}}(1^\lambda,\ \alpha)$ with non-negligible advantage. The proof proceeds through three hybrid games that transition from the real security experiment to an idealized game in which no adversary can gain a non-negligible advantage.

Hybrid $\mathbf{H}_0$ (Real game). This hybrid corresponds precisely to the real IND-CPA experiment between a challenger $\mathcal{C}h$ and the adversary $\mathcal{A}$.

- **Setup.** The challenger generates public parameters

  $$\mathsf{PP} = (n,\ q,\ p_1,\ p_2,\ t,\ w,\ k,\ a,\ \chi) \leftarrow \mathsf{MKFHE}_{\mathsf{Basic}}.\mathsf{Setup}(1^\lambda),$$

  and runs $\mathsf{MKFHE}_{\mathsf{Basic}}.\mathsf{KeyGen}(\mathsf{PP})$ to obtain a key pair $(\mathsf{pk}_j,\ \mathsf{sk}_j)$, where

  $$\mathsf{pk}_j = \left(b_j = \lfloor a \cdot s_j \rceil_{p_1,q},\ a\right) \in \mathcal{R}_{p_1} \times \mathcal{R}_q, \qquad \mathsf{sk}_j = s_j, \qquad s_j \leftarrow \chi.$$

  The challenger publishes pk_j.
- **Corruption and challenge submission.** The adversary $\mathcal{A}$ is permitted to observe the challenger's public key. We additionally assume $\mathcal{A}$ has obtained the public and secret key pairs of $k-1$ other parties through corruption. The adversary then selects two distinct messages $m_0,\ m_1 \in \mathcal{R}_t$ and submits them to the challenger along with the corresponding public keys.
- **Challenge.** The challenger first runs the key extension protocol to derive the extended public key

  $$\mathsf{epk} = (\bar{b},\ a) \leftarrow \mathsf{MKFHE}_{\mathsf{Basic}}.\mathsf{KeyExt}\left(\{\mathsf{pk}_i\}_{i\in[k]}\right),$$

  where $\bar{b} = \sum_{i\in[k]} b_i \bmod p_1$. It then samples a uniformly random bit $\alpha \leftarrow \{0,1\}$ and encrypts m_α under epk by computing

  $$\mathsf{ct}_\alpha = (c_0,\ c_1) = \left((\Delta m_\alpha + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1})_{p_2},\ \lfloor a \cdot r \rceil_{p_1,q}\right) \in \mathcal{R}_{p_2} \times \mathcal{R}_{p_1},$$

  where $r \leftarrow \chi$. The challenger sends ct_α to $\mathcal{A}$.
- **Guess.** The adversary outputs a guess bit α'. The experiment outputs 1 if $\alpha' = \alpha$.

Let $\mathcal{E}_0$ denote the event $\alpha' = \alpha$ in $\mathbf{H}_0$. The advantage of $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathsf{MKFHE}_{\mathsf{Basic}},\,\mathbf{H}_0} = \left|\Pr(\mathcal{E}_0) - \tfrac{1}{2}\right|. \tag{8}$$

Hybrid $\mathbf{H}_1$ (Random extended key). This hybrid is identical to $\mathbf{H}_0$, except that the challenger replaces the extended public key $\mathsf{epk} = (\bar{b},\ a)$ with a uniformly random pair

$$\mathsf{uk} = (b,\ a) \xleftarrow{\$} \mathcal{R}_{p_1} \times \mathcal{R}_q.$$

The challenge ciphertext is then computed honestly as

$$\mathsf{ct}_\alpha = ((\Delta m_\alpha + \lfloor b \cdot r \rceil_{p_2,p_1})_{p_2},\ \lfloor a \cdot r \rceil_{p_1,q}).$$

Let $\mathcal{E}_1$ denote the event $\alpha' = \alpha$ in $\mathbf{H}_1$, with advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathsf{MKFHE}_{\mathsf{Basic}},\,\mathbf{H}_1} = \left|\Pr(\mathcal{E}_1) - \tfrac{1}{2}\right|. \tag{9}$$

Indistinguishability of $\mathbf{H}_0$ *and* $\mathbf{H}_1$. By construction,

$$\bar{b} = \sum_{i\in[k]} \lfloor a \cdot s_i \rceil_{p_1,q} \pmod{p_1}.$$

By Lemma 4, the pair $(\bar{b},\ a)$ is computationally indistinguishable from a uniformly random pair $(b,\ a) \in \mathcal{R}_{p_1} \times \mathcal{R}_q$ under the RLWR_{n,q,p_1} assumption. Therefore, replacing $(\bar{b},\ a)$ with $(b,\ a)$ changes the adversary's view by at most a negligible amount. Let $\mathcal{B}$ be the PPT reduction that solves RLWR_{n,q,p_1} given $\mathcal{A}$'s advantage. Then

$$|\Pr(\mathcal{E}_0) - \Pr(\mathcal{E}_1)| \le \mathsf{Adv}^{\mathsf{dRLWR}_{n,q,p_1}}_{\mathcal{B}}(\lambda), \tag{10}$$

from which

$$\left|\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathbf{H}_0} - \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathbf{H}_1}\right| \le \mathsf{Adv}^{\mathsf{dRLWR}_{n,q,p_1}}_{\mathcal{B}}(\lambda). \tag{11}$$

Hybrid $\mathbf{H}_2$ (Uniformly random ciphertext). This hybrid is identical to $\mathbf{H}_1$, except that the challenger does not compute the ciphertext via the encryption algorithm. Instead, the challenge ciphertext is sampled uniformly at random from the ciphertext space $\mathcal{R}_{p_2} \times \mathcal{R}_{p_1}$ and sent directly to $\mathcal{A}$.

Let $\mathcal{E}_2$ denote the event $\alpha' = \alpha$ in $\mathbf{H}_2$, with advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathsf{MKFHE}_{\mathsf{Basic}},\,\mathbf{H}_2} = \left|\Pr(\mathcal{E}_2) - \tfrac{1}{2}\right|. \tag{12}$$

Indistinguishability of $\mathbf{H}_1$ *and* $\mathbf{H}_2$. In $\mathbf{H}_1$, the component $c_1 = \lfloor a \cdot r \rceil_{p_1,q}$ is a standard RLWR_{n,q,p_1} sample, which is computationally indistinguishable from a uniformly random element of $\mathcal{R}_{p_1}$. Conditioned on c_1 being uniform, $b \leftarrow \mathcal{R}_{p_1}$ is also uniform and independent; hence $c_0 = (\Delta m_\alpha + \lfloor b \cdot r \rceil_{p_2,p_1})_{p_2}$ is computationally indistinguishable from a uniformly random element of $\mathcal{R}_{p_2}$, since a fixed additive shift by Δm_α preserves uniformity. Therefore, the pair $(c_0,\ c_1)$ in $\mathbf{H}_1$ is computationally indistinguishable from a uniformly random element of $\mathcal{R}_{p_2} \times \mathcal{R}_{p_1}$, as in $\mathbf{H}_2$. Let $\mathcal{C}$ denote the corresponding PPT reduction. Then

$$|\Pr(\mathcal{E}_1) - \Pr(\mathcal{E}_2)| \le \mathsf{Adv}^{\mathsf{RLWR}_{n,q,p_1}}_{\mathcal{C}}(\lambda), \tag{13}$$

from which

$$\left|\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathbf{H}_1} - \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathbf{H}_2}\right| \le \mathsf{Adv}^{\mathsf{RLWR}_{n,q,p_1}}_{\mathcal{C}}(\lambda). \tag{14}$$

Final analysis. In $\mathbf{H}_2$, the challenge ciphertext is uniformly random and independent of the challenge bit α. Therefore,

$$\Pr[\mathcal{E}_2] = \frac{1}{2}, \qquad \text{and} \qquad \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathsf{MKFHE}_{\mathsf{Basic}},\,\mathbf{H}_2} = 0.$$

Combining inequalities (11) and (14) via the triangle inequality, we obtain

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\,\mathsf{MKFHE}_{\mathsf{Basic}},\,\mathbf{H}_0} \le \mathsf{Adv}^{\mathsf{dRLWR}_{n,q,p_1}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{RLWR}_{n,q,p_1}}_{\mathcal{C}}(\lambda) \le \mathsf{negl}(\lambda),$$

where the last inequality follows from the assumed hardness of RLWR_{n,q,p_1}. Hence, the $\mathsf{MKFHE}_{\mathsf{Basic}}$ scheme is $\mathsf{IND\text{-}CPA}$ secure. □

The following theorem establishes that the decryption protocol leaks no information about the secret key of any uncorrupted party, even when the adversary $\mathcal{A}$ corrupts up to $k-1$ parties and thereby obtains their secret keys, all observed ciphertexts, the corresponding partial decryption shares, and the final decrypted output.

Theorem 4 *The decryption protocol of* $\mathsf{MKFHE}_{\mathsf{Basic}}$ *is simulation-secure.*

Proof Let $\mathcal{A}$ be a PPT adversary that corrupts at most $k-1$ parties among the set $[k]$. Let $P \subseteq [k]$ denote the set of corrupted parties with $|P| \leq k-1$, and let $H = [k] \setminus P$ denote the set of honest parties. The adversary's view in the real execution is

$$\mathsf{View}_{\mathcal{A}}^{\text{real}} = \left(\mathsf{ct},\ \{s_i\}_{i\in P},\ \{\gamma_i\}_{i\in[k]},\ m\right),$$

where $\mathsf{ct} = (c_0, c_1)$, each partial decryption share is $\gamma_i = c_1 s_i + e_i^{sm}$ for $e_i^{sm} \leftarrow [-B_{sm}, B_{sm}]$, and the output message satisfies

$$m = \left\lfloor \frac{t}{p_2} \left(c_0 - \frac{p_2}{p_1} \sum_{i\in[k]} \gamma_i \right) \right\rceil_t .$$

We construct a simulator $\mathcal{S}$ that, given m, ct, and $\{s_i\}_{i\in P}$, produces a simulated view computationally indistinguishable from $\mathsf{View}_{\mathcal{A}}^{\text{real}}$. Fix an arbitrary honest party $r \in H$; the simulator $\mathcal{S}$ proceeds as follows.

- For each corrupted party $i \in P$, compute the partial decryption share exactly as in the real protocol:
$$\widetilde{\gamma}_i = c_1 \cdot s_i + \widetilde{e}_i^{sm}, \qquad \widetilde{e}_i^{sm} \leftarrow [-B_{sm}, B_{sm}].$$
- For each honest party $j \in H \setminus \{r\}$, set $\widetilde{\gamma}_j = \gamma_j$ (using the real share).
- To ensure consistency with the decryption output, define the remaining share $\widetilde{\gamma}_r$ so that the decryption equation is satisfied. Noting that
$$c_0 - \frac{p_2}{p_1} \sum_{i\in[k]} \gamma_i = \Delta m + e + e^{sm}, \tag{15}$$
where e denotes the ciphertext error and $e^{sm} = -\frac{p_2}{p_1} \sum_{i\in[k]} e_i^{sm}$ is the accumulated smudging noise, the simulator sets
$$\widetilde{\gamma}_r = \frac{p_1}{p_2} \left(c_0 - \frac{p_2}{p_1} \sum_{j\neq r} \widetilde{\gamma}_j - \Delta m - \widetilde{e}_r^{sm} \right), \tag{16}$$
where $\widetilde{e}_r^{sm}$ is sampled from $[-B_{sm}, B_{sm}]$ with the same distribution as e^{sm}.
- Output
$$\mathsf{View}_{\mathcal{A}}^{\text{sim}} = \left(\mathsf{ct},\ \{s_i\}_{i\in P},\ \{\widetilde{\gamma}_i\}_{i\in[k]},\ m\right).$$

Claim 3 $\mathsf{View}_{\mathcal{A}}^{\text{sim}} \approx_c \mathsf{View}_{\mathcal{A}}^{\text{real}}$.

Proof For every corrupted party $i \in P$, the simulated share $\widetilde{\gamma}_i$ is identically distributed to the real share γ_i by construction. For every honest party $j \in H \setminus \{r\}$, $\mathcal{S}$ uses the real share, so $\widetilde{\gamma}_j \equiv \gamma_j$. It remains to compare $\widetilde{\gamma}_r$ with γ_r.

From equation (15), the real share of party r satisfies

$$\gamma_r = \frac{p_1}{p_2}\left(c_0 - \frac{p_2}{p_1}\sum_{i\neq r}\gamma_i - \Delta m - e - e^{sm}\right). \tag{17}$$

Comparing (17) and (16), the difference between γ_r and $\widetilde{\gamma}_r$ is $\frac{p_1}{p_2}\cdot e$. Since $\|e\|_\infty / \|\widetilde{e}_r^{sm}\|_\infty \leq 2^{-\lambda}$, the smudging lemma (Lemma 2) implies

$$e + e^{sm} \;\approx_c\; \widetilde{e}_r^{sm},$$

and therefore $\gamma_r \approx_c \widetilde{\gamma}_r$. Combining the above, we conclude

$$\mathsf{View}_{\mathcal{A}}^{\mathrm{real}} \;\approx_c\; \mathsf{View}_{\mathcal{A}}^{\mathrm{sim}}. \qquad \square$$

Thus, no PPT adversary can distinguish the simulated execution from the real execution except with negligible probability, establishing simulation security of the decryption protocol. $\square$

4 Simplified-SHE and MKFHE-II

In this section, we independently propose a simplified-SHE scheme based on RLWR, then extend it to the multi-key setting as MKFHE-II.

4.1 Simplified-SHE Scheme

$$\text{Simplified-SHE} = \{\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}\}.$$

$\mathsf{PP} \leftarrow \mathsf{SetUp}(1^\lambda,\ 1^d)$: Same as SHE (Section 2.3); output $\mathsf{PP} = (n,\ q,\ p_1,\ p_2,\ t,\ w,\ \chi)$.

$(\mathsf{pk}, \mathsf{sk}, \mathsf{rlk}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$: Sample $s \leftarrow \chi$, $a \leftarrow \mathcal{R}_q$; and compute $b = \lfloor -a \cdot s \rceil_{p_1,q}$. Set $\mathsf{pk} = (b,\ a)$, $\mathsf{sk} = (1,\ s)$. For $i = 0, \ldots, l-1$ ($l = \lceil \log_w p_2 \rceil$), sample $a_i \leftarrow \mathcal{R}_{p_1}$ and set

$$\mathsf{rlk}[i] = \left(\lfloor -a_i \cdot s \rceil_{p_2,p_1} + w^i s^2 \bmod p_2,\ a_i\right).$$

$\mathsf{ct} \leftarrow \mathsf{Enc}(m,\ \mathsf{pk})$: Sample $r \leftarrow \chi$; output $(c_0,\ c_1) \in \mathcal{R}_{p_2}^2$ with $c_0 = \Delta m + \lfloor b \cdot r \rceil_{p_2,p_1} \bmod p_2$, $c_1 = \lfloor a \cdot r \rceil_{p_2,q}$.

$m' \leftarrow \mathsf{Dec}(\mathsf{sk},\ \mathsf{ct})$: $m' = \lfloor (t/p_2)(c_0 + c_1 \cdot s) \rceil_t$.

$\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{pk},\ \mathsf{rlk},\ (\mathsf{ct}_1,\ \mathsf{ct}_2),\ \mathcal{C})$: For two ciphertext $\mathsf{ct}_j = (c_{j0}, c_{j1})\ \ j \in \{1,2\}$.
Addition: $(c_{10} + c_{20},\ c_{11} + c_{21}) \bmod p_2$.
Multiplication: tensor product (all components in $\mathcal{R}_{p_2}$) followed by relinearization with $l = \lceil \log_w p_2 \rceil$.

Theorem 5 *Under* $p_1 \mid q$ *and* $p_2 q = p_1^2$, Simplified-SHE *is* IND-CPA *secure assuming hardness of* RLWR_{n,q,p_1}.

Proof Follows from the proof of [23, Theorem 2]. □

Correctness and error analysis for the scheme are given in Appendix A.

4.2 Construction II: MKFHE based on Simplified-SHE

$$\mathsf{MKFHE\text{-}II} = \{\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{KeyExt}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}\}.$$

SetUp, KeyGen, and KeyExt are identical to MKFHE-I (with $b_i = \lfloor -a \cdot s_i \rceil_{p_1,q}$). The relinearization key uses a sign-adjusted version of Algorithm 1 (see Algorithm 2 below).

Algorithm 2 Relinearization Key Generation Protocol (MKFHE-II)

Input: Public: $\mathbf{a}_t = (a_1, \ldots, a_l) \in \mathcal{R}_q^l,\ l = \lceil \log_w p_1 \rceil$
Input: Private: $\{s_i\}_{i\in[k]}$
Output: $\mathsf{rlk} = (\mathbf{h}'_0 + \mathbf{h}'_1,\ \mathbf{h}_1)$

— Round 1 —

1: **for** each party i **do**
2: Sample $u_i \leftarrow \chi$; compute and broadcast

$$(\mathbf{h}_{0,i},\ \mathbf{h}_{1,i}) = \left(\lfloor -u_i \mathbf{a}_t \rceil_{p_1,q} + \left\lfloor \frac{p_1}{p_2} s_i \mathbf{w} \right\rceil \bmod p_1,\ \lfloor s_i \mathbf{a}_t \rceil_{p_1,q} \right)$$

3: **end for**
4: Aggregate: $\mathbf{h}_0 = \sum_i \mathbf{h}_{0,i},\ \mathbf{h}_1 = \sum_i \mathbf{h}_{1,i}$

— Round 2 —

5: **for** each party i **do**
6: Compute and broadcast $(\mathbf{h}'_{0,i},\ \mathbf{h}'_{1,i}) = (\lfloor s_i \mathbf{h}_0 \rceil_{p_2,p_1},\ \lfloor (u_i - s_i)\mathbf{h}_1 \rceil_{p_2,p_1})$
7: **end for**
8: Aggregate and **return** $\mathsf{rlk} = (\mathbf{h}'_0 + \mathbf{h}'_1,\ \mathbf{h}_1)$

$\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{epk})$: $\mathsf{ct} = (c_0,\ c_1)$, where $c_0 = (\Delta m + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1})_{p_2}$, $c_1 = \lfloor a \cdot r \rceil_{p_2,q}$.

$\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{epk}, \mathsf{rlk}, (\mathsf{ct}_1, \mathsf{ct}_2), \mathcal{C})$: Same homomorphic operations as Simplified-SHE.

$m' \leftarrow \mathsf{Dec}(\{s_i\}_{i\in[k]}, \mathsf{ct})$: Multi-party protocol.

- *Partial decryption.* Each party i broadcasts $\gamma_i = c_1 \cdot s_i + e_i^{sm}$.
- *Final decryption.* $m' = \left\lfloor \frac{t}{p_2} \left[c_0 + \sum_{i\in[k]} \gamma_i \right]_{p_2} \right\rceil_t$.

Theorem 6 *Under $p_1 \mid q$ and $p_2 q = p_1^2$,* MKFHE-II *is* IND-CPA *secure and its decryption protocol is simulation-secure.*

Proof Follows from the proofs of Theorems 3 and 4 by the same hybrid argument (with $\bar{b} = \sum_{i\in[k]} \lfloor -a \cdot s_i \rceil_{p_1,q}$, which is uniform by Lemma 4). □

Correctness and error analysis for MKFHE-II are in Appendix B.

Error Growth Summary (MKFHE-II)

Fresh ciphertext:

$$y_{\text{init}} = \frac{1}{2} + \left(\frac{1}{2} + \frac{p_2}{2p_1}\right)\delta_{\mathcal{R}}\|s\|_\infty.$$

After one multiplication: The noise $\approx b_1 y_{\text{init}} + b_2$, with

$$\begin{aligned} b_1 =& (\delta_{\mathcal{R}}\|s\|_\infty + 4)\delta_{\mathcal{R}} t + \frac{\delta_{\mathcal{R}}}{2}, \\ b_2 =& r_t(p_2) t \delta_{\mathcal{R}} \left(\frac{\delta_{\mathcal{R}}\|s\|_\infty}{2} + 2.5\right) + \frac{\delta_{\mathcal{R}}^2\|s\|_\infty^2 + \delta_{\mathcal{R}}\|s\|_\infty + 1}{2} \\ &+ lwk\delta_{\mathcal{R}}^2\|s\|_\infty \frac{3p_2}{4p_1} + \frac{klw\delta_{\mathcal{R}}}{2} + \frac{\delta_{\mathcal{R}}\|s\|_\infty}{2}. \end{aligned}$$

After d multiplications:

$$b_1^d y_{\text{init}} + b_2 \sum_{i=0}^{d-1} b_1^i \approx b_1^{d-1}(b_1 y_{\text{init}} + b_2).$$

5 Parameter Selection

For the concrete parameter selection, we model the RLWR problem as an equivalent RLWE instance. Following this approach, we describe the parameter selection strategy for our first MKFHE construction, which is based on the SHE scheme of [23]. The parameter selection procedure for the second construction follows a similar methodology.

- **Key recovery attack:** For each party, we have

$$b_i = \lfloor a \cdot s_i \rceil_{p_1,q} = \left\lfloor \frac{p_1}{q}(a \cdot s_i) \right\rceil \bmod p_1 \ ,$$

  where $e_j \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ for $j \in \{1, \ldots, k\}$. Hence, the key recovery attack is modeled as a RLWE instance $(a \cdot s_i + \frac{q}{p_1} e \pmod q)$ over a ring of dimension n modulo q, with a uniformly ternary secret key and an error distribution of standard deviation $\sigma \approx 4.61$.

- **Message recovery attack:** For each party, the ciphertext is given by $\mathsf{ct} = (c_0, c_1) \in \mathcal{R}_{p_2} \times \mathcal{R}_{p_1}$, where $c_0 = \left[\Delta m + \lfloor \bar{b} \cdot r \rceil_{p_2,p_1}\right]_{p_2}$, and $c_1 = \lfloor a \cdot r \rceil_{p_1,q}$, where $\bar{b}$ refers to the extended public key. Using c_1, the message recovery attack can be modeled as a RLWE instance over a ring of dimension n modulo p_1, with a uniformly ternary ephemeral secret r and an error distribution of standard deviation $\sigma \approx 4.61$.

For each possible value of n (the ring dimension), we select parameters by maximizing the supported multiplicative depth d while choosing the smallest modulus q that guarantees correct decryption. For each candidate depth d, we compute the minimum modulus required to control error growth and retain only those parameter sets whose estimated security is at least 128 bits. Among all feasible choices, we select the largest d. We provide four sets of parameters, each corresponding to a different number of parties. If no feasible parameters are found for a given ring dimension n, we increase n and repeat the process. In this way, n is chosen as the smallest value for which a feasible parameter set exists.

Table 2: Parameters for Construction I based on SHE from [23] at 128-bit security.

(a) $k = 2$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	198	202	206	2
14	417	421	425	9
15	860	864	868	22
16	1735	1739	1743	46
17	3511	3515	3519	92

(b) $k = 4$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	201	205	209	2
14	427	431	435	9
15	849	853	858	21
16	1746	1750	1754	45
17	3491	3495	3499	89

(c) $k = 8$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	204	208	212	2
14	405	409	413	8
15	837	841	845	20
16	1719	1723	1727	43
17	3504	3508	3512	87

(d) $k = 128$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	182	186	190	1
14	404	408	412	7
15	844	848	852	18
16	1733	1737	1741	39
17	3515	3519	3523	79

To evaluate security, we rely on the lattice estimator [29][1]. In particular, we consider the `dual_hybrid`, `primal_usvp`, and `primal_bdd` attacks, and select parameters that achieve at least a 128-bit security level. The parameter sets selected for MKFHE-I and II at the 128-bit security level for varying circuit depths are summarized in Tables 2 and 3, respectively.

[1] https://github.com/malb/lattice-estimator, commit e434f89

Table 3: Parameters for Construction II based on Simplified-SHE (from this work) at 128-bit security.

(a) $k = 2$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	201	205	209	2
14	420	424	428	9
15	863	867	871	22
16	1738	1742	1746	46
17	3514	3518	3522	92

(b) $k = 4$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	204	208	212	2
14	430	434	438	9
15	852	856	860	21
16	1714	1718	1722	44
17	3494	3498	3502	89

(c) $k = 8$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	207	211	215	2
14	408	412	416	8
15	840	844	848	20
16	1722	1726	1730	43
17	3507	3511	3515	87

(d) $k = 128$

$\log_2 n$	$\log_2 p_2$	$\log_2 p_1$	$\log_2 q$	d
13	185	189	193	1
14	407	411	415	7
15	847	851	855	18
16	1736	1740	1744	39
17	3518	3522	3526	79

6 Comparison and Discussion

Since our implementation is intended as a proof of concept, comparing its running time with that of highly optimized Ring-LWE-based implementations such as [20] would not constitute a fair comparison. Therefore, following the methodology of [23], we focus on comparing communication and storage costs. These metrics directly influence the efficiency of homomorphic evaluation and distributed decryption and thus provide a meaningful measure of the practical performance of the proposed schemes.

We compare the sizes of the public key pk, relinearization key rlk, extended public key epk, ciphertext ct, and partial decryption share γ_i of our two RLWR-based MKFHE constructions with the corresponding RLWE-based construction of [20] under comparable security levels and homomorphic evaluation depths. The asymptotic comparison is summarized in Table 1.

For MKFHE-I, we denote the moduli by $(q,\ p_1,\ p_2)$, where $p_2 q = p_1^2$. For MKFHE-II, the moduli are denoted by $(q',\ p_1',\ p_2')$, where $p_2' q' = p_1'^2$. The modulus of the RLWE scheme in [20] is denoted by q''. Based on the error-growth analyses of the three schemes, we determine parameters that provide comparable security and support the evaluation of a circuit $\mathcal{C}$ of a given multiplicative depth. In particular, our analysis indicates that

$$q \approx q'', \qquad q' \approx 8q.$$

Overall, both proposed schemes achieve lower storage and communication costs than the corresponding scheme in [20].

The improved compactness of our proposed schemes stems from their use of smaller moduli. For the public key pk and the extended public key epk, the RLWE-MKFHE

Table 4: Comparison of storage and communication costs for different numbers of parties. Public-key, ciphertext, and decryption-share sizes are reported in kB, while relinearization-key sizes are reported in GB.

k	Depth	$\|\mathsf{pk}\|$			$\|\mathsf{ct}\|$			$\|\gamma_i\|$			$\|\mathsf{rlk}\|$		
		RLWR_1	RLWR_2	RLWE	RLWR_1	RLWR_2	RLWE	RLWR_1	RLWR_2	RLWE	RLWR_1	RLWR_2	RLWE
2	2	**408**	414	412	**400**	406	412	202	**201**	206	**0.079**	0.079	0.081
	9	**1692**	1704	1700	**1676**	1688	1700	842	**840**	850	**0.679**	0.683	0.689
	22	**6928**	6952	6944	**6896**	6920	6944	3456	**3452**	3472	**5.708**	5.722	5.748
	46	**27856**	27904	27888	**27792**	27840	27888	13912	**13904**	13944	**46.197**	46.250	46.356
4	2	**414**	420	418	**406**	412	418	205	**204**	209	**0.081**	0.082	0.083
	8/9†	1732	1744	1616	1716	1728	1616	862	860	808	0.712	0.715	0.623
	21	**6840**	6864	6856	**6808**	6832	6856	3412	**3408**	3428	**5.564**	5.577	5.603
	44/45‡	28032	27520	27504	27968	27456	27504	14000	13712	13752	46.787	44.985	45.090
8	2	**420**	426	424	**412**	418	424	208	**207**	212	**0.083**	0.084	0.086
	8	**1644**	1656	1652	**1628**	1640	1652	818	**816**	826	**0.641**	0.644	0.651
	20	**6744**	6768	6760	**6712**	6736	6760	3364	**3360**	3380	**5.408**	5.421	5.447
	43	**27600**	27648	27632	**27536**	27584	27632	13784	**13776**	13816	**45.347**	45.400	45.505
128	1	**376**	382	380	**368**	374	380	186	**185**	190	**0.067**	0.067	0.069
	7	**1640**	1652	1648	**1624**	1636	1648	816	**814**	824	**0.638**	0.641	0.647
	18	**6800**	6824	6816	**6768**	6792	6816	3392	**3388**	3408	**5.498**	5.512	5.538
	39	**27824**	27872	27856	**27760**	27808	27856	13896	**13888**	13928	**46.091**	46.145	46.252

† RLWR_1 and RLWR_2 correspond to multiplicative depth 9, whereas RLWE corresponds to multiplicative depth 8. Therefore, the reported sizes correspond to different circuit depths and are not directly comparable.

‡ RLWR_1 corresponds to multiplicative depth 45, whereas RLWR_2 and RLWE correspond to multiplicative depth 44. Therefore, the reported sizes correspond to different circuit depths and are not directly comparable.

consists of two ring elements in $\mathcal{R}_{q''}$, whereas MKFHE-I uses one element from $\mathcal{R}_q$ and other from $\mathcal{R}_{p_1}$, and MKFHE-II uses one element from $\mathcal{R}_{q'}$ and other from $\mathcal{R}_{p_1'}$. Since $p_1,\ p_1' < q''$, MKFHE-I achieves smaller public keys than the MKFHE of [20], whereas MKFHE-II yields public keys of comparable size.

A similar observation holds for ciphertexts. In the RLWE-MKFHE, ciphertexts belong to $\mathcal{R}_{q''}^2$. By comparison, ciphertexts in MKFHE-I belong to $\mathcal{R}_{p_1} \times \mathcal{R}_{p_2}$, while those in MKFHE-II belong to $\mathcal{R}_{p_1'} \times \mathcal{R}_{p_2'}$. As these moduli are smaller than q'', both proposed constructions achieve more compact ciphertexts.

The communication cost during distributed decryption is likewise reduced. In the RLWE-MKFHE scheme, each partial decryption share is an element of $\mathcal{R}_{q''}$. MKFHE-I instead produces shares in $\mathcal{R}_{p_1}$, while MKFHE-II produces shares in $\mathcal{R}_{p_2'}$. The smaller moduli, therefore, translate directly into lower communication costs during distributed decryption.

For the relinearization key, the RLWE-MKFHE consists of $2l''$ polynomials in $\mathcal{R}_{q''}$, where $l'' = \lceil \log_w(q'') \rceil$. MKFHE-I employs l polynomials in $\mathcal{R}_{p_1}$ and l polynomials in $\mathcal{R}_{p_2}$, where $l = \lceil \log_w(q) \rceil$. Likewise, MKFHE-II employs l' polynomials in $\mathcal{R}_{p_1'}$ and l' polynomials in $\mathcal{R}_{p_2'}$, where $l' = \lceil \log_w(q') \rceil$. Since the constituent moduli are smaller, the resulting relinearization keys are also more compact.

Table 4 reports the concrete storage and communication costs of the three constructions for $k = 2, 4, 8, \text{and}, 128$ parties and decomposition base $w = 2$ across different multiplicative depths. Since the extended public key has the same size as the public key in all constructions, it is omitted from the comparison.

Overall, the results indicate that both proposed MKFHE consistently outperform the RLWE-based MKFHE in terms of storage and communication efficiency. MKFHE-I achieves the smallest public keys, ciphertexts, and relinearization keys, whereas MKFHE-II attains the lowest distributed-decryption communication cost. Since every participating party must transmit a decryption share for each decryption request, the communication savings of MKFHE-II scale naturally with both the number of parties and the number of decryptions. Consequently, MKFHE-II is particularly attractive in communication-intensive settings. More broadly, both RLWR-based proposed MKFHE constructions offer a more favorable efficiency trade-off than the corresponding RLWE-based construction while maintaining comparable security and functionality.

7 Dynamic Setting

In the context of MKFHE, *dynamicity* refers to the ability of the scheme to permit parties to join or leave an ongoing computation without requiring existing ciphertexts to be re-encrypted or the protocol to be restarted from scratch. This property is particularly important in collaborative computation settings, where the set of participating users may not remain fixed during the entire computation.

The constructions MKFHE-I and MKFHE-II as described above assume a static participant set. More precisely, the set of parties must be fixed before the execution of KeyExt algorithm and ciphertexts are generated. In particular, the extended public key $\mathsf{epk} = (\bar{b}, a)$ with $\bar{b} = \sum_{i=1 \in [k]} b_i$. If a new party joins or an existing party leaves, the value of $\bar{b}$ changes, and ciphertexts encrypted under the previous extended key are no longer directly compatible with the modified key. Therefore, the static version of the scheme does not immediately support dynamic participation.

We address this limitation below by introducing a dynamic variant of MKFHE-I.

7.1 Dynamic Extension: MKFHE-Dyn

The dynamic scheme is defined as
MKFHE-Dyn = {SetUp, KeyGen, KeyExt, KeyUpd, CiphUpd, KeyRef, CiphRef, Enc, Eval, Dec}.
The algorithms SetUp, KeyGen, KeyExt, Enc, Eval, and Dec are the same as in MKFHE-I. The new algorithms KeyUpd and CiphUpd handle the arrival of a new party, whereas KeyRef and CiphRef handle the departure of an existing party.

Party arrival.

Assume that the current participant set is $[k]$, and the current extended public key is $\mathsf{epk} = (\bar{b},\ a)$, $\bar{b} = \sum_{i \in [k]} b_i$. A new party joins with an independently generated key pair $(\mathsf{pk}_{k+1},\ \mathsf{sk}_{k+1})$, where $\mathsf{pk}_{k+1} = (b_{k+1},\ a)$.

KeyUpd: The extended key is updated *without interaction among existing parties* as

$$\mathsf{epk}' = (\bar{b}' = \bar{b} + b_{k+1},\ a).$$

Furthermore, it runs the relinearization algorithm to generate a new relinearization key corresponding to a new set of parties.
CiphUpd: Let $\mathsf{ct} = (c_0,\ c_1)$ be a ciphertext encrypted under epk. To make ct compatible with epk', the joining party updates the c_0 as $c_0' = c_0 + \lfloor c_1 \cdot s_{k+1} \rceil_{p_2,p_1} + e_{sm}$. The $e_s m$ is the smudging error to safeguard the secret s_{k+1}. The pair $(c_0',\ c_1)$ is the updated ciphertext encrypted under epk'.

This update is not fully public, since it depends on the new party's secret key s_{k+1}. However, it is non-interactive with respect to the existing parties: only the arriving party needs to perform this protocol.

Party departure.

Now suppose that the j-th party leaves the computation. The goal is to remove the share of this party from both the extended public key and the existing ciphertexts.

KeyRef: This updates the extended key by removing the departing party's contribution:

$$\mathsf{epk}'' = (\bar{b}'' = \bar{b} - b_j,\ a).$$

Furthermore, it runs the relinearization algorithm to generate a new relinearization key corresponding to a new set of parties.
CiphRef: Let $\mathsf{ct} = (c_0,\ c_1)$ be an existing ciphertext encrypted under epk. To refresh ct so that it is decryptable under epk''. The departing party needs to share the partial decryption of the ciphertexts of all other parties, as each ciphertext has a public key share of each participant party.

The departing party refreshes the ciphertext ct as $\mathsf{ct}'' = (c_0 - \lfloor c_1 \cdot s_j \rceil_{p_2,p_1} + e_{sm},\ c_1)$, where e_{sm} is the smudging noise, added to safeguard the secret of the departing party.

Dynamicity Comparison in Ciphertext-Extension and Key-Extension based* MKFHE *schemes

As discussed, in ciphertext extension-based MKFHE schemes, each participant independently generates their ciphertexts, which are further extended to a multi-key ciphertext associated with the set of parties involved in the computation. This approach naturally supports the inclusion of newly arriving ciphertexts, since the key set attached to an evaluated ciphertext can be enlarged by applying an extension. However, the ciphertext size, evaluation key size, and the cost of homomorphic operations grow with the number of participating parties [30]. Thus, joining a new party is relatively natural from a functional point of view, but it increases the ciphertext's dimension and the complexity of subsequent evaluation. Moreover, removing a party is non-trivial, because a ciphertext already associated with that party's public key generally requires that party's secret-key share for decryption.

Instead, key-extension-based MKFHE schemes provide a notable efficiency advantage: the ciphertext dimension remains independent of the number of participating parties, and the evaluation procedure remains closer to the underlying single-key FHE schemes. Nevertheless, dynamicity becomes more delicate because the extended public key is tied to a fixed participant set. If a new party joins this set, then the old ciphertexts must be updated to the new extended key. This update is computationally

expensive because it requires updating all stored ciphertexts to ensure they remain usable after the new party joins. The extended key and evaluation key need to be updated. In addition, the ciphertext update introduces an extra error term, known as smudging noise, to hide the secret. Similar operations need to be done when a party leaves the protocol.

From the above discussion, we can conclude that ciphertext-extension and key-extension-based MKFHE schemes exhibit different efficiency-dynamicity trade-offs. Ciphertext-extension-based constructions provide a more direct mechanism for handling newly added parties. However, this flexibility is achieved at the cost of ciphertext expansion, larger evaluation key, and higher homomorphic evaluation complexity as the number of parties increases. On the other hand, key-extension-based constructions remain compact, and the evaluation procedure is closer to that of the underlying single-key FHE scheme. Nevertheless, this advantage comes with a reduced form of dynamicity: whenever a party joins or leaves the protocol, the extended public key changes. The existing ciphertexts and evaluation keys need to be updated accordingly. Such updates may introduce additional smudging noise and increase the computational burden. Therefore, ciphertext-extension-based MKFHE is more suitable when frequent changes in the participant set are expected, whereas key-extension-based MKFHE is more efficient when the participant set is mostly fixed, and compact ciphertext size is a priority.

8 Conclusion

In this work, we proposed ulti-key fully homomorphic encryption (MKFHE) schemes based on the Ring Learning with Rounding (RLWR). We presented two constructions: the first is based on the SHE scheme of Bolboceanu *et al.* [23], while the second is based on a Simplified-SHE construction proposed in this work. We conducted a careful error analysis and showed that both constructions achieve greater compactness than comparable RLWE-based schemes in the multi-key FHE setting. We further instantiated the schemes with parameter sets supporting circuits of varying multiplicative depths and provided a proof-of-concept implementation to validate our analysis. As future work, we plan to investigate threshold-decryption protocols for the proposed constructions and explore their integration into more general MPC frameworks. We also intend to further optimize our implementation and evaluate the performance of the schemes in large-scale settings.

Acknowledgement

This work was partly supported by the Information Security Education and Awareness (ISEA) Phase-3 project funded by MeitY, Government of India (Grant No. L-14017/1/2022-HRD, dated 01.07.2024).

Appendix A Simplified-SHE: Correctness and Error Analysis

A.1 Correctness of Decryption

Case 1: Fresh Ciphertext

Let $\mathsf{ct} = (c_0,\ c_1)$ encrypt m with $b = \lfloor -a \cdot s \rceil_{p_1,q} = (-p_1/q)a \cdot s + e_1$. Expanding:

$$\begin{aligned} c_0 + c_1 \cdot s &= \Delta m + \frac{p_2}{p_1} b \cdot r + e_0 + \left(\frac{p_2}{q} a \cdot r + e_2\right) \cdot s + p_2 h \\ &= \Delta m - \frac{p_2}{q} a \cdot s \cdot r + \frac{p_2}{p_1} e_1 \cdot r + e_0 + \frac{p_2}{q} a \cdot r \cdot s + e_2 \cdot s + p_2 h \\ &= \Delta m + v + p_2 h, \end{aligned}$$

where $v = e_0 + \frac{p_2}{p_1} e_1 \cdot r + e_2 \cdot s$ (the $a \cdot s \cdot r$ terms cancel), with $\|v\|_\infty \leq \frac{\delta_\mathcal{R} \|s\|_\infty}{2}\left(1 + \frac{p_2}{p_1}\right) + \frac{1}{2}$. Decryption is correct if $\left\| \frac{t}{p_2} v - \frac{r_t(p_2)}{p_2} m \right\|_\infty < \frac{1}{2}$.

Case 2: Evaluated Ciphertext

Let $c_{j0} + c_{j1} \cdot s = \Delta m_j + v_j + p_2 h_j$ for $j = 1, 2$.

Addition. $\mathsf{ct}_{\mathsf{add}} \cdot \mathsf{sk} = \Delta(m_1 + m_2)_t + v_{\mathsf{add}} + p_2(\cdots)$, where $v_{\mathsf{add}} = -r_t(p_2) r_3 + v_1 + v_2$, giving

$$\|v_{\mathsf{add}}\|_\infty \leq t + \|v_1\|_\infty + \|v_2\|_\infty. \tag{A1}$$

Multiplication. After the tensor product and relinearisation (all in $\mathcal{R}_{p_2}$), expanding $\mathsf{ct}_{\mathsf{mult}} \cdot \mathsf{sk}$ and substituting the relinearisation key yields $c_0' + c_1' \cdot s = \Delta(m_1 m_2)_t + v_{\mathsf{mult}} + p_2(\cdots)$, where

$$\begin{aligned} \|v_{\mathsf{mult}}\|_\infty \leq\ & \delta_\mathcal{R} t \left(\frac{\delta_\mathcal{R} \|s\|_\infty}{2} + 2\right)(\|v_1\|_\infty + \|v_2\|_\infty) + \frac{\delta_\mathcal{R}}{2} \min\{\|v_1\|_\infty, \|v_2\|_\infty\} \\ & + r_t(p_2) \delta_\mathcal{R} t \left(\frac{\delta_\mathcal{R} \|s\|_\infty}{2} + 2.5\right) + \frac{1 + \delta_\mathcal{R} \|s\|_\infty + \delta_\mathcal{R}^2 \|s\|_\infty^2}{2} + \frac{\delta_\mathcal{R} l w}{4} + \frac{\delta_\mathcal{R} \|s\|_\infty}{2}. \end{aligned} \tag{A2}$$

Lemma 5 Let $\mathsf{ct}_1, \mathsf{ct}_2$ be β_1- and β_2-noisy ciphertexts. Then $\mathsf{ct}_{\mathsf{add}}$ is $(\beta_1 + \beta_2 + t)$-noisy for $(m_1 + m_2)_t$, and $\mathsf{ct}_{\mathsf{mult}}$ is β_{mult}-noisy for $(m_1 \cdot m_2)_t$ (bound from (A2)).

Proof Follows from (A1) and (A2). □

Appendix B MKFHE-II: Correctness and Error Analysis

B.1 Correctness of Decryption

Let $s = \sum_{i\in[k]} s_i$ and $\gamma_i = c_1 \cdot s_i + e_i^{sm}$ denote each partial share.

Case 1: Fresh Ciphertext

Expanding $c_0 + \sum_{i\in[k]} \gamma_i$ using $\bar{b} = \sum_i b_i = \sum_{i\in[k]}(-(p_1/q)a \cdot s_i + e_i)$:

$$c_0 + \sum_{i\in[k]} \gamma_i = \Delta m + y + p_2 h + \sum_i e_i^{sm},$$

where $y = \frac{p_2}{p_1}\sum_i e_i \cdot r + e_0 + e' \cdot s$ with $\|y\|_\infty < \frac{\delta_\mathcal{R}\|s\|_\infty}{2}\left(\frac{p_2}{p_1} + 1\right) + \frac{1}{2}$.
Decryption is correct if $\left\| -\frac{r_t(p_2)}{p_2} m + \frac{t}{p_2} y + \frac{t}{p_2}\sum_i e_i^{sm} \right\|_\infty < \frac{1}{2}$.

Case 2: Evaluated Ciphertext

Let $c_{j0} + c_{j1} \cdot s = \Delta m_j + y_j + p_2 h_j$.

Addition. $\|y_{\mathsf{add}}\|_\infty \le t + \|y_1\|_\infty + \|y_2\|_\infty$ (same argument as Appendix A).

$$\|y_{\mathsf{add}}\|_\infty < t + \|y_1\|_\infty + \|y_2\|_\infty. \tag{B3}$$

Multiplication. After relinearisation and collecting terms, $c_0' + \sum_{i\in[k]} \gamma_i = \Delta(m_1 m_2)_t + y_{\mathsf{mult}} + p_2(\cdots)$, where r_a is bounded as before and the relinearisation error is now

$$\|e''\|_\infty \le \frac{3p_2}{4p_1} klw\delta_\mathcal{R}^2 \|s\|_\infty + \frac{1}{2} klw\delta_\mathcal{R} + \frac{1}{2}\delta_\mathcal{R}\|s\|_\infty,$$

giving

$$\begin{aligned}\|y_{\mathsf{mult}}\|_\infty \le{}& \delta_\mathcal{R} t\left(\frac{\delta_\mathcal{R}\|s\|_\infty}{2} + 2\right)(\|y_1\|_\infty + \|y_2\|_\infty) + \frac{\delta_\mathcal{R}}{2}\min\{\|y_1\|_\infty, \|y_2\|_\infty\} \\ &+ r_t(p_2)\delta_\mathcal{R} t\left(\frac{\delta_\mathcal{R}\|s\|_\infty}{2} + 2.5\right) + \frac{1 + \delta_\mathcal{R}\|s\|_\infty + \delta_\mathcal{R}^2\|s\|_\infty^2}{2} + lwk\delta_\mathcal{R}^2\|s\|_\infty\frac{3p_2}{4p_1} \\ &+ \frac{klw\delta_\mathcal{R}}{2} + \frac{\delta_\mathcal{R}\|s\|_\infty}{2}.\end{aligned} \tag{B4}$$

Lemma 6 Let $\mathsf{ct}_1, \mathsf{ct}_2$ be β_1- and β_2-noisy ciphertexts under the extended key. Then $\mathsf{ct}_{\mathsf{add}}$ is $(\beta_1 + \beta_2 + t)$-noisy for $(m_1 + m_2)_t$, and $\mathsf{ct}_{\mathsf{mult}}$ is β_{mult}-noisy for $(m_1 \cdot m_2)_t$ (bound from (B4)).

Proof Follows from (B3) and (B4). □

Appendix C RLWE Parameter sets

Considering the error growth discussion as per [20, 24], we can summarize the error growth for the MKFHE based on RLWE as the following

Reference: Error Growth (RLWE-based MKFHE)

Fresh ciphertext:

$$y_{\text{init}} = 2\delta_{\mathcal{R}}\|s\|_\infty\|e\|_\infty + \|e\|_\infty, \qquad \|e\|_\infty \leq \beta_\chi.$$

After one multiplication: The noise $\approx b_1 y_{\text{init}} + b_2$, where b_1 is same as previous schemes and

$$b_2 = r_t(p_2)t\delta_{\mathcal{R}}\left(\frac{\delta_{\mathcal{R}}\|s\|_\infty}{2} + 2.5\right) + \frac{\delta_{\mathcal{R}}^2\|s\|_\infty^2 + \delta_{\mathcal{R}}\|s\|_\infty + 1}{2} + 3lwk\delta_{\mathcal{R}}^2\|s\|_\infty\|e\|_\infty + 2lwk\delta_{\mathcal{R}}\|e\|_\infty\|s\|_\infty.$$

Therefore, considering equivalent parameters for the MKFHE based on RLWR in Section 5, Table C1 shows the parameter sets for different parties and circuit depths for the LWE-based construction.

Table C1: Parameters for the RLWE-based MKFHE scheme [20] at 128-bit security (reference construction).

(a) $k = 2$

$\log_2 n$	$\log_2 q$	d
13	206	2
14	425	9
15	868	22
16	1743	46
17	3519	92

(b) $k = 4$

$\log_2 n$	$\log_2 q$	d
13	209	2
14	404	8
15	857	21
16	1719	44
17	3499	89

(c) $k = 8$

$\log_2 n$	$\log_2 q$	d
13	212	2
14	413	8
15	845	20
16	1727	43
17	3512	87

(d) $k = 128$

$\log_2 n$	$\log_2 q$	d
13	190	1
14	412	7
15	852	18
16	1741	39
17	3523	79

References

[1] Gentry, C.: Fully homomorphic encryption using ideal lattices. PhD thesis, Stanford University, Stanford, CA, USA (2009). https://doi.org/10.1145/1536414.1536440

[2] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS), pp. 309–325 (2012). https://doi.org/10.1145/2090236.2090262

[3] Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. In: IACR Cryptology ePrint Archive, vol. 2012, p. 144 (2012)

[4] Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology – CRYPTO 2013. LNCS, vol. 8042, pp. 75–92 (2013). https://doi.org/10.1007/978-3-642-40041-4_5 . Springer

[5] Ducas, L., Micciancio, D.: FHEW: Bootstrapping homomorphic encryption in less than a second. In: Advances in Cryptology – EUROCRYPT 2015. Lecture Notes in Computer Science, vol. 9056, pp. 617–640 (2015). https://doi.org/10.1007/978-3-662-46800-5_24 . Springer

[6] Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology – ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437 (2017). https://doi.org/10.1007/978-3-319-70694-8_15 . Springer

[7] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Tfhe: Fast fully homomorphic encryption over the torus. Journal of Cryptology **33**(1), 34–91 (2020) https://doi.org/10.1007/s00145-019-09319-x

[8] López-Alt, A., Tromer, E., Vaikuntanathan, V.: Multikey fully homomorphic encryption and applications. SIAM Journal on Computing **46**(6), 1827–1892 (2017) https://doi.org/10.1137/14100124X

[9] Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. LNCS, vol. 7237, pp. 719–737 (2012). https://doi.org/10.1007/978-3-642-29011-4_42 . Springer

[10] Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference. LNCS, vol. 8042, pp. 57–74 (2013). https://doi.org/10.1007/978-3-642-40041-4_4 . Springer

[11] Bogdanov, A., Guo, S., Masny, D., Richelson, S., Rosen, A.: On the hardness of learning with rounding over small modulus. In: Theory of Cryptography - 13th International Conference, TCC. LNCS, vol. 9562, pp. 209–224 (2016). https://doi.org/10.1007/978-3-662-49096-9_9 . Springer

[12] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. STOC '05, pp. 84–93. ACM, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060603

[13] Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard: Cut off the tail! practical post-quantum public-key encryption from lwe and lwr. In: Security and Cryptography for Networks. Lecture Notes in Computer Science, vol. 11099, pp. 160–177 (2018). https://doi.org/10.1007/978-3-319-98113-0_9

[14] Bhattacharya, S., Garcia-Morchon, O., Laarhoven, T., Rietman, R., Saarinen, M.-J.O., Tolhuizen, L., Zhang, Z.: Round5: Compact and fast post-quantum public-key encryption. In: Post-Quantum Cryptography. Lecture Notes in Computer Science, vol. 11505, pp. 83–102 (2019). https://doi.org/10.1007/978-3-030-25510-7_5 . Springer

[15] Karmakar, A., Roy, S.S., Tunstall, M., Vercauteren, F., Verbauwhede, I., Brenner, G., Chen, Z., Clercq, R., Dhooghe, E., Fischer, V., Khalil, R.M., Lyubashevsky, V., Mehrshahi, A.: SABER: Module-LWR based KEM (Round 3 Submission) (2021). https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[16] Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled fhe from learning with errors. In: Advances in Cryptology – CRYPTO 2015, pp. 630–656 (2015). https://doi.org/10.1007/978-3-662-48000-7_31 . Springer Berlin Heidelberg

[17] Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key fhe. In: Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part II 35, pp. 735–763 (2016). https://doi.org/10.1007/978-3-662-49896-5_25 . Springer

[18] Peikert, C., Shiehian, S.: Multi-key fhe from lwe, revisited. In: Theory of Cryptography, pp. 217–238 (2016). https://doi.org/10.1007/978-3-662-53644-5_9 . Springer

[19] Brakerski, Z., Perlman, R.: Lattice-based fully dynamic multi-key fhe with short ciphertexts. In: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I 36, pp. 190–213 (2016). https://doi.org/10.1007/978-3-662-53018-4_8 . Springer

[20] Mouchet, C., Troncoso-Pastoriza, J., 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

[21] Goyal, M., Kar Gangopadhyay, A.: Key extension: Multi-key fhe utilizing lwr. In: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security, pp. 58–71 (2025). https://doi.org/10.1145/3708821.3736209

[22] Chen, L., Zhang, Z., Wang, X.: Batched multi-hop multi-key fhe from ring-lwe with compact ciphertext extension. In: Theory of Cryptography Conference, pp. 577–602 (2017). https://doi.org/10.1007/978-3-319-70503-3_20 . Springer

[23] Bolboceanu, M., Costache, A., Hales, E., Player, R., Rosca, M., Titiu, R.: Designs for practical SHE schemes based on ring-lwr. IACR Commun. Cryptol. **2**(1), 21 (2025) https://doi.org/10.62056/AV7TUDY6B

[24] Zucca, V.: Towards efficient arithmetic for ring-lwe based homomorphic encryption. PhD thesis, Sorbonne Université (2018). https://theses.hal.science/tel-02108993/file/2018SORUS080.pdf

[25] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 1–23 (2010). https://doi.org/10.1007/978-3-642-13190-5_1 . Springer

[26] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Annual International Cryptology Conference, pp. 595–618 (2009). https://doi.org/10.1007/978-3-642-03356-8_35 . Springer

[27] Langlois, A., Stehlé, D.: Hardness of decision (r) lwe for any modulus. Cryptology ePrint Archive (2012). https://eprint.iacr.org/2012/091

[28] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 483–501 (2012). https://doi.org/10.1007/978-3-642-29011-4_29 . Springer

[29] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015) https://doi.org/10.1515/jmc-2015-0016

[30] Jiang, B.: Multi-key fhe without ciphertext-expansion in two-server model. Frontiers of Computer Science **16**(1), 161809 (2022) https://doi.org/10.1007/s11704-021-0479-5 . Springer