Low Communication Threshold FHE from Standard (Module-)LWE

Hiroki Okada[1,2] and Tsuyoshi Takagi[2]

[1] KDDI Research, Inc., Japan
[2] The University of Tokyo, Japan

March 4, 2025

Abstract. Threshold fully homomorphic encryption (ThFHE) is an extension of FHE that can be applied to multiparty computation (MPC) with low round complexity. Recently, Passelègue and Stehlé (Asiacrypt 2024) presented a simulation-secure ThFHE scheme with polynomially small decryption shares from "yet another" learning with errors assumption (LWE), in which the norm of the secret key is leaked to the adversary. While "yet another" LWE is reduced from standard LWE, its module variant, "yet another" module-LWE (MLWE), lacks a known reduction from standard MLWE. Because of this, it is left as an open question to extend their scheme to the MLWE-based construction.

In this paper, we address this open problem: we propose a simulation-secure ThFHE scheme with polynomially small decryption shares whose security is (directly) reduced from standard LWE/MLWE. Our core technique, which we call "noise padding", eliminates the need of "yet another" assumptions: we distribute shares of a small error and use them to adjust the distribution of decryption noise so that no information about the secret key is leaked. As side benefits of our construction, our ThFHE efficiently realizes arbitrary T-out-of-N threshold decryption via simple Shamir secret sharing instead of $\{0,1\}$-linear secret sharing. Furthermore, the sizes of keys, ciphertexts and decryption shares in our scheme are constant w.r.t. the number of parties N; we achieve compactness w.r.t. N.

1 Introduction

Fully homomorphic encryption (FHE) was first realized by Gentry [Gen09] based on ideal lattices. Constructions based on the learning with errors (LWE) problem [Reg09], and its variants ring-LWE (RLWE) [LPR10] or module-LWE (MLWE) [BGV12] are efficient, leading to active research in this field [CKKS17; CGGI20]. Threshold FHE (ThFHE) is an extension of FHE in which ciphertexts can be decrypted by collecting *partial decryption shares* from T out of N parties, where N is the number of parties and T ($\leq N$) is a threshold. In 2023, the US agency NIST initiated a project to establish guidelines and recommendations for implementing threshold cryptosystems including ThFHE [BP23]. ThFHE is an important cryptographic tool that can be applied to construct round

optimal MPC protocols [GGHR14; GLS15; BJMS20] and *universal thresholdizer* [BGG+18], which can be used to add threshold functionality to many cryptosystems, such as CCA-secure PKE, signature schemes, pseudorandom functions (PRF) and functional encryptions.

ThFHE was first constructed by Asharov et al. [AJL+12], and their results have been extended to ThFHE with arbitrary T-out-of-N decryption by Boneh et al. [BGG+18]. A notable drawback of these schemes is that they require (M)LWE with a superpolynomially large modulus to prove security, leading to superpolynomially long keys, ciphertexts and decryption shares. To address this drawback, Boudgoust and Scholl [BS23] proposed a ThFHE scheme based on (M)LWE with a polynomial modulus q, providing a game-based security. However, Passelègue and Stehlé [PS25] recently noted a flaw in the security proof of [BS23]. In response to this attack, Boudgoust and Scholl updated their construction in the full version [BS24], and now their scheme achieves only *selective IND-CPA* security, where the adversary is allowed to query the encryption (and homomorphic evaluation) oracle only before receiving challenge ciphertexts and partial decryption shares.

In addition to the cryptanalysis, Passelègue and Stehlé [PS25] proposed a simulation-secure ThFHE scheme with polynomially short decryption shares, realizing threshold decryption with a small communication size. Although this scheme still requires LWE with a superpolynomially large modulus and the ciphertexts are superpolynomially long, the communication size of sending long FHE ciphertexts can be reduced by transciphering [NLV11; BCK+23], although this requires homomorphic evaluation of the decryption circuit of symmetric key encryption (SKE) such as AES or FHE-friendly SKEs [ARS+15; DEG+18; HKL+22]. Hence, we can achieve threshold FHE with small input/output communications.

One notable drawback of the ThFHE scheme of [PS25] is that it cannot be extended to the construction from MLWE (or RLWE). Thus, it cannot be applied to most instantiations of FHE, including [CKKS17; CGGI20]. This is because ThFHE of [PS25] relies on "yet another" variant of LWE called yaLWE, where the norm of the secret vector of LWE is given to the adversary. Since a (non-tight) reduction from (standard) LWE to yaLWE is proven by Micciancio and Suhl [MS23], ThFHE of [PS25] can be based on LWE. To extend their scheme into an MLWE-based construction, "yet another" variants of MLWE is required. However, no reduction from standard assumption such as MLWE to the required assumption is known. Hence, Passelègue and Stehlé posed the following as a challenging open question:

Question 1.1. Can we construct (simulation-secure) Threshold FHE with polynomially short decryption shares from MLWE (or RLWE)?

Another drawback in ThFHE of [PS25] and prior works [BS23] and [BGG+18, Constr. 5.6] is that the size of the secret key shares is large, e.g., $O(N^{4.2})$ or $O(\binom{N}{T})$ to achieve arbitrary T-out-of-N threshold decryption. This is because their construction relies on $\{0,1\}$-linear secret sharing scheme (LSSS). A simple and efficient method for achieving arbitrary threshold decryption is Shamir secret

Table 1. Summary of our contributions: "●" = satisfied; "◐" = satisfied at high cost; "-" = not satisfied

	Efficiency				Security		
	$q = \mathrm{poly}$		Compact	Shamir	Simulation	Module-	Discrete
	Enc	Dec	w.r.t. N	Sharing	Security	LWE	
[BGG+18]	-	-	◐[1]	◐[2]	●	●	●
[BS23]	●	●	-	-	-[3]	●	●
[PS25]	-[4]	●	-	-	●	-	-
Ours	-[4]	●	●	●	●	●	●

[1] [BGG+18, Constr. 8.35] is based on *universal thresholdizer*, which itself needs (non-compact) ThFHE and NIZK.
[2] [BGG+18, Constr. 5.11] requires q to be super-exponential ($q = \omega((N!)^3)$).
[3] Only achieves *selective IND-CPA* [BS24], because of the attack of [PS25].
[4] Communication size of sending (superpolynomially) long FHE ciphertexts can be reduced (to polynomially short size) by transciphering [NLV11; BCK+23].

sharing, as in [BGG+18, Constr. 5.11]. However, the construction requires super-exponentially large modulus, e.g., $\omega((N!)^3)$ because the Lagrange coefficients blows up the noise; the sizes of ciphertexts and decryption shares greatly increase. Hence, as noted in [PS25], the following open question remains:

Question 1.2. Can we construct Threshold FHE with polynomially short decryption shares from Shamir secret sharing (in stead of $\{0,1\}$-LSSS)?

1.1 Our Contributions

In this paper, we address both Question 1.1 and Question 1.2 at the same time: we propose a simulation-secure Threshold FHE scheme with polynomially short decryption shares from standard LWE/MLWE with Shamir secret sharing by modifying ThFHE of [PS25]. Our core technique is what we call *noise padding*, which adds a small noise ζ whose (scaled) standard deviation is $\sqrt{B_{\mathsf{pub}} - \|\mathbf{s}\|}$ and adjusts the standard deviation of the noise in LWE ciphertexts to be a public value composed of B_{pub}; we thereby prevent the leakage of $\|\mathbf{s}\|$. With this technique, we construct ThFHE without yaLWE and prove its security directly from standard LWE. Furthermore, we also construct a ThFHE scheme from MLWE, by naturally extending our LWE-based ThFHE scheme.

As a side benefit of our construction, our ThFHE scheme becomes compact ($= O(1)$) w.r.t. N; the sizes of ciphertexts and decryption shares do not increase depending on N. Although a compact ThFHE scheme was already proposed in [BGG+18, Constr. 8.35], it utilizes a heavy cryptographic tool called an *universal thresholdizer*, which requires (non-compact) ThFHE and NIZK. In contrast, we construct a compact ThFHE scheme directly from (M)LWE.

Furthermore, as a minor contribution, we construct ThFHE solely with discrete values, while [PS25] requires (flooding) noise to be sampled from

continuous Gaussian distribution. When we implement continuous values, we need to approximate them with, e.g., floating-point numbers. However, it needs to be formally proved that the rounding errors incurred by the approximation do not affect the security and correctness. In contrast, we use discrete Gaussian (flooding) noise instead of the continuous Gaussian and formally analyze correctness and security.

In Table 1, we provide a brief summary of our contributions mentioned above.

1.2 Organization

The remainder of this paper is organized as follows: In Section 2, we provide technical overview of our scheme. In Section 3, we describe necessary definitions and lemmas. Then, we present our LWE-based threshold FHE scheme in Section 4. Finally, we show in Section 5 that our scheme can (naturally) be extended to an MLWE-based threshold FHE scheme.

2 Technical Overview

We first recall the ThFHE schemes of [AJL+12; BGG+18], which are constructed (solely) with a superpolynomially large modulus in Section 2.1. Then, in Section 2.2, we describe the prior work [PS25], which achieves a polynomially small modulus for the decryption share, and we explain why their scheme needs "yet another" variant of LWE. Finally, in Section 2.3, we provide a technical overview of our ThFHE scheme, which can be (directly) based on standard LWE/MLWE.

2.1 Threshold FHE with Noise Flooding [AJL+12; BGG+18]

We first recall the threshold decryption procedure of the ThFHE schemes of [AJL+12; BGG+18]. For simplicity, we assume the underlying FHE is based on LWE. Let the public key be $\mathsf{pk} := (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e})$, where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_Q^{m\times n})$, $\mathbf{s} \in \mathbb{Z}_Q^n$ is a secret key, $\mathbf{e} \in \mathbb{Z}_Q^m$ is a (short) secret error, and Q is a modulus. We assume that ciphertexts, after some homomorphic evaluation, are in the following form:

$$\mathsf{ct}_{\mathsf{eval}} := (\mathbf{a} \sim \mathcal{U}(\mathbb{Z}_Q^n), b := \mathbf{a}^\intercal \mathbf{s} + e_{\mathsf{eval}} + \lfloor \tfrac{q}{2} \rfloor \cdot \mu \bmod Q), \tag{1}$$

where e_{eval} is a (small) noise, $\mu \in \{0, 1\}$ is the plaintext. For simplicity, we now only consider N-out-of-N threshold decryption with additive secret sharing: each party $P_1, \ldots, P_N$ holds a secret key share $\mathbf{s}_1, \ldots, \mathbf{s}_N$ such that $\sum_{i=1}^N \mathbf{s}_i = \mathbf{s}$. Given ct, each party computes a partial decryption share as follows;

$$\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathbf{s}_i) := \mathbf{a}^\intercal \mathbf{s}_i + e_i, \tag{2}$$

and broadcasts pd_i to all parties. Given $\{\mathsf{pd}_i\}_{i\in[N]}$, each party computes

$$\begin{aligned} \overline{\mu} := \mathsf{FinDec}(\mathsf{ct}, \{\mathsf{pd}_i\}_{i\in[N]}) &:= \lfloor (b - \textstyle\sum_{i\in[N]} \mathsf{pd}_i) / \lfloor \tfrac{q}{2} \rfloor \rceil \\ &= \mu + \lfloor (e_{\mathsf{eval}} + \textstyle\sum_{i\in[N]} e_i) / \lfloor \tfrac{q}{2} \rfloor \rceil \end{aligned}$$

and then correctly recovers μ if the parameters are selected so that the error term $\overline{e} := e_{\mathsf{eval}} + \sum_{i=1}^{N} e_i$ satisfies $|\overline{e}| < \lfloor \frac{q}{4} \rfloor$ (with overwhelming probability).

Notably, each parties can also recover the error term by calculating

$$\overline{e} := b - \textstyle\sum_{i\in[N]} \mathsf{pd}_i - \lfloor \frac{q}{2} \rfloor \cdot \overline{\mu} = e_{\mathsf{eval}} + \sum_{i\in[N]} e_i. \tag{3}$$

The error term e_{eval} of the ciphertext after homomorphic evaluations may strongly depend on the input plaintexts: For example, in the CKKS FHE scheme [CHK+18], the error term of the multiplication of two ciphertexts encrypting μ_0, μ_1 with error terms e_0, e_1, is as follows:

$$e_{\mathsf{eval}} = \mu_0 e_1 + \mu_1 e_0 + e'.$$

More simply, even without homomorphic evaluation, the ciphertext noise contains secret information; e.g., the ciphertext noise of Regev PKE [Reg09] has the form of $e_{\mathsf{eval}} = \mathbf{r}^\intercal \mathbf{e}$, where $\mathbf{r}$ is a short vector and $\mathbf{e}$ is the secret error vector of the LWE public key $\mathsf{pk} := (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e})$.

This leak of the error e_{eval} in the ciphertext is not problematic in single-key FHE (or PKE) since the decryption is performed only by the party who owns the secret key $\mathbf{s}$ (and $\mathbf{e}$) and is allowed to recover plaintexts. However, it is obviously problematic in threshold FHE since the decryption is executed by parties who are only permitted to have the share of the secret key $\mathbf{s}_i$ and the input and result plaintexts (excluding the input plaintexts of other parties).

Hence, we need to hide undesirable information in e_{eval} for security. Asharov et al. [AJL+12, Lemma 2.1] presented a simple method to "smudge out" any information in e_{eval}, known as "noise flooding", by adding superpolynomially large noise to the partial decryption share. We rewrite Eq. (2) as

$$\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathbf{s}_i) := \mathbf{a}^\intercal \mathbf{s}_i + \mathfrak{E}_i, \text{ where } |\mathfrak{E}_i| = n^{\omega(1)} \cdot B_{\mathsf{eval}},$$

and we assume $|e_{\mathsf{eval}}| < B_{\mathsf{eval}}$. Then, the error term in the ciphertext is

$$\overline{e} = e_{\mathsf{eval}} + \textstyle\sum_{i\in[N]} \mathfrak{E}_i \approx_{\mathsf{s}} \sum_{i\in[N]} \mathfrak{E}_i,$$

which information-theoretically hides (=smudges out) e_{eval}. A notable drawback is that we need a superpolynomially large modulus Q to satisfy correctness.

2.2 Prior Work: Flood-and-Round Threshold FHE [PS25]

Passelègue and Stehlé [PS25] also use noise flooding and smudge out the noise e_{eval} to prevent the leakage of information. Thus, their ThFHE scheme also requires a superpolynomially large modulus Q for the ciphertexts. However, they switch the modulus Q to a polynomially small modulus q by rounding the ciphertexts, thereby obtaining small decryption shares. This flood-and-round[3]

[3] Passelègue and Stehlé [PS25] call their scheme "double-flood-and-round", as they note that one more rounding can be performed on the partial decryption shares. However, they also mention that they do not use this for security. Thus, we call this method "flood-and-round".

procedure, denoted by $\mathsf{ServerDec}$, takes a superpolynomially long ciphertext after homomorphic evaluation $\mathsf{ct}_{\mathsf{eval}}$ as input and outputs a ciphertext that has been sanitized (by noise flooding) and rounded polynomially short ciphertext $\mathsf{ct}_{\mathsf{dec}}$:

$$\mathsf{ServerDec}(\mathsf{ct}_{\mathsf{eval}} \in \mathbb{Z}_Q^{n+1}) \to \mathsf{ct}_{\mathsf{dec}} \in \mathbb{Z}_q^{n+1}.$$

$\mathsf{ServerDec}$ is executed by a special party called Server, who is untrusted (semi-honest) but assumed not to be corrupted by the adversary. The threshold decryption procedure is performed only on $\mathsf{ct}_{\mathsf{dec}}$, the output of $\mathsf{ServerDec}$; this explains why it is called Server"Dec": it is a part of the decryption process.

We next describe the details of $\mathsf{ServerDec}$. On the input $\mathsf{ct}_{\mathsf{eval}} := (\mathbf{a}, b)$ (defined as in Eq. (1)), it generates $\mathsf{ct}_{\mathsf{flood}} := \mathsf{ct}_{\mathsf{eval}} + (0, \mathfrak{E})$, where $\mathfrak{E}$ is a superpolynomially large (Gaussian) flooding noise. Then, by the smudging lemma for Gaussian [PS25, Lemma 2.1] (Lemma 3.15), we obtain

$$\begin{aligned} \mathsf{ct}_{\mathsf{flood}} &:= (\mathbf{a}, \mathbf{a}^\intercal \mathbf{s} + e_{\mathsf{eval}} + \mathfrak{E} + \lfloor \tfrac{Q}{2} \rfloor \cdot \mu) \\ \approx_{\mathsf{s}} \mathsf{ct}'_{\mathsf{flood}} &:= (\mathbf{a}, \mathbf{a}^\intercal \mathbf{s} + \mathfrak{E} + \lfloor \tfrac{Q}{2} \rfloor \cdot \mu). \end{aligned}$$

Next, the ciphertext with a large modulus Q, constructed as $Q = p \cdot q$ for $p = \Omega(2^\kappa)$ and $q = \mathrm{poly}(\kappa)$, is rounded to one with small modulus q;

$$\mathsf{ct}_{\mathsf{dec}} := \left(\left\lfloor \frac{1}{p} \cdot \mathbf{a} \right\rceil_{\sigma_0}, \left\lfloor \frac{1}{p} \cdot (\mathbf{a}^\intercal \mathbf{s} + \mathfrak{E} + \lfloor \tfrac{Q}{2} \rfloor \cdot \mu) \right\rceil_{\sigma_1} \right) := (\overline{\mathbf{a}}, \overline{b}),$$

where $\lfloor \cdot \rceil_\sigma$ denotes a randomized Gaussian rounding (Definition 3.16) and σ_0 and σ_1 are public parameters. Then, we have

$$\mathsf{ct}_{\mathsf{dec}} = (\overline{\mathbf{a}}, \overline{b} = \overline{\mathbf{a}}^\intercal \mathbf{s} + \mathfrak{e} + \lfloor \tfrac{q}{2} \rfloor \cdot \mu \bmod q), \tag{4}$$

$$\text{where } \mathfrak{e} = \left\lfloor \frac{1}{p} \cdot (-\mathbf{r}^\intercal \mathbf{s} + \mathfrak{E}) \right\rceil_{\sigma_1} \approx_{\mathsf{s}} D_{\mathbb{Z}, \sigma_{\mathsf{dec}}}, \quad \sigma_{\mathsf{dec}} := \sqrt{\sigma_0^2 \|\mathbf{s}\|^2 + \sigma_1^2},$$

and $\mathbf{r} := \mathbf{a} - p \cdot \overline{\mathbf{a}}$ is a rounding error. Thus, the error term $\mathfrak{e}$ depends on the norm of the secret key $\mathbf{s}$. Because of this, [PS25] relies on "yet another" variant of LWE, i.e., the *known-norm* LWE proposed in [MS23]. Although known-norm LWE is reduced from standard LWE, its ring variant, known-covariance ring-LWE, has no reduction from standard assumptions such as RLWE. Thus, it is an open question to extend the scheme of [PS25] to the ring or module setting.

2.3 Our Solution: "Noise Padding" and Masking

We modify the $\mathsf{ServerDec}$ procedure of [PS25] to prevent the leakage of $\|\mathbf{s}\|$. Our core technique is what we call "noise padding", and we construct it with the aid of the "zero share masking" technique proposed by [PKM+24a].

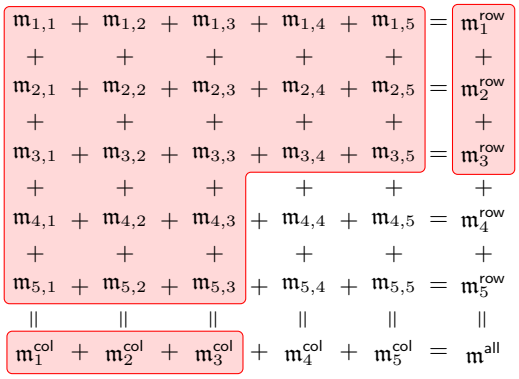

Fig. 1. Relationships between $\mathfrak{m}_{i,j} = \mathsf{PRF}(\mathsf{seed}_{i,j})$, $\mathfrak{m}_i^{\mathsf{row}} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{i,j}$, $\mathfrak{m}_i^{\mathsf{col}} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{j,i}$, and $\mathfrak{m}^{\mathsf{all}} := \sum_{i\in\mathsf{act},j\in\mathsf{act}} \mathfrak{m}_{j,i}$, where we drop the subscript $\mathsf{act} = \{1,\dots,5\}$. An adversary corrupting the user set $\mathsf{corr} = \{1,\dots,3\}$ learns the masks $\{\mathfrak{m}_{i,j}\}_{\min(i,j)\le 3}$ and $\{\mathfrak{m}_i^{\mathsf{row}}, \mathfrak{m}_i^{\mathsf{col}}\}_{i\in\mathsf{corr}}$ (highlighted in red). Note that we let both $\mathfrak{m}_i^{\mathsf{row}}$ and $\mathfrak{m}_i^{\mathsf{col}}$ be private, while [PKM+24a] let the row-sum masks $\mathfrak{m}_i^{\mathsf{row}}$ be public.

Concept. Let $\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b})$ be the output of $\mathsf{ServerDec}(\mathsf{ct})$ as in Eq. (4). Our goal is to add a small error ζ to $\mathsf{ct}_{\mathsf{dec}}$ to adjust the distribution of the error term:

$$\mathsf{ct}'_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}' := \overline{b} + \zeta) = (\overline{\mathbf{a}}, \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \mathfrak{e} + \zeta + \lfloor \tfrac{q}{2} \rfloor \cdot \mu), \tag{5}$$

$$\text{where}\ \ \zeta \sim D_{\mathbb{Z},\sigma_{\mathsf{padding}}},\ \ \sigma_{\mathsf{padding}} := \sqrt{\sigma_0^2(B_{\mathsf{pub}}^2 - \|\mathbf{s}\|^2) + \sigma_1^2},$$

$$\mathfrak{e} + \zeta \approx_{\mathsf{s}} D_{\mathbb{Z},\sigma_{\mathsf{pub}}},\ \ \sigma_{\mathsf{pub}} := \sqrt{\sigma_0^2 B_{\mathsf{pub}}^2 + \sigma_1^2}, \tag{6}$$

for some public constant $B_{\mathsf{pub}} > \|\mathbf{s}\|$ (for any $\mathbf{s}$). Then, the error term $(\mathfrak{e} + \zeta)$ of $\mathsf{ct}'_{\mathsf{dec}}$ does not contain any information about $\|\mathbf{s}\|$: We can construct a ThFHE scheme (directly) from standard LWE.

Masking with zero share [PKM+24a]. Since ζ contains information about $\|\mathbf{s}\|$, it cannot directly be sent to the untrusted (semi-honest) Server or any parties. Hence, we perform secret sharing on ζ, as with the secret key $\mathbf{s}$: $\mathsf{Share}(\zeta) \to \{\zeta_1,\dots,\zeta_N\}$. Here, we now describe with N-out-of-N additive secret-sharing for simplicity, although we support arbitrary T-out-of-N secret-sharing.

Furthermore, we use the masking technique of [PKM+24a] to mask $\{\zeta_i\}$ to hide ζ from the untrusted Server. For $(i,j) \in [N] \times [N]$ let $\mathsf{seed}_{i,j} \leftarrow \{0,1\}^{\kappa}$. Each party P_i is given $\mathsf{seedset}_i := \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j\in[N]}$. Then, the parties can have "pairwise shared" mask value $\mathfrak{m}_{i,j} := \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$ generated with a pseudorandom function PRF, where sid is a session ID of the threshold decryption sessions. See Fig. 1 for more details of the shared masks. We now assume that all parties attend partial decryption protocol (i.e., let the set of active parties $\mathsf{act} = [N]$). Using this mask, each party calculates the masked share of ζ;

$$\mathsf{maskerr}_i := \zeta_i + \mathfrak{m}_i^{\mathsf{row}},$$

and sends it to the Server. Given $\{\mathsf{maskerr}_i\}_{i\in[N]}$, the Server calculates

$$\textstyle\sum_{i\in[N]}\mathsf{maskerr}_i = \sum_{i\in[N]}\zeta_i + \sum_{i\in[N]}\mathfrak{m}_i^{\mathsf{row}} = \zeta + \mathfrak{m}^{\mathsf{all}}$$

and adds it to $\mathsf{ct}_{\mathsf{dec}}$ in Eq. (4), then outputs:

$$\mathsf{ct}'_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}' := \overline{b} + \textstyle\sum_{i\in[N]}\mathsf{maskerr}_i) = (\overline{\mathbf{a}}, \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \mathfrak{e} + \zeta + \lfloor \frac{q}{2} \rfloor \cdot \mu + \mathfrak{m}^{\mathsf{all}}). \quad (7)$$

This is almost identical to Eq. (5), except that the mask $\mathfrak{m}^{\mathsf{all}}$ is added.

Note that Server is untrusted (semi-honest) but assumed not to be corrupted by the adversary (among the parties), as with the prior work [PS25]. Hence, any $\mathfrak{m}_{i,j}$, $\mathfrak{m}_i^{\mathsf{row}}$ and $\mathfrak{m}^{\mathsf{all}}$ are pseudorandom to the Server, and thus, no information about ζ is revealed to the Server.

Masked partial decryption. To decrypt $\mathsf{ct}'_{\mathsf{dec}}$ in Eq. (7), each party P_i calculates a partial decryption as follows:

$$\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}'_{\mathsf{dec}}, \mathbf{s}_i) := \overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}_i^{\mathsf{col}} \quad (8)$$

Recall that only the party P_i knows $\mathfrak{m}_i^{\mathsf{col}}$ (under the pseudorandomness assumption for PRF), as can be seen from Fig. 1. Thus, pd_i is essentially pseudorandom to the adversary, which is the idea behind our security proof. Given all decryption shares $\{\mathsf{pd}_i\}_{i\in[N]}$, the parties can calculate

$$\textstyle\sum_{i\in[N]}\mathsf{pd}_i = \sum_{i\in[N]}(\overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}_i^{\mathsf{col}}) = \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \mathfrak{m}^{\mathsf{all}},$$

since $\mathfrak{m}^{\mathsf{all}} = \sum_{i\in[N]}\mathfrak{m}_i^{\mathsf{col}} = \sum_{i\in[N]}\mathfrak{m}_i^{\mathsf{row}}$ holds by construction. Thus, parties can decrypt as follows:

$$\mathsf{FinDec}(\mathsf{ct}'_{\mathsf{dec}}, \{\mathsf{pd}_i\}_{i\in[N]}) := \overline{\mu} := \lfloor (\overline{b}' - \textstyle\sum_{i\in[N]}\mathsf{pd}_i)/\lfloor \frac{q}{2} \rfloor \rceil = \mu + \lfloor (\mathfrak{e}+\zeta)/\lfloor \frac{q}{2} \rfloor \rceil. \quad (9)$$

Since we select the parameters so that $|\mathfrak{e} + \zeta| < \lfloor \frac{q}{4} \rfloor$ holds (with overwhelming probability), $\overline{\mu} = \mu$ holds. This also means that the error term is revealed:

$$\overline{b}' - \textstyle\sum_{i\in[N]}\mathsf{pd}_i - \lfloor \frac{q}{2} \rfloor \cdot \overline{\mu} = \mathfrak{e} + \zeta \quad (10)$$

Importantly, recall that the distribution of $\mathfrak{e} + \zeta$ contains no information about the secret key $\mathbf{s}$ by our construction:

$$\mathfrak{e} + \zeta \approx_{\mathsf{s}} D_{\mathbb{Z},\sigma_{\mathsf{pub}}}, \quad \sigma_{\mathsf{pub}} := \sqrt{\sigma_0^2 B_{\mathsf{pub}}^2 + \sigma_1^2} \quad \text{(same as Eq.(6))}$$

Equivalently, this means that we can simulate $\mathfrak{e} + \zeta$ in the security proof. Therefore, we can construct our scheme directly from standard LWE, without relying on the "yet another" variant as in [PS25].

In the following paragraphs, we explain two side benefits of our construction.

Side benefit 1: Compactness w.r.t. N. The error term in the FinDec output of the prior works [AJL+12; PS25] is in the form of $e_{\mathsf{eval}} + \sum_{i\in[N]} e_i$ as in Eq. (3). Since its standard deviation is $O(\sqrt{N})$ w.r.t. N, the moduli (q and Q) should also be selected as $O(\sqrt{N})$. Thus, the length of their ciphertexts and public keys increase depending on N. In contrast, the error term in the FinDec output of our scheme is $\mathfrak{e} + \zeta$, whose the standard deviation is independent on N, i.e., $O(1)$ w.r.t. N. Thus, we achieve compactness. This is because we do not add an error in the partial decryption share as in Eq. (8); instead, we add "zero share" mask $\mathfrak{m}_i^{\mathsf{col}}$, which will be canceled out to zero during FinDec, as in Eq. (9).

Side benefit 2: Shamir-sharing-friendly. The masking technique of [PKM+24a] is introduced to address the difficulty of handling Lagrange coefficients in lattice-based threshold signatures. Our construction demonstrates that this technique can also be applied to overcome the difficulty in threshold (fully homomorphic) encryption. We first explain the difficulty of handling Lagrange coefficients. The ThFHE with Shamir sharing of [BGG+18, Constr. 5.11] constructs the partial decryption in the same form as Eq. (2), and the Lagrange coefficient $\lambda_{\mathsf{act},i}$ is multiplied when summing the partial decryptions during FinDec;

$$\textstyle\sum_{i\in\mathsf{act}} \lambda_{\mathsf{act},i} \cdot \mathsf{pd}_i = \mathbf{a}^\intercal \mathbf{s}_i + \sum_{i\in\mathsf{act}} \lambda_{\mathsf{act},i} \cdot e_i \mod q,$$

where $\mathsf{act} \subseteq [N]$ is the set of active parties in the decryption session s.t. $|\mathsf{act}| \geq T$. However, this requires the modulus q to scale with $O(N!^2)$ to ensure that large Lagrange coefficients relatively small to q. A naive approach to address this inefficiency is to construct the partial decryption as follows:

$$\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathbf{s}_i, \mathsf{act}) := \lambda_{\mathsf{act},i} \cdot \mathbf{a}^\intercal \mathbf{s}_i + e_i \mod q. \tag{11}$$

While correctness is satisfied, there is a trivial attack on this construction since the adversary can choose act to craft the coefficients $\lambda_{\mathsf{act},i}$, and e_i is relatively small compared to q. Let $\mathsf{act} \subseteq [N]$ s.t. $\widehat{\lambda} := \lambda_{\mathsf{act},i}$ satisfies $q = \widehat{\lambda} \cdot q'$ for some integer $q' < q$, and assume $|e_i| < \widehat{\lambda}$ holds (with overwhelming probability). Then, we have $b'_i := \lfloor \frac{\mathsf{pd}_i}{\widehat{\lambda}} \rfloor \bmod q' = \mathbf{a}^\intercal \mathbf{s}_i \bmod q'$. By querying a sufficient number of such b'_i, the lower bits of the secret share ($\mathbf{s}_i \bmod q'$) can be recovered via simple linear algebra.

This type of attack is not possible in our scheme. In our scheme, the partial decryption share for T-out-of-N threshold decryption is defined as follows:

$$\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}_{\mathsf{dec}}, \mathbf{s}_i, \mathsf{act}) := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} \bmod q.$$

In contrast to Eq. (11), the mask value of $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ of any uncorrupted party P_i is essentially pseudorandom over $\mathbb{Z}_q$, except that $\sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} = \sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} = \mathfrak{m}^{\mathsf{all}}$ holds by construction (see Fig. 1) and $\mathfrak{m}^{\mathsf{all}}$ is included in the ciphertext as described in Eq. (7). This condition is used to simulate the partial decryption in the security proof. We now let $|\mathsf{act}| = T$ and $\mathsf{corr} = \mathsf{act}\backslash\{h\}$ be the set of the corrupted parties. Then, using Eq. (10), we can simulate pd_h as follows:

$$\mathsf{pd}_h \approx_{\mathsf{s}} \overline{b}' - \lfloor \tfrac{q}{2} \rfloor \cdot \mu - \textstyle\sum_{i\in\mathsf{corr}} \mathsf{pd}_i + e_{\mathsf{Sim}}, \text{ where } e_{\mathsf{Sim}} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{pub}}} (\text{see, Eq.}(6)).$$

Given simulated pd_h instead of real one (in the ideal experiment), the adversary only has information related to $\{\mathbf{s}_i\}_{i\in\mathsf{corr}}$. Due to the security of secret sharing, no information about $\mathbf{s}_h$ (and $\mathbf{s}$) is obtained from $\{\mathbf{s}_i\}_{i\in\mathsf{corr}}$.

2.4 Future Work

In this work, we present a threshold FHE scheme with polynomially short partial decryption shares from (LWE or) Module-LWE, addressing the open problem posed by [PS25]. Furthermore, as side effects of our construction, we achieve compactness w.r.t. the number of parties N and enable efficient (arbitrary) T-out-of-N threshold decryption with Shamir secret sharing (see Table 1 for the summary of our contributions).

A notable open question in our scheme is that we need to bound the number of partial decryption queries, as in the prior work [BS23]. We need to generate a number of secret shares of the padding noise ζ that is equal to the decryption query bound L_{Dec} and distribute them to the parties at the setup stage. As a result, the storage cost for the shares linearly increases w.r.t. L_{Dec}. However, it is important to note that we can choose the value L_{Dec} independently of any other security-related parameters, unlike in [BS23]. Thus, this limitation is not problematic for use-cases in which the number of decryption queries is small or the parties can tolerate a large storage cost. Alternatively, this drawback can be easily removed if we may assume a trusted third party (TTP) and ask TTP to distribute shares of ζ_i for every partial decryption session. We leave it as an open problem to improve our scheme by removing the bound on L_{Dec}.

3 Preliminaries

In this section, we provide necessary definitions and lemmas.

3.1 Notation

We denote the base 2 logarithm by log. For $N \in \mathbb{N}$, we define $[N] := \{i \in \mathbb{N} \mid 1 \le i \le N\}$. The number of elements in a set S is denoted by $|S|$. We define $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ and $\mathbb{R}_q := \mathbb{R}/q\mathbb{R}$ for a modulus $q \in \mathbb{N}$. Unless otherwise specified, we treat κ as a security parameter. We write $\mathrm{negl}(\kappa) = \kappa^{-\omega(1)}$ for the set of negligible functions and $\mathrm{poly}(\kappa) = \kappa^{O(1)}$ for the set of polynomial functions. We call $1 - \mathrm{negl}(\kappa)$ overwhelming functions.

We use bold lower-case letters for vectors and bold upper-case letters for matrices. The transpose of $\mathbf{x}$ is written as $\mathbf{x}^\intercal$. We denote the l_2-norm and l_∞-norm of $\mathbf{x}$ by $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_\infty$, respectively. The identity matrix is denoted by $\mathbf{I}_n \in \mathbb{Z}^{n\times n}$. We write $\mathbf{\Sigma} \succ 0$ if $\mathbf{\Sigma}$ is positive definite. A square root of $\mathbf{\Sigma} \succ 0$ is a nonsingular matrix $\mathbf{S}$ such that $\mathbf{S}\mathbf{S}^\intercal = \mathbf{\Sigma}$, which is written as $\mathbf{S} = \sqrt{\mathbf{\Sigma}}$. Note that $(\sqrt{\mathbf{\Sigma}})^{-1} = \mathbf{S}^{-1} = (\mathbf{S}^{-\intercal})^\intercal = (\sqrt{\mathbf{\Sigma}^{-1}})^\intercal$ holds. The largest and smallest singular values of a matrix $\mathbf{S}$ are denoted by $\sigma_{\max}(\mathbf{S})$ and $\sigma_{\min}(\mathbf{S})$, respectively. Similarly, the largest and smallest eigenvalues are denoted by $\lambda_{\max}(\mathbf{S})$ and $\lambda_{\min}(\mathbf{S})$. We

denote by $\|\mathbf{S}\|$ the matrix norm of $\mathbf{S}$ induced by the l_2-norm. Let $\|\mathbf{S}\|_{\text{len}} = \max_{i\in[n]} \|\mathbf{s}_i\|$, where $\mathbf{s}_i$ is the i-th column vector of $\mathbf{S}$; then, we have:

Fact 3.1. $\|\mathbf{S}\|_{\text{len}} \leq \|\mathbf{S}\|$ *and* $\|\mathbf{S}_1\mathbf{S}_2\|_{\text{len}} \leq \|\mathbf{S}_1\|\|\mathbf{S}_2\|_{\text{len}} \leq \|\mathbf{S}_1\|\|\mathbf{S}_2\|$.

3.2 Statistics

We denote the uniform distribution over a set S by $\mathcal{U}(S)$. For any distributions χ_1 and χ_2, we denote by $\chi_1+\chi_2$ the distribution $\{X_1+X_2 \mid X_1 \leftarrow \chi_1$ and $X_2 \leftarrow \chi_2$ are mutually independent$\}$. We denote $X_1 \approx X_2$ if X_1 and X_2 are identically distributed. We provide other necessary definitions as follows:

Definition 3.2. *The statistical distance between χ_1 and χ_2 is defined as $\Delta(\chi_1,\chi_2) := \frac{1}{2}\sum_{x\in\Omega}|f_{\chi_1}(x) - f_{\chi_2}(x)|$, where $f_{\chi_1}(x)$ and $f_{\chi_2}(x)$ are the probability functions of χ_1 and χ_2, respectively, and $\Omega := \text{Supp}(\chi_1) \cup \text{Supp}(\chi_2)$.*

Definition 3.3 ($\approx_{\mathsf{s}}$). *The distributions χ_1 and χ_2 are statistically indistinguishable and are denoted as $\chi_1 \approx_{\mathsf{s}} \chi_2$ if we have $\Delta(\chi_1,\chi_2) = \text{negl}(\kappa)$.*

Definition 3.4 ($\approx_{\mathsf{c}}$). *The distributions χ_1 and χ_2 over the set Ω are computationally indistinguishable and are denoted as $\chi_1 \approx_{\mathsf{c}} \chi_2$ if $|\Pr[\mathcal{A}(\chi_1) = 1] - \Pr[\mathcal{A}(\chi_2) = 1]| = \text{negl}(\kappa)$ holds for any PPT algorithm $\mathcal{A} : \Omega \to \{0,1\}$.*

Definition 3.5 (B-bounded distribution). *For $B > 0$, we say a distribution χ over $\mathbb{R}$ is B-bounded if $\Pr_{X\leftarrow\chi}[|X| \geq B] = \text{negl}(n)$*

3.3 Lattices

A lattice $\mathcal{L}$ is the set of all integer linear combinations of linearly independent vectors $\mathbf{b}_1,\ldots,\mathbf{b}_n \in \mathbb{R}^m$, i.e., $\mathcal{L} = \{\sum_{i=1}^n z_i\mathbf{b}_i \mid \mathbf{z} \in \mathbb{Z}^n\}$. The rank of this lattice is n and its dimension is m. If $n = m$, then the lattice is called full rank. If we arrange the vectors $\mathbf{b}_i$ as the columns of a matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$, then we can write $\mathcal{L} := \mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} \mid \mathbf{z} \in \mathbb{Z}^n\} = \mathbf{B}\cdot\mathbb{Z}^n$. For arbitrary $\mathbf{c} \in \mathbb{R}^m$, a coset of lattice $\mathcal{L}$ is defined as $A := \mathcal{L} + \mathbf{c} := \{\mathbf{v} + \mathbf{c} \mid \mathbf{v} \in \mathcal{L}\}$. The dual of a lattice $\mathcal{L}$ is $\widehat{\mathcal{L}} := \{\mathbf{x} \mid \forall \mathbf{y} \in \mathcal{L}, \langle\mathbf{x},\mathbf{y}\rangle \in \mathbb{Z}\}$. For the n-rank lattice $\mathcal{L}$ and $i \in [n]$, the successive minimum $\lambda_i(\mathcal{L})$ is defined as the radius of the smallest ball that contains i linearly independent vectors in $\mathcal{L}$.

3.4 Gaussians

For a rank-n matrix $\mathbf{S} \in \mathbb{R}^{n\times m}$, the Gaussian function on $\mathbb{R}^n$ with the (scaled) covariance matrix $\mathbf{\Sigma} = \mathbf{SS}^{\intercal} \in \mathbb{R}^{n\times n}$ and a center $\mathbf{c} \in \mathbb{R}^n$ is defined as follows:

$$\rho_{\mathbf{S},\mathbf{c}}(\mathbf{x}) := \exp(-\pi(\mathbf{x}-\mathbf{c})^{\intercal}(\mathbf{SS}^{\intercal})^{-1}(\mathbf{x}-\mathbf{c})).$$

Since the function $\rho_{\mathbf{S},\mathbf{c}}(\mathbf{x})$ is determined exactly by $\mathbf{\Sigma}$ (and $\mathbf{c}$), we have $\rho_{\mathbf{S},\mathbf{c}} = \rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}$. When $\mathbf{S} = s\mathbf{I}_n$, we write $\rho_{\mathbf{S},\mathbf{c}}$ as $\rho_{s,\mathbf{c}}$. For any set $A \subseteq \mathbb{R}^n$, we define $\rho_{\mathbf{S},\mathbf{c}}(A) := \sum_{\mathbf{x}\in A}\rho_{\mathbf{S},\mathbf{c}}(\mathbf{x})$. The subscript is $\mathbf{c} = \mathbf{0}$ when omitted.

We define the discrete Gaussian distribution over the lattice $\mathcal{L}$ as follows:

Definition 3.6 (Discrete Gaussian). *For a full column-rank matrix* $\mathbf{S}$*, the discrete Gaussian distribution over a lattice* $\mathcal{L}$ *with a center* $\mathbf{c}$ *is defined as* $\forall \mathbf{x} \in \mathcal{L}, D_{\mathcal{L},\mathbf{S},\mathbf{c}}(\mathbf{x}) = \rho_{\mathbf{S},\mathbf{c}}(\mathbf{x})/\rho_{\mathbf{S},\mathbf{c}}(\mathcal{L})$*. In particular, when* $\mathbf{S}\mathbf{S}^\intercal = s^2\mathbf{I}_n$ *for some* $s > 0$*, we write* $D_{\mathcal{L},\mathbf{S},\mathbf{c}}$ *as* $D_{\mathcal{L},s,\mathbf{c}}$ *and call it as the spherical discrete Gaussian distribution. The subscript is* $\mathbf{c} = \mathbf{0}$ *when omitted.*

The *smoothing parameter* of $\mathcal{L}$ is defined as $\eta_\epsilon(\mathcal{L}) = \min\{s \mid \rho_{1/s}(\widehat{\mathcal{L}}) \leq 1+\epsilon\}$ for $\epsilon > 0$. Unless otherwise specified, we set $\epsilon = \mathrm{negl}(\kappa)$. An upper-bound of $\eta_\epsilon(\mathcal{L})$ is obtained with the successive minimum[4] $\tilde{\lambda}_n(\mathcal{L})$:

Lemma 3.7 ([MR07, Lemma 3.3]). *Define* $\eta^+_\epsilon(\mathbb{Z}^n) := \sqrt{\ln(2n(1+1/\epsilon))/\pi}$*. For any rank-n lattice* $\mathcal{L}$ *and any* $\epsilon > 0$*, we have* $\eta_\epsilon(\mathcal{L}) \leq \tilde{\lambda}_n(\mathcal{L}) \cdot \eta^+_\epsilon(\mathbb{Z}^n)$*.*

In particular, $\eta_\epsilon(\mathbb{Z}^n) \leq \eta^+_\epsilon(\mathbb{Z}^n)$ *holds; for any* $\omega(\sqrt{\log n})$ *function, there is* $\epsilon = \mathrm{negl}(\kappa)$ *s.t* $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$*.*

The smoothing parameter is extended to matrices as follows:

Definition 3.8 ([Pei10, Definition 2.3]). *Let* $\mathbf{\Sigma} \succ 0$ *be any positive definite matrix. For any lattice* $\mathcal{L}$*, we say that* $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\mathcal{L})$ *if* $1 \geq \eta_\epsilon(\sqrt{\mathbf{\Sigma}}^{-1}\mathcal{L})$*.*

We obtain a sufficient condition for showing $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\mathcal{L})$ as follows[5]:

Fact 3.9. *For any full-rank lattice* $\mathcal{L}(\mathbf{B})(= \mathbf{B}\mathbb{Z}^n)$ *and* $\mathbf{\Sigma} \succ 0$*, we have* $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\mathcal{L})$ *if* $\sigma_{\min}(\sqrt{\mathbf{\Sigma}}) \geq \|\mathbf{B}\|_{\mathrm{len}}\eta^+_\epsilon(\mathbb{Z}^n)$*.*

Proof. By Fact 3.1, Lemma 3.7 and the hypothesis, we have $\eta_\epsilon(\sqrt{\mathbf{\Sigma}}^{-1}\mathcal{L}) \leq \tilde{\lambda}_n(\sqrt{\mathbf{\Sigma}}^{-1}\mathcal{L})\eta^+_\epsilon(\mathbb{Z}^n) \leq \|\sqrt{\mathbf{\Sigma}}^{-1}\mathbf{B}\|_{\mathrm{len}}\eta^+_\epsilon(\mathbb{Z}^n) \leq \|\sqrt{\mathbf{\Sigma}}^{-1}\|\|\mathbf{B}\|_{\mathrm{len}}\eta^+_\epsilon(\mathbb{Z}^n) \leq 1$. □

The linear sum of discrete Gaussians is close to a discrete Gaussian:

Lemma 3.10 ([MP13, Theorem 3.3]). *Let* $\mathcal{L}$ *be an* n*-dimensional lattice,* $\epsilon = \mathrm{negl}(\lambda)$*,* $\mathbf{z} \neq \mathbf{0} \in \mathbb{Z}^m$*,* $s_i \geq \sqrt{2}\|\mathbf{z}\|_\infty\eta_\epsilon(\mathcal{L})$*,* $A_i := \mathcal{L} + \mathbf{c}_i$ *be arbitrary cosets of* $\mathcal{L}$*, and* $\mathbf{y}_i \leftarrow D_{A_i,s_i}$ *for* $i \in [m]$*. Then, we have* $\mathbf{y} := \sum_{i=1}^m z_i\mathbf{y}_i \approx_{\mathsf{s}} D_{Y,s}$*, where* $Y = \gcd(\mathbf{z})\mathcal{L} + \sum_{i=1}^m z_i\mathbf{c}_i$*, and* $s = \sqrt{\sum_{i=1}^m (z_i s_i)^2}$*.*

In particular, if $\gcd(\mathbf{z}) = 1$ *and* $\sum_{i=1}^m z_i\mathbf{c}_i \in \mathcal{L}$*, then* $\mathbf{y} \approx_{\mathsf{s}} D_{\mathcal{L},s}$*.*

More generally, the sum of two ellipsoid discrete Gaussians is statistically close to an ellipsoid discrete Gaussian:

Lemma 3.11 ([GMPW20, Thm. 3]). *Let* $\epsilon = \mathrm{negl}(\kappa)$*. Let* A_1 *and* A_2 *be cosets of full-rank lattices* $\mathcal{L}_1$ *and* $\mathcal{L}_2$*, respectively. Let* $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2 \succ 0$ *be positive definite matrices and define* $\mathbf{\Sigma}_3 := (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}$*. Let* $X := \{(\mathbf{x}_1, \mathbf{x}_2) \mid \mathbf{x}_1 \leftarrow D_{A_1,\sqrt{\mathbf{\Sigma}_1}}, \mathbf{x}_2 \leftarrow \mathbf{x}_1 + D_{A_2-\mathbf{x}_1,\sqrt{\mathbf{\Sigma}_2}}\}$*. If* $\sqrt{\mathbf{\Sigma}_2} \geq \eta_\epsilon(\mathcal{L}_2)$ *and* $\sqrt{\mathbf{\Sigma}_3} \geq \eta_\epsilon(\mathcal{L}_1)$ *hold, The marginal distribution of* $\mathbf{x}_2$ *in* X *is statistically close to* $D_{\mathcal{L}_2,\sqrt{\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}}$*.*

[4] Although [GPV08, Lemma 3.1] provides a sharper bound with the *Gram–Schmidt minimum*, we use Lemma 3.7 for ease of analysis.

[5] Although sharper bounds can be obtained by the successive minimum $\tilde{\lambda}_n$ or Gram–Schmidt minimum, we rely on $\|\cdot\|_{\mathrm{len}}$ for ease of analysis.

In particular, when $A_1 \subseteq A_2$, we can simplify the above lemma:

Corollary 3.12. *Let $\epsilon = \mathrm{negl}(\kappa)$. Let $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2 \succ 0$ be positive definite matrices and define $\mathbf{\Sigma}_3 := (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}$. Let A_1 and A_2 be cosets of full-rank lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $A_1 \subseteq A_2$. If $\sqrt{\mathbf{\Sigma}_2} \geq \eta_\epsilon(\mathcal{L}_2)$ and $\sqrt{\mathbf{\Sigma}_3} \geq \eta_\epsilon(\mathcal{L}_1)$ hold, we have $D_{A_1,\sqrt{\mathbf{\Sigma}_1}} + D_{A_2,\sqrt{\mathbf{\Sigma}_2}} \approx_{\mathsf{s}} D_{A_2,\sqrt{\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}}$.*

The linear transformation of a discrete Gaussian is analyzed as follows:

Lemma 3.13 ([GMPW20, Lemma 1]). *For any lattice coset $A = \mathcal{L}+\mathbf{c} \subseteq \mathbb{R}^n$ and matrices $\mathbf{S}, \mathbf{T}$ representing linear functions where $\mathbf{T}$ is injective on A, we have $\mathbf{T} \cdot D_{A,\mathbf{S}} = D_{\mathbf{T}\cdot A,\mathbf{TS}}$. In particular, for any $s, t > 0$, we have $t \cdot D_{\mathbb{Z},s} = D_{t\mathbb{Z},ts}$.*

The tail bound of discrete Gaussian can be obtained as follows:

Lemma 3.14. *For any $\epsilon \in (0,1)$, $s \geq \eta_\epsilon^+(\mathbb{Z})$ and $t \in \mathbb{N}$, we have $\Pr_{x\leftarrow D_{\mathbb{Z},s}}[|x| \geq t] \leq \frac{s}{(1-\epsilon)\pi t} e^{-\pi t^2/s^2}$. In particular, for any $t = s \cdot \Omega(\sqrt{\kappa})$, we have $\Pr_{x\leftarrow D_{\mathbb{Z},s}}[|x| \geq t] = \mathrm{negl}(\kappa)$, i.e., $D_{\mathbb{Z},s}$ is $s \cdot \Omega(\sqrt{\kappa})$-bounded (Definition 3.5).*

Proof. We have $\rho_s(\mathbb{Z}) \in (1 \pm \epsilon)s$ by [Reg09, Claim 3.8]. Hence, we obtain

$$\begin{aligned}
&\Pr_{x\leftarrow D_{\mathbb{Z},s}}[|x| \geq t] = 2\Pr_{x\leftarrow D_{\mathbb{Z},s}}[x \geq t] = 2\textstyle\sum_{y=t}^{\infty}\rho_s(y)/\sum_{y\in\mathbb{Z}}\rho_s(y) \\
&\leq \tfrac{2}{1-\epsilon}\textstyle\sum_{y=t}^{\infty}\frac{1}{s}e^{-\pi y^2/s^2} \leq \tfrac{2}{(1-\epsilon)t}\textstyle\sum_{y=t}^{\infty}\frac{y}{s}e^{-\pi y^2/s^2} \leq \tfrac{2}{(1-\epsilon)t}\int_t^{\infty}\frac{y}{s}e^{-\pi y^2/s^2}dy \qquad \square
\end{aligned}$$

Passelègue and Stehlé [PS25] show the smudging lemma for Gaussian:

Lemma 3.15 ([PS25, Lemma 2.1]). *Let $\sigma > 0$ and $c_0, c_1 \in \mathbb{Z}$. Then:*

$$\Delta(D_{\mathbb{Z},\sigma,c_0}, D_{\mathbb{Z},\sigma,c_1}) \leq O\left(\frac{|c_0 - c_1|}{\sigma}\right)$$

In particular, for $\kappa > 0$, $c \in \mathbb{Z}$ and $\sigma = \Omega(c2^\kappa)$, we have $\Delta(D_{\mathbb{Z},\sigma}, D_{\mathbb{Z},\sigma,c}) < 2^{-\kappa}$.

Finally, we define the randomized Gaussian rounding operation:

Definition 3.16 ($\lfloor\cdot\rceil_\sigma$). *For any $n \in \mathbb{N}$, $\mathbf{c} \in \mathbb{R}^n$ and $\sigma > 0$, $\lfloor\mathbf{c}\rceil_\sigma$ denotes the randomized Gaussian rounding that returns $\mathbf{y} \leftarrow D_{\mathbb{Z}^n,\sigma,\mathbf{c}}$.*

3.5 Learning with Errors (LWE)

The LWE distribution and the LWE problem are defined as follows:

Definition 3.17 (LWE). *Let n, m, q be integers and χ be an error distribution over $\mathbb{Z}_q$. The LWE distribution is defined as $\mathsf{LWE}(n, m, q, \chi) := \left\{(\mathbf{A}, \mathbf{b} := \mathbf{As} + \mathbf{e}) \mid \mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n}), \mathbf{s} \leftarrow \chi^n, \mathbf{e} \leftarrow \chi^m\right\}$. The advantage of an algorithm $\mathcal{A}$ for solving $\mathsf{d\text{-}LWE}$ is defined as*

$$\mathrm{Adv}_{\mathcal{A}}^{\mathsf{d\text{-}LWE}} = |\Pr[\mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)) = 1]|.$$

We say that $\mathsf{d\text{-}LWE}$ is hard if $\mathrm{Adv}_{\mathcal{A}}^{\mathsf{d\text{-}LWE}} = \mathrm{negl}(n)$ for any PPT algorithm $\mathcal{A}$.

3.6 Shamir Secret Sharing

We describe a construction of Shamir secret sharing [Sha79] as follows:

Construction 3.18. *(T, N)-Shamir secret sharing for a finite field $\mathbb{K}$ is a tuple of PPT algorithms $\mathsf{Shamir}_{\mathbb{K},T,N} := (\mathsf{Share}, \mathsf{Recon})$ defined as follows:*

- $\mathsf{Share}(s \in \mathbb{K})$*: Choose a degree $T-1$ polynomial $P \in \mathbb{K}[X]$ that satisfies $P(0) = s$ and outputs $(s_1, \ldots, s_N) = (P(1), \ldots, P(N)) \in \mathbb{K}^N$.*
- $\mathsf{Recon}(\{s_i\}_{i \in S \subseteq [N]})$*: Outputs $s = \sum_{i \in S} \lambda_{S,i} s_i$, where $\lambda_{S,i} = \prod_{j \in S \setminus \{i\}} \left(\frac{-j}{i-j}\right)$.*

$\mathsf{Shamir}_{\mathbb{K},T,N}$ *satisfies correctness and privacy:*

- *Correctness: For all $S \subseteq [N]$ s.t. $|S| \geq T$, $s \in \mathbb{K}$, $(s_1, \ldots, s_N) \leftarrow \mathsf{Shamir}_{\mathbb{K},T,N}.\mathsf{Share}(s)$, we have $\mathsf{Shamir}_{\mathbb{K},T,N}.\mathsf{Combine}(\{\mathsf{s}_i\}_{i \in S}) = s$.*
- *Privacy: For all $S \subseteq [N]$ s.t. $|S| < T$, $s^{(0)}, s^{(1)} \in \mathbb{K}$, $(s_1^{(b)}, \ldots, s_N^{(b)}) \leftarrow \mathsf{Shamir}_{\mathbb{K},T,N}.\mathsf{Share}(s^{(b)})$ for $b \in \{0,1\}$, we have $\{s_i^{(0)}\}_{i \in S} \approx \{s_i^{(1)}\}_{i \in S}$.*

The above Shamir secret sharing scheme naturally extends to secrets in vector form, by performing the algorithm in an elementwise manner.

4 Our Threshold FHE Scheme from LWE

In this section, we present our threshold FHE scheme, which can be (directly) constructed from (standard) LWE without using "yet another" variant of LWE as in [PS25].

We first define the structure of underlying FHE scheme in Section 4.1. Then, we describe our construction in Section 4.2. In Section 4.3, we analyze the noise distribution of the output ciphertext $\mathsf{ct}_{\mathsf{dec}}$ of $\mathsf{ServerDec}$ to provide Lemma 4.3. Then, relying on the lemma, we prove the correctness and security of our scheme in Section 4.4 and Section 4.5, respectively.

4.1 Structure of the Underlying FHE Scheme

We specify in Algo. 1 a high-level structure for the underlying FHE scheme based on LWE, which is identical to that of [PS25]. We also define the syntax of FHE as follows:

Definition 4.1 (FHE). *A fully homomorphic encryption (FHE) scheme $\mathsf{FHE} = (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ for a plaintext space $\mathcal{M}$ is defined as follows.*

- $\mathsf{FHE.PP}(1^\kappa) \to \mathsf{pp}$*: On input a security parameter κ, outputs a set of public parameters pp. The following algorithms implicitly take pp as an argument.*
- $\mathsf{FHE.KeyGen}(\mathsf{pp}) \to (\mathsf{evk}, \mathsf{pk}, \mathsf{sk})$*: Outputs a public evaluation key evk, a public key pk and a secret key sk.*
- $\mathsf{FHE.Enc}(\mathsf{pk}, m \in \mathcal{M})$, $\mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{ct})$*: Have the usual syntax for public-key encryption/decryption.*

Algorithm 1: Underlying $\mathsf{FHE} := (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$

$\underline{\mathsf{PP}(1^\kappa, 1^N) \to \mathsf{pp}}$:
1 **return** $\mathsf{pp} := (n, m, Q, \chi_{\mathsf{lwe}}, B_{\mathsf{eval}})$

$\underline{\mathsf{KeyGen}()}$:
2 $\mathsf{sk} := \mathbf{s} \leftarrow \chi_{\mathsf{lwe}}^n$, $\mathsf{pk} := (\mathbf{A}, \mathbf{b} := \mathbf{As} + \mathbf{e}) \sim \mathsf{LWE}(n, m, Q, \chi_{\mathsf{lwe}})$
3 Generate the evaluation key evk ▷ We omit the details of evk
4 **return** $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}(\mathsf{pk}, \mu \in \mathcal{M} := \mathcal{R}_2)}$:
5 $\mathbf{r} \leftarrow \chi_{\mathsf{lwe}}^m, \mathbf{f} \leftarrow \chi_{\mathsf{lwe}}^n, f \leftarrow \chi_{\mathsf{lwe}}$
6 **return** $\mathsf{ct} := (\mathbf{a}, b) := (\mathbf{r}^\intercal \mathbf{A} + \mathbf{f}, \mathbf{r}^\intercal \mathbf{b} + f + \lfloor \frac{Q}{2} \rfloor \cdot \mu)$
Note: PP outputs pp s.t. the noise in the ciphertext $e_{\mathsf{ct}} := b - \mathbf{a}^\intercal \mathbf{s} - \lfloor \frac{Q}{2} \rfloor \cdot \mu$ satisfies $|e_{\mathsf{ct}}| < B_{\mathsf{eval}}$ with overwhelming probability

$\underline{\mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)}$:
7 **return** $\overline{\mathsf{ct}} := (\overline{\mathbf{a}}, \overline{b})$ s.t. $e_{\mathsf{eval}} := \overline{b} - \overline{\mathbf{a}}^\intercal \mathbf{s} - \lfloor \frac{Q}{2} \rfloor \cdot C(\mu_1, \ldots, \mu_l)$ satisfies $|e_{\mathsf{eval}}| < B_{\mathsf{eval}}$ for any C, where $\mu_1, \ldots, \mu_l$ are the plaintexts of $\mathsf{ct}_1, \ldots, \mathsf{ct}_l$

$\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} := (\mathbf{a}, b)) \to \overline{\mu} \in \mathcal{M}}$:
8 **return** $\overline{\mu} := \lfloor (b - \mathbf{a}^\intercal \mathbf{s}) / \lfloor \frac{Q}{2} \rfloor \rceil$

- $\mathsf{FHE.Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l) = \mathsf{ct}_{\mathsf{eval}}$*: The homomorphic evaluation algorithm takes the evaluation key* evk*, a function* $C : \mathcal{M}^l \to \mathcal{M}$*, and a set of* l *ciphertexts* $\mathsf{ct}_1, \ldots, \mathsf{ct}_l$ *of* $\mu_1, \ldots, \mu_l$*. It outputs the result ciphertext* $\mathsf{ct}_{\mathsf{eval}}$*.* FHE *is* correct*, if* $\mathsf{Dec}(\mathsf{sk}, \overline{\mathsf{ct}}) = f(m_1, \ldots, m_l)$ *holds with overwhelming probability.*

We construct our ThFHE scheme in Section 4 on the basis of this underlying FHE scheme. This framework captures most known FHE schemes, except for GSW-like [GSW13] schemes. Note that we can also use ring- or module- variants of this FHE scheme, because we do not require the "known-norm" variant of LWE for security in contrast to ThFHE construction of [PS25]. We also describe a variant of the FHE scheme based on module-LWE in Algo. 13 of Appendix C. We construct our MLWE-based ThFHE in Section 5 upon this underlying MLWE-based FHE scheme.

4.2 Construction and the Threshold Decryption Procedure

Passelègue and Stehlé [PS25] generalized the definition of threshold functional encryption with the additional ServerDec algorithm. Since we slightly modify the definition with our additional MaskErr algorithm, we formally describe the syntax of our threshold FHE as follows:

Definition 4.2 (Threshold FHE). *Let* $P = \{P_1, \ldots, P_N\}$ *be a set of parties and* $T(\leq N)$ *is a threshold. A threshold FHE scheme is a tuple of PPT algorithms* $\mathsf{ThFHE} = (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{MaskErr}, \mathsf{ServerDec}, \mathsf{PartDec}, \mathsf{FinDec})$ *with the following properties:*

- $\mathsf{ThFHE.PP}(1^\kappa) \to \mathsf{pp}$*: On input a security parameter* κ *and a number of parties* N*, outputs a set of public parameters* pp*. The following algorithms implicitly take* pp *as argument.*
- $\mathsf{ThFHE.KeyGen}(T, N) \to (\mathsf{evk}, \mathsf{pk}, \mathsf{sk}, \mathsf{err}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in[N]})$*: On input a threshold* T *and a number of parties* N*, outputs a public evaluation key* evk*, a public key* pk *and a secret key* sk*, a padding error* err *and thier shares* $\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in[N]}$*, as well as sets of the seed values* $\{\mathsf{seedset}_i\}_{i\in[N]}$*.*
- $\mathsf{ThFHE.Enc}(\mathsf{pk}, m) \to \mathsf{ct}$ *and* $\mathsf{ThFHE.Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \dots \mathsf{ct}_l) \to \mathsf{ct}_{\mathsf{eval}}$*: Same as* $\mathsf{FHE.Enc}$ *and* $\mathsf{FHE.Eval}$ *in Definition 4.1.*
- $\mathsf{ThFHE.MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid}) \to \mathsf{maskerr}_i$*: On input an error share* err_i*, a set of seeds* $\mathsf{seedset}_i$*, and a set of active parties* act *for the decryption session with ID* sid*, outputs the masked error share* $\mathsf{maskerr}_i$ *related to* P_i*.*
- $\mathsf{ThFHE.ServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}}) \to \mathsf{ct}_{\mathsf{dec}}$*: On input a ciphertext* ct *(or an evaluation result* $\mathsf{ct}_{\mathsf{eval}}$*),* act*,* sid *and the set of masked error shares* $\{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}}$*, outputs a sanitized ciphertext* $\mathsf{ct}_{\mathsf{dec}}$*.*
- $\mathsf{ThFHE.PartDec}(\mathsf{ct}_{\mathsf{dec}}, \mathsf{sk}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid}) \to \mathsf{pd}_i$*: On input a ciphertext* $\mathsf{ct}_{\mathsf{dec}}$ *outputted by* $\mathsf{ServerDec}$*, a secret key share* sk_i*,* $\mathsf{seedset}_i, \mathsf{act}$ *and* sid*, outputs a partial decryption share* pd_i *related to the party* P_i*.*
- $\mathsf{ThFHE.FinDec}(\{\mathsf{pd}_i\}_{i\in\mathsf{act}}) \to \overline{m}$*: On input a set of partial decryption share* $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$*, outputs a decryption result* $\overline{m} \in \{0,1\}$ *if* $|\mathsf{act}| \geq T$ *and* $\perp$ *otherwise.*

We present our construction of ThFHE in Algo. 2. We assume the deterministic function $\mathsf{PRF} : \{0,1\}^\kappa \times [L_{\mathsf{Dec}}] \to \mathbb{Z}_q$ is pseudorandom function (see Definition A.1 in Appendix A). Our threshold decryption procedure is described in the following.

Setting. As in prior works [AJL+12; BGG+18], we assume all parties are semi-honest, i.e., they follow the protocol but are curious about secret information. In addition, as with [PS25], we assume the existence of a special party called Server, which is untrusted (semi-honest) but assumed not to be corrupted by the adversary (among the parties). Note that security against malicious adversaries can be obtained by a generic transformation aided by NIZK (see, e.g., [AJW11, Section E]), proving that all outputs are (semi-)honestly generated. Also note that we assume a trusted key generation process as in [BGG+18, Constr. 5.11].

Input/output. Let a (possibly evaluated) ciphertext ct, given to the Server be an input of the threshold decryption process. If we apply the threshold decryption protocol to an MPC, each party P_i sends a ciphertext ct_i of its own input plaintext μ_i to the Server (during the first round). Then, the Server performs homomorphic evaluation of C to $\mathsf{ct}_1, \dots, \mathsf{ct}_N$. The result ciphertext $\mathsf{ct}_{\mathsf{eval}}$ is used as an input of the threshold decryption protocol (i.e., an input of the $\mathsf{ServerDec}$). As an output of threshold decryption, all parties (except for the Server) obtain the plaintext of $\mathsf{ct}_{\mathsf{eval}}$.

Algorithm 2: Our threshold FHE from LWE:
$\mathsf{ThFHE} := (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{MaskErr}, \mathsf{ServerDec}, \mathsf{PartDec}, \mathsf{FinDec})$

$\underline{\mathsf{PP}(1^\kappa, 1^N)}$:
return $\mathsf{pp} := (T, n, m, p, q, Q = p \cdot q, \chi_{\mathsf{lwe}}, \sigma_0, \sigma_1, \sigma_{\mathsf{flood}}, B_{\mathsf{pub}}, B_{\mathsf{eval}}, L_{\mathsf{Dec}})$

$\underline{\mathsf{KeyGen}(T, N)}$:
$(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.KeyGen}()$ ▷ $\mathsf{sk} := \mathbf{s} \leftarrow \chi_{\mathsf{lwe}}^n$
$\mathsf{err} := \boldsymbol{\zeta} := (\zeta^{(1)}, \ldots, \zeta^{(L_{\mathsf{Dec}})}) \leftarrow D^{L_{\mathsf{Dec}}}_{\mathbb{Z}, \sigma_0 \sqrt{B_{\mathsf{pub}}^2 - \|\mathbf{s}\|^2}}$
$\{\mathsf{sk}_i := \mathbf{s}_i, \mathsf{err}_i := \boldsymbol{\zeta}_i\}_{i \in [N]} \leftarrow \mathsf{Shamir}_{\mathbb{Z}_q, T, N}.\mathsf{Share}((\mathsf{sk}, \mathsf{err}))$
for $(i, j) \in [N] \times [N]$ **do** $\mathsf{seed}_{i,j} \xleftarrow{\$} \{0,1\}^\kappa$
for $i \in [N]$ **do** $\mathsf{seedset}_i := \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j \in [N]}$
return $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}, \mathsf{err}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i \in [N]})$

$\underline{\mathsf{Enc}(\mathsf{pk}, \mu \in \mathcal{M} := \{0,1\})}$:
return $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, \mu)$ (defined in Algo. 1)

$\underline{\mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)}$:
return $\mathsf{ct}_{\mathsf{eval}} \leftarrow \mathsf{FHE.Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)$ (defined in Algo. 1)

$\underline{\mathsf{MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})}$:
$\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j \in \mathsf{act}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid}) \bmod q$ ▷ Private row-sum mask of P_i
return $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \bmod q$ ▷ Masked share of ζ

$\underline{\mathsf{ServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i \in \mathsf{act}})}$:
$\mathsf{ct}_{\mathsf{fresh}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(0)$, $\mathfrak{E} \leftarrow D_{\mathbb{Z}, \sigma_{\mathsf{flood}}}$ ▷ We use the discrete Gaussian for $\mathfrak{E}$
$\mathsf{ct}_{\mathsf{in}} := (\mathsf{ct}_{\mathsf{in},0}, \mathsf{ct}_{\mathsf{in},1}) := \mathsf{ct} + \mathsf{ct}_{\mathsf{fresh}} + (0, \mathfrak{E}) \bmod Q$,
$\mathsf{ct}_{\mathsf{dec},0} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},0} \right\rceil_{\sigma_0} \bmod q$
$\mathsf{ct}_{\mathsf{dec},1} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},1} \right\rceil_{\sigma_1} + \sum_{i \in \mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} \bmod q$
return $\mathsf{ct}_{\mathsf{dec}} := (\mathsf{ct}_{\mathsf{dec},0}, \mathsf{ct}_{\mathsf{dec},1}) := (\overline{\mathbf{a}}, \overline{b})$

$\underline{\mathsf{PartDec}(\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}), \mathsf{sk}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})}$:
$\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j \in \mathsf{act}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid}) \bmod q$ ▷ Private column-sum mask of P_i
return $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} \bmod q$

$\underline{\mathsf{FinDec}(\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}), \{\mathsf{pd}_i\}_{i \in \mathsf{act}}, \mathsf{act}) \to \overline{\mu} \in \mathcal{M} \text{ or } \perp}$:
assert $\{|\mathsf{act}| \geq T\}$
return $\overline{\mu} := \lfloor (\overline{b} - \sum_{i \in \mathsf{act}} \mathsf{pd}_i) / \lfloor \frac{q}{2} \rfloor \rceil$

First round. Each party P_i in the active party set act of the distributed decryption session with the session ID sid, generates its masked error share $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \leftarrow \mathsf{MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})$, where $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ is the (secret) row-sum mask of P_i. See Fig. 1 for a graphical explanation of the relations between mask terms. The parties then send $\mathsf{maskerr}_i^{(\mathsf{sid})}$ to the Server.

Second round. The Server performs $\mathsf{ServerDec}$ on the input ciphertext ct of large modulus Q to sanitize ct with noise flooding and round it to a ciphertext

$\mathsf{ct}_{\mathsf{dec}}$ with a small modulus q as in [PS25]. In our scheme, the Server further adds a sum of all given masked error shares $\{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}}$ during the $\mathsf{ServerDec}$ procedure, to adjust the distribution of noise in $\mathsf{ct}_{\mathsf{dec}}$ so that it leaks no information about $\|\mathsf{sk}\|$ in contrast to [PS25]. The Server then broadcasts $\mathsf{ct}_{\mathsf{dec}}$ to all parties in act.

Third round. Each party P_i performs $\mathsf{PartDec}$ on $\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b})$ and generates the partial decryption share $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$, where $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ is the (secret) column-sum mask of P_i (see Fig. 1). P_i then broadcasts pd_i to all parties in act. Note that all (semi-honest) parties perform $\mathsf{PartDec}$ only on the ciphertexts sent by the Server, and they never send partial decryption shares to the Server. Additionally, recall that the Server is untrusted (semi-honest) but assumed not to be corrupted by any parties, as in [PS25]; the Server does not obtain any partial decryption shares.

At the end of the third round, given all partial decryption shares $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$, each party P_i performs FinDec to recover the plaintext.

4.3 Noise Analysis

We next analyze the noise distribution of the output ciphertext $\mathsf{ct}_{\mathsf{dec}}$ of $\mathsf{ServerDec}$. Although the analysis is essentially identical to the correctness proof of [PS25, Theorem 5.1], we use a discrete Gaussian for the flooding noise $\mathfrak{E} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{flood}}}$, while [PS25] requires a continuous Gaussian. The following lemma is used to prove both the correctness and security of our threshold FHE scheme:

Lemma 4.3 (Noise analysis). *Let $\epsilon := \mathrm{negl}(\kappa)$, $p \in \mathbb{N}$, and let χ_{lwe} be a B_{lwe}-bounded (Definition 3.5) distribution s.t. $\mathbf{s} \leftarrow \chi_{\mathsf{lwe}}^n$ satisfies $\gcd(\mathbf{s}) = 1$ with overwhelming probability. Let $\sigma_{\mathsf{flood}} \geq \sqrt{2}p\eta_\epsilon(\mathbb{Z})$, $\sigma_0 \geq \sqrt{2}B_{\mathsf{lwe}}\eta_\epsilon(\mathbb{Z})$, $\sigma_1 \geq \frac{\sqrt{2}}{p}\eta_\epsilon(\mathbb{Z})$. Define $\mathfrak{E} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{flood}}}$, $\mathbf{u} \in \mathbb{Z}^n$ and $\mathbf{r} := \mathbf{u} - p \cdot \lfloor \frac{1}{p}\mathbf{u} \rceil_{\sigma_0}$. Then, we have*

$$\mathfrak{e} := \lfloor \tfrac{1}{p}(\mathbf{r}\cdot\mathbf{s} + \mathfrak{E}) \rceil_{\sigma_1} \approx_{\mathsf{s}} D_{\mathbb{Z},\sqrt{(\sigma_0\|\mathbf{s}\|)^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}}. \tag{12}$$

Furthermore, let $B_{\mathsf{pub}} \geq \sqrt{nB^2+1}$ and $\zeta \sim D_{\mathbb{Z},\sigma_0\sqrt{B_{\mathsf{pub}}^2 - \|\mathbf{s}\|^2}}$, then we have

$$\mathfrak{e} + \zeta \approx_{\mathsf{s}} D_{\mathbb{Z},\sigma_{\mathsf{dec}}}, \quad \textit{where } \sigma_{\mathsf{dec}} := \sqrt{(\sigma_0 B_{\mathsf{pub}})^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}. \tag{13}$$

Proof. Note that $\mathbf{r} \sim D_{p\{\frac{1}{p}\mathbf{u}\}+p\mathbb{Z}^n, p\sigma_0}$ holds, where $\{x\} := x - \lfloor x \rceil$ is the fractional part of $x \in \mathbb{R}$. Let $\mathbf{c} := p\{\frac{1}{p}\mathbf{u}\}$, then we have $\mathbf{c} \in p \cdot (\frac{1}{p}\mathbb{Z}^n) = \mathbb{Z}^n$ is an integer vector. Let us denote the i-th elements of $\mathbf{c}$ and $\mathbf{r}$ by c_i and r_i; then, we have $r_i \sim D_{c_i+p\mathbb{Z},p\sigma_0}$ for $i \in [n]$. By the hypothesis, we have $p \cdot \sigma_0 \geq \sqrt{2}B\eta_\epsilon(p\mathbb{Z}) \geq \sqrt{2}\|\mathbf{s}\|_\infty \eta_\epsilon(p\mathbb{Z})$. Hence, by Lemma 3.10, we obtain

$$\mathbf{r}^\intercal\mathbf{s} \approx_{\mathsf{s}} D_{\mathbf{c}^\intercal\mathbf{s}+p\mathbb{Z},p\sigma_0\|\mathbf{s}\|}.$$

Algorithm 3: Correctness game for our ThFHE (Algo. 2).

$\underline{\mathsf{Game}^{\mathsf{correct}}_{\mathsf{ThFHE}}(1^\kappa, T, N, \mathsf{act}, \mathsf{sid}, l, C, (\mu_1, \ldots, \mu_l) \in \mathcal{M}^l)}$:
assert $\{ \mathsf{act} \subseteq [N] \wedge |\mathsf{act}| \geq T \wedge \mathsf{sid} \leq L_{\mathsf{Dec}}\}$
$\mathsf{pp} \leftarrow \mathsf{PP}(1^\kappa, 1^N)$, $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}, \mathsf{err}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in[N]}) \leftarrow \mathsf{KeyGen}(T, N)$
for $j \in [l]$ **do** $\mathsf{ct}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu_j)$
$\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)$
for $i \in \mathsf{act}$ **do** $\mathsf{maskerr}_i^{(\mathsf{sid})} \leftarrow \mathsf{MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})$
$\mathsf{ct}_{\mathsf{dec}} \leftarrow \mathsf{ServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$
for $i \in \mathsf{act}$ **do** $\mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}_{\mathsf{dec}}, \mathsf{sk}_i, \mathsf{seedset}_i)$
$\overline{\mu} \leftarrow \mathsf{FinDec}(\mathsf{ct}_{\mathsf{dec}}, \{\mathsf{pd}_i\}_{i\in\mathsf{act}}, \mathsf{act})$
if $\overline{\mu} = C(\mu_1, \ldots, \mu_l)$ **then return** 1 **else return** 0

Next, since $((\sigma_0\|\mathbf{s}\|)^{-2} + (\sigma_{\mathsf{flood}}/p)^{-2})^{-1/2} \geq \frac{1}{\sqrt{2}}\min(\sigma_0\|\mathbf{s}\|, \sigma_{\mathsf{flood}}/p) \geq \eta_\epsilon(\mathbb{Z})$ holds by the hypothesis $\sigma_{\mathsf{flood}} \geq \sqrt{2}p\eta_\epsilon(\mathbb{Z})$ (and $\sigma_0 \geq \sqrt{2}\eta_\epsilon(\mathbb{Z})$), by Corollary 3.12, we have

$$\mathbf{r}^\intercal\mathbf{s} + \mathfrak{E} \approx_{\mathsf{s}} D_{\mathbb{Z}, \sqrt{(p\sigma_0\|\mathbf{s}\|)^2 + \sigma_{\mathsf{flood}}^2}},$$

where we use the fact that $\mathbf{c}^\intercal\mathbf{s} \in \mathbb{Z}$; thus $\mathbf{c}^\intercal\mathbf{s} + p\mathbb{Z} \subseteq \mathbb{Z}$. Hence, we also have

$$\mathfrak{e}' := \tfrac{1}{p}(\mathbf{r}^\intercal\mathbf{s} + \mathfrak{E}) \approx_{\mathsf{s}} D_{\frac{1}{p}\mathbb{Z}, \sqrt{(\sigma_0\|\mathbf{s}\|)^2 + (\sigma_{\mathsf{flood}}/p)^2}}$$

by Lemma 3.13. Finally, we have $((\sqrt{(\sigma_0\|\mathbf{s}\|)^2 + (\sigma_{\mathsf{flood}}/p)^2})^{-2} + \sigma_1^{-2})^{-1/2} \geq \frac{1}{\sqrt{2}}\min(\sqrt{(\sigma_0\|\mathbf{s}\|)^2 + (\sigma_{\mathsf{flood}}/p)^2}, \sigma_1) \geq \eta_\epsilon(\frac{1}{p}\mathbb{Z})$ by the hypothesis $\sigma_1 \geq \frac{\sqrt{2}}{p}\eta_\epsilon(\mathbb{Z})$. Hence, by Lemma 3.11, we have $\mathfrak{e} = \lfloor\mathfrak{e}'\rceil_{\sigma_1} = D_{\mathbb{Z},\sigma_1,\mathfrak{e}'} = \mathfrak{e}' + D_{\mathbb{Z}-\mathfrak{e}',\sigma_1} \approx_{\mathsf{s}} D_{\mathbb{Z}, \sqrt{(\sigma_0\|\mathbf{s}\|)^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}}$, i.e., Eq. (12). Furthermore, we obtain Eq. (13) by Lemma 3.10 since we have $\sigma_0\sqrt{B_{\mathsf{pub}}^2 - \|\mathbf{s}\|^2} \geq \sigma_0\sqrt{B_{\mathsf{pub}}^2 - nB^2} \geq \sigma_0 \geq \sqrt{2}\eta_\epsilon(\mathbb{Z})$. □

4.4 Correctness

We then define the correctness of our ThFHE scheme.

Definition 4.4 (Correctness). *We define* $\mathsf{Game}^{\mathsf{correct}}_{\mathsf{ThFHE}}$ *in Algo. 3.* ThFHE *(Algo. 2) is correct if, for any* $\kappa, N > 0, T \in [1, N]$, $l > 0$, $C : \mathcal{M}^l \to \mathcal{M}$, $(\mu_1, \ldots, \mu_l) \in M^l$, $\mathsf{act} \subseteq [N]$ *s.t.* $|\mathsf{act}| \geq T$, $L_{\mathsf{Dec}} > 0$ *and* $\mathsf{sid} \leq L_{\mathsf{Dec}}$, *we have:*

$$\Pr[\mathsf{Game}^{\mathsf{ThFHE\text{-}correct}}(1^\kappa, T, N, \mathsf{act}, \mathsf{sid}, l, C, (\mu_1, \ldots, \mu_l) = 1] \geq 1 - \mathrm{negl}(\kappa).$$

We show the correctness of our ThFHE scheme as follows:

Theorem 4.5. *Assume that parameters* $\epsilon, p, \chi_{\mathsf{lwe}}, B_{\mathsf{lwe}}, B_{\mathsf{pub}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ *are selected as in Lemma 4.3. Furthermore, let* $\sigma_{\mathsf{flood}} = \Omega(2^\kappa B_{\mathsf{eval}})$, $Q = p \cdot q = \sigma_{\mathsf{flood}} \cdot \Omega(\sqrt{\kappa})$, *and* $q = \sigma_{\mathsf{dec}} \cdot \Omega(\sqrt{\kappa})$, *where* $\sigma_{\mathsf{dec}} := \sqrt{(\sigma_0 B_{\mathsf{pub}})^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}$. *Then,* ThFHE *(Algo. 2) is correct.*

Proof. The proof is essentially identical to the proof of the correctness of [PS25, Theorem 5.1], except for the additional term $\sum_{i\in\mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})}$ on line 15 in Algo. 2. Additionally, note that we use a discrete Gaussian for $\mathfrak{E} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{flood}}}$ owing to our Lemma 4.3, while [PS25] uses a continuous Gaussian. We first analyze the output $\mathsf{ct}_{\mathsf{dec}} := (\mathsf{ct}_{\mathsf{dec},0}, \mathsf{ct}_{\mathsf{dec},1})$ of $\mathsf{ServerDec}$ on line 6. Let the rounding error of $\mathsf{ct}_{\mathsf{dec},0}$ be denoted as

$$\mathbf{r}_0 := \mathsf{ct}_{\mathsf{in},0} - p \cdot \mathsf{ct}_{\mathsf{dec},0} \sim D_{p\{\frac{1}{p}\mathsf{ct}_{\mathsf{in},0}\}+p\mathbb{Z}^n, p\sigma_0},$$

where $\{x\} := x - \lfloor x \rceil$ is the fractional part of $x \in \mathbb{R}$. Assume that we have $\mathsf{ct}_{\mathsf{in},1} := \mathsf{ct}_{\mathsf{in},0}^{\mathsf{T}}\mathbf{s} + \lfloor \frac{Q}{2} \rfloor \cdot \mu + e_{\mathsf{eval}} + e_{\mathsf{fresh}}$, where e_{eval} and e_{fresh} are the decryption noises of $\mathsf{ct}_{\mathsf{eval}}$ and $\mathsf{ct}_{\mathsf{fresh}}$. Then, as shown in [PS25], we have

$$\mathsf{ct}_{\mathsf{dec},1} = \mathsf{ct}_{\mathsf{dec},0}^{\mathsf{T}}\mathbf{s} + \bar{\mathfrak{e}} + \textstyle\sum_{i\in\mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} + \lfloor \frac{q}{2} \rfloor \cdot \mu \bmod q, \tag{14}$$

where $\bar{\mathfrak{e}} := \lfloor \frac{1}{p}(\mathbf{r}_0 \cdot \mathbf{s} + e_{\mathsf{eval}} + e_{\mathsf{fresh}} + \mathfrak{E}) \rceil_{\sigma_1}$. Furthermore, by the correctness of Shamir secret sharing (Constr. 3.18), we have

$$\textstyle\sum_{i\in\mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} = \sum_{i\in\mathsf{act}} \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}_{\mathsf{act},i}^{\mathsf{row}} = \zeta^{(\mathsf{sid})} + \textstyle\sum_{i\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i}^{\mathsf{row}}. \tag{15}$$

Let $\mathsf{pd}_i := \mathsf{PartDec}(\overline{\mathsf{ct}}, \mathsf{sk}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid}) := \lambda_{\mathsf{act},i} \cdot \mathsf{ct}_{\mathsf{dec},0}^{\mathsf{T}}\mathbf{s}_i + \mathfrak{m}_{\mathsf{act},i}^{\mathsf{col}} \bmod q$ for $i \in \mathsf{act}$; then, again by the correctness of Shamir secret sharing, we have

$$\textstyle\sum_{i\in\mathsf{act}} \mathsf{pd}_i := \mathsf{ct}_{\mathsf{dec},0}^{\mathsf{T}}\mathbf{s} + \sum_{i\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i}^{\mathsf{col}} \bmod q. \tag{16}$$

Finally, by Eqs. (14) to (16), we have

$$\begin{aligned} \mathsf{FinDec}(\overline{\mathsf{ct}}, \{\mathsf{pd}_i\}_{i\in\mathsf{act}}, \mathsf{act}) &= \lfloor (\mathsf{ct}_{\mathsf{dec},1} - \textstyle\sum_{i\in\mathsf{act}} \mathsf{pd}_i)/\lfloor \frac{q}{2} \rfloor \rceil \\ &= \mu + \lfloor (\bar{\mathfrak{e}} + \zeta^{(\mathsf{sid})})/\lfloor \tfrac{q}{2} \rfloor \rceil, \end{aligned} \tag{17}$$

where we use the fact that $\sum_{i\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i}^{\mathsf{col}} = \sum_{i\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i}^{\mathsf{row}}$ by construction (see Fig. 1 for a graphical explanation).

Hence, we only need to analyze the bound of $\bar{\mathfrak{e}} + \zeta^{(\mathsf{sid})}$. Since we have $\sigma_{\mathsf{flood}} = \Omega(2^\kappa B_{\mathsf{eval}})$ by the hypothesis, we obtain $\bar{\mathfrak{e}} \approx_{\mathsf{s}} \mathfrak{e} := \lfloor \frac{1}{p}(\mathbf{r}_0 \cdot \mathbf{s} + \mathfrak{E}) \rceil_{\sigma_1}$ by the smudging lemma Lemma 3.15[6]. Then, by subsequent Lemma 4.3, we have

$$\mathfrak{e} + \zeta^{(\mathsf{sid})} \approx_{\mathsf{s}} D_{\mathbb{Z},\sigma_{\mathsf{dec}}}, \quad \text{where} \quad \sigma_{\mathsf{dec}} := \sqrt{(\sigma_0 B_{\mathsf{pub}})^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}.$$

Note that $D_{\mathbb{Z},\sigma_{\mathsf{dec}}}$ is $\sigma_{\mathsf{dec}} \cdot \Omega(\sqrt{\kappa})$-bounded by Lemma 3.14. Thus, by selecting $q = \sigma_{\mathsf{dec}} \cdot \Omega(\sqrt{\kappa})$, we have $\lfloor (\bar{\mathfrak{e}} + \zeta^{(\mathsf{sid})})/\lfloor \frac{q}{2} \rfloor \rceil = 0$ in Eq. (17) with overwhelming probability; i.e., correctness holds. □

[6] This strong noise flooding lemma is only needed for the security proof, but it is also used for the correctness just for the simplicity of analysis.

Compactness with respect to N. The security of our scheme (subsequent Theorem 4.7) can also be satisfied with the parameters for the correctness Theorem 4.5. Note that all parameters $\epsilon, p, \chi_{\mathsf{lwe}}, B_{\mathsf{lwe}}, B_{\mathsf{pub}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ are independent of N and $O(1)$ w.r.t. N; thus, our ciphertexts (and decryption shares) achieve compactness w.r.t N. This is because we do not need to add a noise for partial decryption share unlike prior works (as in Eq. (2)). Instead, we add a "zero-share" mask ($=\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$) that is canceled out to zero in the decryption procedure (line 18 of Algo. 14).

4.5 Security

We define the security of our ThFHE scheme in Definition 4.6. Intuitively, the definition means that the view of the real world ($:= \mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$), where the adversary receives an honestly generated set of partial decryption shares

$$\{\mathsf{pd}_i\}_{i\in\mathsf{act}} \leftarrow \{\mathsf{PartDec}(\overline{\mathsf{ct}}, \mathsf{sk}_i, \mathsf{seedset}_i)\}_{i\in\mathsf{act}}$$

is (computationally) indistinguishable from the view of the ideal world ($:= \mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$), where the adversary receives simulated inputs

$$\{\mathsf{pd}_i\}_{i\in\mathsf{act}} \leftarrow \mathsf{Sim}(\overline{\mathsf{ct}}, \{\mathsf{sk}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}})$$

generated by some PPT algorithm Sim by using only incomplete share sets $\{\mathsf{sk}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}}$ ($|\mathsf{corr}| < t$) that are given to the adversary. Then, the privacy of secret sharing (Constr. 3.18) implies that the secret share $\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in\mathsf{corr}}$ provide no information regarding $\mathsf{sk}, \mathsf{err}$ to the adversary.

Definition 4.6 (Security). ThFHE *(Algo. 2) is simulation-secure if, for any PPT algorithms $\mathcal{D}$ and $\mathcal{A}$, there exists a PPT simulator* Sim *s.t.* $\mathrm{Adv}^{\mathsf{ThFHE}}_{\mathcal{D},\mathcal{A}}(\kappa) := |\Pr[\mathcal{D}(\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)) = 1] - \Pr[\mathcal{D}(\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}(1^\kappa)) = 1]| = \mathrm{negl}(\kappa)$*, where and* $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)$ *and* $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}(1^\kappa)$ *are defined in Algo. 4 and Algo. 5, respectively.*

Then, we prove the above security in Theorem 4.7.

Theorem 4.7. ThFHE *(Algo. 2) is simulation-secure under the assumption on the pseudorandomness of* PRF *and* d-LWE$(n, m, Q, \chi_{\mathsf{lwe}})$*: Formally, for any PPT algorithms $\mathcal{D}$ and $\mathcal{A}$, there exists PPT algorithms* Sim*, $\mathcal{B}_{\mathsf{LWE}}$ and $\mathcal{B}_{\mathsf{PRF}}$ s.t.*

$$\mathrm{Adv}^{\mathsf{ThFHE}}_{\mathcal{D},\mathcal{A}}(\kappa) < \mathrm{Adv}^{\mathsf{d\text{-}LWE}(n,m,Q,\chi_{\mathsf{lwe}})}_{\mathcal{B}_{\mathsf{LWE}}} + \mathrm{Adv}^{\mathsf{PRF}}_{\mathcal{B}_{\mathsf{PRF}}}(\kappa) + \mathrm{negl}(\kappa),$$

if $\epsilon, p, q, Q, \chi_{\mathsf{lwe}}, B_{\mathsf{lwe}}, B_{\mathsf{pub}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ are selected as in Theorem 4.5.

Proof. With the subsequent Lemmas 4.8–4.14, we obtain the claim. □

We define hybrid experiments $\mathsf{Hyb}_1, \ldots, \mathsf{Hyb}_6$ in Algorithms 6,...,11, respectively. We now prove the deferred lemmas, i.e., the indistinguishability of the hybrid experiments. As a shorthand, for any algorithm $\mathcal{D}$ and experiments $\mathsf{H}_0(1^\kappa)$ and $\mathsf{H}_1(1^\kappa)$, we define $\mathrm{Adv}^{\mathsf{H}_0\text{-}\mathsf{H}_1}_{\mathcal{D}}(\kappa) := |\Pr[\mathcal{D}(\mathsf{H}_0(1^\kappa)) = 1] - \Pr[\mathcal{D}(\mathsf{H}_1(1^\kappa)) = 1]|$.

Algorithm 4: The real experiment $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)$.

$\underline{\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)}$:
1 $\mathsf{pp} \leftarrow \mathsf{PP}(1^\kappa, 1^N)$, $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}, \mathsf{err}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in[N]}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}, T, N)$
2 $(\mathsf{corr}, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})$
3 **assert** $\{\mathsf{corr} \subseteq [N] \wedge |\mathsf{corr}| < T\}$, $\mathsf{hon} := [N]\backslash\mathsf{corr}$, $\mathsf{ctr} \leftarrow 0, \mathsf{sid} \leftarrow 0, \mathsf{List} \leftarrow \emptyset$
4 **return** $\{0,1\} \leftarrow \mathcal{A}_2^{\mathsf{OEnc},\mathsf{ODec}}(\{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}}, \mathsf{st})$

$\underline{\mathsf{OEnc}(\mu)}$:
5 $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu)$, $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$, $\mathsf{List}[\mathsf{ctr}] \leftarrow (\mu, \mathsf{ct})$

$\underline{\mathsf{OEval}(C, (i_1, \ldots, i_l))}$:
6 **assert** $\{(i_1, \ldots, i_l) \subseteq [\mathsf{ctr}]\}$, **for** $j \in [l]$ **do** $(\mu_j, \mathsf{ct}_j) \leftarrow \mathsf{List}[i_j]$
7 $\mu \leftarrow C(\mu_1, \ldots, \mu_l)$, $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)$
8 $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$, $\mathsf{List}[\mathsf{ctr}] \leftarrow (\mu, \mathsf{ct})$

$\underline{\mathsf{ODec}(\mathsf{act})}$:
9 **assert** $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
10 $\mathsf{sid} \leftarrow \mathsf{sid} + 1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
11 **for** $i \in \mathsf{act}$ **do** $\mathsf{maskerr}_i^{(\mathsf{sid})} \leftarrow \mathsf{MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})$
12 $\mathsf{ct}_{\mathsf{dec}} := (\bar{\mathbf{a}}, \bar{b}) \leftarrow \mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$
13 **return** $\{\mathsf{pd}_i\}_{i\in\mathsf{act}} \leftarrow \{\mathsf{PartDec}(\mathsf{ct}_{\mathsf{dec}}, \mathsf{sk}_i, \mathsf{seedset}_i)\}_{i\in\mathsf{act}}$

$\underline{\mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})}$:
14 **return** $\mathsf{ct}_{\mathsf{dec}} \leftarrow \mathsf{ServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$

Algorithm 5: The ideal experiment $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}(1^\kappa)$. Differences from $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)$ are highlighted.

$\underline{\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}(1^\kappa)}$:
1 $\mathsf{pp} \leftarrow \mathsf{PP}(1^\kappa, 1^N)$, $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE.KeyGen}()$
2 Generate $\{\mathsf{seedset}_i\}_{i\in[N]}$ as in $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}(1^\kappa)$
3 $\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in[N]} \leftarrow \mathsf{Shamir.Share}_{\mathbb{Z}_q,T,N}((\mathbf{0}, \mathbf{0}))$ ▷ Shares with no information
4 $(\mathsf{corr}, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})$
5 **assert** $\{\mathsf{corr} \subseteq [N] \wedge |\mathsf{corr}| < T\}$, $\mathsf{hon} := [N]\backslash\mathsf{corr}$, $\mathsf{ctr} \leftarrow 0, \mathsf{sid} \leftarrow 0, \mathsf{List} \leftarrow \emptyset$
6 **return** $\{0,1\} \leftarrow \mathcal{A}_2^{\mathsf{OEnc},\mathsf{ODec}}(\{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}}, \mathsf{st})$

▷ OEnc and OEval are identical to those in $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$.
$\underline{\mathsf{ODec}(\mathsf{act})}$:
7 **assert** $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
8 $\mathsf{sid} \leftarrow \mathsf{sid} + 1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
9 $\{\mathsf{pd}_i\}_{i\in\mathsf{act}} \leftarrow \mathsf{Sim}^{\mathsf{OServerDec}}(\mu, \mathsf{ct}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}}, \mathsf{act})$,
where the algorithm Sim is defined on lines 3-12 of Hyb_6.
10 **return** $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$

$\underline{\mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})}$:
▷ Identical to OServerDec in Hyb_2 (and thus Hyb_6).

Algorithm 6: Hyb_1: Differences from $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$ are highlighted.

▷ Expt, OEnc, OEval and ODec are identical to those in $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$

$\mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$:

1 $\mathsf{ct}_{\mathsf{in}} := (\mathsf{ct}_{\mathsf{in},0} \leftarrow \mathcal{U}(\mathbb{Z}_Q^n), \mathsf{ct}_{\mathsf{in},1} := \mathsf{ct}_{\mathsf{in},0}^\intercal \mathbf{s} + \mathfrak{E} + \lfloor \frac{Q}{2} \rfloor \cdot \mu)$, where μ is the plaintext of ct and $\mathfrak{E} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{flood}}}$.

2 $\mathsf{ct}_{\mathsf{dec},0} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},0} \right\rceil_{\sigma_0} \bmod q$

3 $\mathsf{ct}_{\mathsf{dec},1} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},1} \right\rceil_{\sigma_1} + \sum_{i\in\mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} \bmod q$

4 **return** $\mathsf{ct}_{\mathsf{dec}} := (\mathsf{ct}_{\mathsf{dec},0}, \mathsf{ct}_{\mathsf{dec},1})$

Algorithm 7: Hyb_2: Differences from Hyb_1 are highlighted.

▷ Expt, OEnc, OEval and ODec are identical to those in $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$

$\mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$:

1 $\mathsf{ct}_{\mathsf{dec},0} \leftarrow \mathcal{U}(\mathbb{Z}_q^n), \mathfrak{e} \leftarrow D_{\mathbb{Z},\sqrt{(\sigma_0\|\mathbf{s}\|)^2+(\sigma_{\mathsf{flood}}/p)^2+\sigma_1^2}}$

2 $\mathsf{ct}_{\mathsf{dec},1} := \mathsf{ct}_{\mathsf{dec},0}^\intercal \mathbf{s} + \mathfrak{e} + \lfloor \frac{q}{2} \rfloor \cdot \mu + \sum_{i\in\mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} \bmod q$, where μ is the plaintext of ct

3 **return** $\mathsf{ct}_{\mathsf{dec}} := (\mathsf{ct}_{\mathsf{dec},0}, \mathsf{ct}_{\mathsf{dec},1})$

Lemma 4.8 ($\mathsf{Expt}_{\mathcal{A},\mathsf{Real}} \approx_{\mathsf{c}} \mathsf{Hyb}_1$)**.** *Assume that* $\sigma_{\mathsf{flood}} = \Omega(2^\kappa B_{\mathsf{eval}})$*, then we have* $\mathrm{Adv}_{\mathcal{D}}^{\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}\text{-}\mathsf{Hyb}_1}(\kappa) \le \mathrm{Adv}_{\mathcal{B}_{\mathsf{LWE}}}^{\mathsf{LWE}(m,n,Q,\chi_{\mathsf{lwe}})}(\kappa) + 2^{-\kappa}$.

Proof. Although the proof is identical to that of [PS25, Lemma 5.2], we provide it here for the completeness. Let $\mathsf{ct}_{\mathsf{fresh}} := (\mathbf{a}, b) \leftarrow \mathsf{Enc}_{\mathsf{pk}}(0)$ in the original $\mathsf{ServerDec}$ algorithm (line 12 in Algo. 2), and define its noise as $e_{\mathsf{fresh}} := b - \mathbf{a}^\intercal \mathbf{s}$. Recall $\mathbf{a} := \mathbf{r}^\intercal \mathbf{A} + \mathbf{f}$ as defined in Algo. 1, then we have $\mathbf{a} \approx_{\mathsf{c}} \mathcal{U}(\mathbb{Z}_q^n)$ under the LWE assumption. Since $\mathsf{ct}_{\mathsf{fresh}}$ is not used elsewhere, we also have $\mathsf{ct}_{\mathsf{in},0} \approx_{\mathsf{c}} \mathcal{U}(\mathbb{Z}_q^n)$. Furthermore, we have $\mathsf{ct}_{\mathsf{in},1} := \mathsf{ct}_{\mathsf{in},0}^\intercal \mathbf{s} + \mathfrak{E} + e_{\mathsf{fresh}} + \lfloor \frac{Q}{2} \rfloor \cdot \mu \approx_{\mathsf{s}} \mathsf{ct}_{\mathsf{in},0}^\intercal \mathbf{s} + \mathfrak{E} + \lfloor \frac{Q}{2} \rfloor \cdot \mu$ by the smudging lemma, Lemma 3.15, and the hypothesis $\sigma_{\mathsf{flood}} = \Omega(2^\kappa B_{\mathsf{eval}})$. □

Lemma 4.9 ($\mathsf{Hyb}_1 \approx_{\mathsf{s}} \mathsf{Hyb}_2$)**.** *Assume that parameters* $\epsilon, p, q, B_{\mathsf{lwe}}, \chi_{\mathsf{lwe}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ *are selected as in Lemma 4.3. Then, we have* $\mathrm{Adv}_{\mathcal{D}}^{\mathsf{Hyb}_1\text{-}\mathsf{Hyb}_2}(\kappa) = \mathrm{negl}(\kappa)$.

Proof. The claim holds by the proof of Eq. (12) in Lemma 4.3. □

Lemma 4.10 ($\mathsf{Hyb}_2 \approx_{\mathsf{c}} \mathsf{Hyb}_3$)**.** *We have* $\mathrm{Adv}_{\mathcal{D}}^{\mathsf{Hyb}_2\text{-}\mathsf{Hyb}_3} \le \mathrm{Adv}_{\mathcal{B}_{\mathsf{PRF}}}^{\mathsf{PRF}}(\kappa)$.

Proof. In Hyb_3, for all $\{(i,j) \mid i \in \mathsf{hon}_{\mathsf{sid}} \text{ and } j \in \mathsf{hon}_{\mathsf{sid}}\}$, the challenger samples $\mathfrak{m}_{\mathsf{act},i,j} \leftarrow \mathcal{U}(\mathbb{Z}_q)$ instead of the output of a pseudorandom function. It is easy to see that this hybrid is indistinguishable from Hyb_2 under the assumption on the pseudorandomness of the PRF (Definition A.1). □

Algorithm 8: Hyb_3: Differences from Hyb_2 are highlighted.

▷ Expt, OEnc, OEval and OServerDec are identical to those in Hyb_2

ODec(act):

assert $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
$\mathsf{sid} \leftarrow \mathsf{sid}+1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
for $i \in \mathsf{corr}_{\mathsf{sid}}$ **do**
  for $j \in \mathsf{act}$ **do** $\mathfrak{m}_{\mathsf{act},i,j} := \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$, $\mathfrak{m}_{\mathsf{act},j,i} := \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid})$
  $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,j}$ ▷ Same as in $\mathsf{Expt}_{\mathcal{A},\mathsf{Real}}$
for $i \in \mathsf{hon}_{\mathsf{sid}}$ **do**
  for $j \in \mathsf{hon}_{\mathsf{sid}}$ **do** $\mathfrak{m}_{\mathsf{act},i,j} \leftarrow \mathcal{U}(\mathbb{Z}_q)$ ▷ Sample uniformly
  $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,j}$
for $i \in \mathsf{act}$ **do** $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ ▷ MaskErr without PRF
$\mathsf{ct}_{\mathsf{dec}} := (\bar{\mathbf{a}}, \bar{b}) \leftarrow \mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$
for $i \in \mathsf{act}$ **do** $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \bar{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ ▷ PartDec without PRF
return $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$

Algorithm 9: Hyb_4: Differences from Hyb_3 are highlighted.

▷ Expt, OEnc, OEval and OServerDec are identical to those in Hyb_2

ODec(act):

assert $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
$\mathsf{sid} \leftarrow \mathsf{sid}+1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
for $i \in \mathsf{corr}_{\mathsf{sid}}$ **do**
  for $j \in \mathsf{act}$ **do** $\mathfrak{m}_{\mathsf{act},i,j} := \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$, $\mathfrak{m}_{\mathsf{act},j,i} := \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid})$
  $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,j}$
Fix some $h \in \mathsf{hon}_{\mathsf{sid}}$.
for $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ **do** $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \leftarrow \mathcal{U}(\mathbb{Z}_q)$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} \leftarrow \mathcal{U}(\mathbb{Z}_q)$
$\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},h} \leftarrow \mathcal{U}(\mathbb{Z}_q)$
$\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h} := \sum_{i\in\mathsf{hon}_{\mathsf{sid}}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} - \sum_{i\in\mathsf{hon}_{\mathsf{sid}}\backslash\{h\}} \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} + \sum_{i\in\mathsf{hon}_{\mathsf{sid}}, j\in\mathsf{corr}_{\mathsf{sid}}} (\mathfrak{m}_{\mathsf{act},j,i} - \mathfrak{m}_{\mathsf{act},i,j})$
for $i \in \mathsf{act}$ **do** $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$
$\mathsf{ct}_{\mathsf{dec}} := (\bar{\mathbf{a}}, \bar{b}) \leftarrow \mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$
for $i \in \mathsf{act}$ **do** $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \bar{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$
return $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$

Lemma 4.11 ($\mathsf{Hyb}_3 \approx \mathsf{Hyb}_4$). *The distributions of* Hyb_3 *and* Hyb_4 *are identical.*

Proof. This proof is almost identical to the unforgeability proof of [PKM+24b]. In Hyb_4, the challenger samples $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ and $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ at random for all honest parties except some $h \in \mathsf{hon}_{\mathsf{sid}}$ that is uniquely defined by the other mask values. The row masks $\{\mathfrak{m}^{\mathsf{row}}_i\}_{i\in\mathsf{hon}_{\mathsf{sid}}}$ are distributed uniformly at random in Hyb_3 because these masks include the individual masks $\mathfrak{m}_{\mathsf{act},i,i} \sim \mathcal{U}(\mathbb{Z}_q)$ that are used nowhere else. This is identical to Hyb_4. Furthermore, in Hyb_3, all the column

Algorithm 10: Hyb_5: Differences from Hyb_4 are highlighted.

▷ Expt, OEnc, OEval and OServerDec are identical to those in Hyb_2

$\underline{\mathsf{ODec}(\mathsf{act})}$:

1 **assert** $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
2 $\mathsf{sid} \leftarrow \mathsf{sid}+1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
3 **for** $i \in \mathsf{corr}_{\mathsf{sid}}$ **do**
4 **for** $j \in \mathsf{act}$ **do** $\mathfrak{m}_{\mathsf{act},i,j} := \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$, $\mathfrak{m}_{\mathsf{act},j,i} := \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid})$
5 $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,j}$
6 Fix some $h \in \mathsf{hon}_{\mathsf{sid}}$.
7 **for** $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ **do** $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \leftarrow \mathcal{U}(\mathbb{Z}_q), \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} \leftarrow \mathcal{U}(\mathbb{Z}_q)$
8 $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},h} \leftarrow \mathcal{U}(\mathbb{Z}_q)$ ($\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}$ is no longer used)
9 **for** $i \in \mathsf{act}$ **do** $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$
10 $\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}) \leftarrow \mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i\in\mathsf{act}})$
11 **for** $i \in \mathsf{corr}_{\mathsf{sid}}$ **do** $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$
12 **for** $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ **do** $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$
13 $\mathsf{pd}_h := \overline{b} - \lfloor \frac{q}{2} \rfloor \cdot \mu - \sum_{i\in\mathsf{act}\backslash\{h\}} \mathsf{pd}_i + e_{\mathsf{Sim}}$,
 where $e_{\mathsf{Sim}} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{dec}}}$ and $\sigma_{\mathsf{dec}} := \sqrt{(\sigma_0 B_{\mathsf{pub}})^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}$.
14 **return** $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$

masks $\mathfrak{m}^{\mathsf{col}}_i := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$ for $i \neq h$ include $\mathfrak{m}_{\mathsf{act},h,i}$ used nowhere before the challenger calculates $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},h}$ on line 8. Hence, $\{\mathfrak{m}^{\mathsf{col}}_i\}_{i\in\mathsf{hon}_{\mathsf{sid}}\backslash\{h\}}$ are distributed uniformly random as in Hyb_4. Finally, we analyze the distribution of $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}$. In Hyb_3, we have

$$\begin{aligned}
&\sum_{i\in\mathsf{hon}_{\mathsf{sid}}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} - \sum_{i\in\mathsf{hon}_{\mathsf{sid}}\backslash\{h\}} \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} + \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}},\\ j\in\mathsf{corr}_{\mathsf{sid}}}} (\mathfrak{m}_{\mathsf{act},j,i} - \mathfrak{m}_{\mathsf{act},i,j}) \\
&= \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}},\\ j\in\mathsf{act}}} \mathfrak{m}_{\mathsf{act},i,j} - \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}}\backslash\{h\},\\ j\in\mathsf{act}}} \mathfrak{m}_{\mathsf{act},j,i} + \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}},\\ j\in\mathsf{corr}_{\mathsf{sid}}}} (\mathfrak{m}_{\mathsf{act},j,i} - \mathfrak{m}_{\mathsf{act},i,j}) \\
&= \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}},\\ j\in\mathsf{hon}_{\mathsf{sid}}}} \mathfrak{m}_{\mathsf{act},i,j} - \sum_{\substack{i\in\mathsf{hon}_{\mathsf{sid}}\backslash\{h\},\\ j\in\mathsf{hon}_{\mathsf{sid}}}} \mathfrak{m}_{\mathsf{act},j,i} + \sum_{j\in\mathsf{corr}_{\mathsf{sid}}} \mathfrak{m}_{\mathsf{act},j,h} \\
&= \sum_{i\in\mathsf{hon}_{\mathsf{sid}}} \mathfrak{m}_{\mathsf{act},i,h} + \sum_{j\in\mathsf{corr}_{\mathsf{sid}}} \mathfrak{m}_{\mathsf{act},j,h} = \sum_{i\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,h} = \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}. \qquad (18)
\end{aligned}$$

Thus, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}$ in Hyb_3 and Hyb_4 are identically distributed. □

Lemma 4.12 ($\mathsf{Hyb}_4 \approx_{\mathsf{s}} \mathsf{Hyb}_5$). *Assume* $B_{\mathsf{pub}} \geq \|\mathbf{s}\|^2 + 1$ *and* $\sigma_0 \geq \sqrt{2}\eta_\epsilon(\mathbb{Z})$, *then we have* $\mathrm{Adv}^{\mathsf{Hyb}_4\text{-}\mathsf{Hyb}_5}_{\mathcal{D}} = \mathrm{negl}(\kappa)$.

Proof. We analyze the distribution of the partial decryption of the party P_h in Hyb_4; $\mathsf{pd}_h := \lambda_{\mathsf{act},h} \cdot \overline{\mathbf{a}}^{\intercal}\mathbf{s}_h + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}$. Note that Eq. (18) implies that $\sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} =$

Algorithm 11: Hyb_6: Differences from Hyb_5 are highlighted.

▷ Expt, OEnc, OEval and OServerDec are identical to those in Hyb_2

$\underline{\mathsf{ODec}(\mathsf{act})}$:

1 **assert** $\{|\mathsf{act}| \geq T \wedge \mathsf{sid} < \min(\mathsf{ctr}, L_{\mathsf{Dec}})\}$
2 $\mathsf{sid} \leftarrow \mathsf{sid}+1$, $(\mu, \mathsf{ct}) \leftarrow \mathsf{List}[\mathsf{sid}]$, $\mathsf{corr}_{\mathsf{sid}} := \mathsf{corr} \cap \mathsf{act}$, $\mathsf{hon}_{\mathsf{sid}} := \mathsf{hon} \cap \mathsf{act}$
3 **for** $i \in \mathsf{corr}_{\mathsf{sid}}$ **do**
4 | **for** $j \in \mathsf{act}$ **do** $\mathfrak{m}_{\mathsf{act},i,j} := \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid})$, $\mathfrak{m}_{\mathsf{act},j,i} := \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid})$
5 | $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},j,i}$, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j\in\mathsf{act}} \mathfrak{m}_{\mathsf{act},i,j}$
6 Fix some $h \in \mathsf{hon}_{\mathsf{sid}}$. (For $i \in \mathsf{hon}_{\mathsf{sid}}$, $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ and $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ are no longer sampled)
7 **for** $i \in \mathsf{corr}_{\mathsf{sid}}$ **do** $\mathsf{maskerr}^{(\mathsf{sid})}_i := \lambda_{\mathsf{act},i} \cdot \zeta^{(\mathsf{sid})}_i + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ ▷ Same as in Hyb_5
8 **for** $i \in \mathsf{hon}_{\mathsf{sid}}$ **do** $\mathsf{maskerr}^{(\mathsf{sid})}_i \leftarrow \mathcal{U}(\mathbb{Z}_q)$ ▷ Sample uniformly
9 $\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}) \leftarrow \mathsf{OServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}^{(\mathsf{sid})}_i\}_{i\in\mathsf{act}})$
10 **for** $i \in \mathsf{corr}_{\mathsf{sid}}$ **do** $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$
11 **for** $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ **do** $\mathsf{pd}_i \leftarrow \mathcal{U}(\mathbb{Z}_q)$ ▷ Sample uniformly
12 $\mathsf{pd}_h := \overline{b} - \lfloor \frac{q}{2} \rfloor \cdot \mu - \sum_{i\in\mathsf{act}\backslash\{h\}} \mathsf{pd}_i + e_{\mathsf{Sim}}$, $e_{\mathsf{Sim}} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{dec}}}$
13 **return** $\{\mathsf{pd}_i\}_{i\in\mathsf{act}}$ ▷ Constructed only with $\{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i\in\mathsf{corr}_{\mathsf{sid}}}$

$\sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ holds in Hyb_4 (as in Hyb_3). Hence, we have

$$\sum_{i\in\mathsf{act}} \mathsf{pd}_i = \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h} = \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},h}. \tag{19}$$

Furthermore, the output $(\overline{\mathbf{a}}, \overline{b})$ of $\mathsf{OServerDec}$ (in both Hyb_4 and Hyb_5) satisfies

$$\overline{b} := \overline{\mathbf{a}}^{\intercal}\mathbf{s} + \mathfrak{e} + \zeta^{(\mathsf{sid})} + \lfloor \tfrac{q}{2} \rfloor \cdot \mu + \textstyle\sum_{i\in\mathsf{act}} \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \bmod q. \tag{20}$$

Thus, by (19) and (20), we obtain $\sum_{i\in\mathsf{act}} \mathsf{pd}_i = \overline{b} - \lfloor \frac{q}{2} \rfloor \cdot \mu + \mathfrak{e} + \zeta^{(\mathsf{sid})}$. As shown in Lemma 4.3, we obtain $\mathfrak{e} + \zeta^{(\mathsf{sid})} \approx_{\mathsf{s}} D_{\mathbb{Z},\sigma_{\mathsf{dec}}}$. Thus, let $e_{\mathsf{Sim}} \leftarrow D_{\mathbb{Z},\sigma_{\mathsf{dec}}}$, then we have $\sum_{i\in\mathsf{act}} \mathsf{pd}_i \approx_{\mathsf{s}} \overline{b} - \lfloor \frac{q}{2} \rfloor \cdot \mu + e_{\mathsf{Sim}}$, and it is equivalent to $\mathsf{pd}_h \approx_{\mathsf{s}} \overline{b} - \lfloor \frac{q}{2} \rfloor \cdot \mu - \sum_{i\in\mathsf{act}\backslash\{h\}} \mathsf{pd}_i + e_{\mathsf{Sim}}$, the right hand side of which is pd_h in Hyb_5. □

Lemma 4.13 ($\mathsf{Hyb}_5 \approx \mathsf{Hyb}_6$). *The distributions of* Hyb_5 *and* Hyb_6 *are identical.*

Proof. In Hyb_5, $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},h}$ is no longer used. Thus, $\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ for $i \in \mathsf{hon}_{\mathsf{sid}}$ is used only for calculating $\mathsf{maskerr}^{(\mathsf{sid})}_i := \lambda_{\mathsf{act},i} \cdot \zeta^{(\mathsf{sid})}_i + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i}$ (line 9), and $\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ for $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ is used only for calculating $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^{\intercal}\mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i}$ (line 12). Hence, we have $\mathsf{maskerr}^{(\mathsf{sid})}_i \sim \mathcal{U}(\mathbb{Z}_q)$ for $i \in \mathsf{hon}_{\mathsf{sid}}$ and $\mathsf{pd}_i \sim \mathcal{U}(\mathbb{Z}_q)$ for $i \in \mathsf{hon}_{\mathsf{sid}} \backslash \{h\}$ in Hyb_5, which are identically distributed as in Hyb_6. □

Lemma 4.14 ($\mathsf{Hyb}_6 \approx \mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$). *The distributions of* Hyb_6 *and* $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$ *are identical.*

Proof. The only difference between Hyb_6 and $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$ is that the secret shares $\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in[N]} \leftarrow \mathsf{Shamir.Share}_{\mathbb{Z}_q,T,N}((\mathsf{sk}, \mathsf{err}))$ in Hyb_5 are replaced with

$\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in[N]} \leftarrow \mathsf{Shamir.Share}_{\mathbb{Z}_q,T,N}((\mathbf{0},\mathbf{0}))$ in $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$, which contain no information about sk or err. In Hyb_6 and $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$, only the corrupted secret shares $\{\mathsf{sk}_i, \mathsf{err}_i\}_{i\in\mathsf{corr}_{\mathsf{sid}}}$ are used, and $|\mathsf{corr}_{\mathsf{sid}}| \leq |\mathsf{corr}| < T$ holds for any sid. Hence, Hyb_6 and $\mathsf{Expt}_{\mathcal{A},\mathsf{Ideal}}$ are identically distributed owing to the privacy of Shamir secret sharing (Constr. 3.18). □

The need for semantic security of the underlying FHE. Interestingly, we do not require the semantic security of the underlying FHE for our security proof (Theorem 4.7); we use LWE assumption only for "sanitizing" ciphertext in the $\mathsf{ServerDec}$ procedure. Intuitively, this is because the partial decryption shares (and shares of padding error ζ) are masked with pseudorandom outputs of the PRF. Our security definition (Definition 4.6) keeps the encryption oracle Enc unchanged in the ideal experiment, and we only show that partial decryption shares (and shares of padding error ζ) do not reveal any information to the adversary. The semantic security of underlying FHE becomes necessary for further application of our threshold decryption procedure. For example, in MPC, we can use the semantic security of the underlying FHE to prove that no information about the plaintext input of the ciphertext of each party P_i is provided to Server (and other parties).

Security against the untrusted (semi-honest) Server. Definition 4.6 defines the security against a (semi-honest) adversary who corrupts up to $T-1$ parties. As mentioned in Section 4.2, although the Server is assumed neither to be corrupted by nor to corrupt any parties, it is untrusted (semi-honest). We can prove the security against the Server more easily than in Theorem 4.7. Although the Server receives masked errors from all parties act, $\{\mathsf{maskerr}_i := \lambda_{\mathsf{act},i}\cdot\zeta_i + \mathsf{m}^{\mathsf{row}}_{\mathsf{act},i}\}_{i\mathsf{act}}$, these values are pseudorandom to Server under the pseudorandomness assumption of PRF (as shown in Lemma 4.10). Additionally, note that the Server does not receive any partial decryption shares.

5 Our Threshold FHE Scheme from Module-LWE

Our threshold FHE scheme in the previous section is constructed from standard LWE, without relying on "yet another" variant of LWE as in [PS25]. Thus, the scheme can be naturally extended to the construction based on module-LWE. Since the construction is almost the same as our LWE-based scheme, we describe our ThFHE from MLWE (=mThFHE) in Appendix C.

However, the noise analysis, e.g., Lemma 4.3 is nontrivially different in the MLWE setting: The distribution of our padding error is changed from $D_{\mathbb{Z},\sigma_0\sqrt{B^2_{\mathsf{pub}}-\|\mathbf{s}\|^2}}$ in the LWE setting (line 3 in Algo. 2) to $D_{\mathbb{Z}^n,\sigma_0\sqrt{B^2_{\mathsf{pub}}\mathbf{I}-\mathbf{\Sigma}_{\mathbf{s}}}}$ in the MLWE setting (line 3 in Algo. 14), where $\mathbf{\Sigma}_{\mathbf{s}}$ is a covariance matrix that corresponds to the secret vector $\mathbf{s}$ over the polynomial ring.

We first provide preliminaries for MLWE in Section 5.1, and provide the noise analysis in the MLWE setting (=Lemma 5.7) in Section 5.2. Then, using

Lemma 5.7, we prove the correctness and security of mThFHE in Section 5.3 and Section 5.4, respectively.

5.1 Preliminaries related to Module-LWE

We define $\mathcal{R} = \mathbb{Z}[X]/(X^n+1)$ and $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ for n a power of 2 and $q \in \mathbb{N}$. For ease of notation, we define the *coefficient vector*, *coefficient matrix* and *coefficient Gram matrix* as follows:

Definition 5.1. *Let* $a = \sum_{i=0}^{n-1} a_i X^i \in \mathcal{R}$*, and define the coefficient vector of* a *as* $\mathbf{a} := \mathsf{vec}(a) := (a_0, a_1, \ldots, a_{n-1})^\intercal \in \mathbb{Z}^n$*. Let* $\mathbf{P} := \left(\begin{array}{c|c} \mathbf{0} & -1 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right) \in \mathbb{Z}^{n\times n}$ *be a (negacyclic) permutation matrix, and define the coefficient matrix of* a *as* $\mathbf{A} := \mathsf{mat}(a) := \left(\mathbf{a}\ \mathbf{Pa} \cdots \mathbf{P}^{n-1}\mathbf{a}\right) \in \mathbb{Z}^{n\times n}$*. The coefficient Gram matrix of* a *is defined as* $\mathbf{\Sigma}_a := \mathsf{Gram}(a) := \mathbf{A}\mathbf{A}^\intercal \in \mathbb{Z}^{n\times n}$*.*

For any $a \in \mathcal{R}$, we define $\|a\| := \|\mathsf{vec}(a)\|$ and $\|a\|_\infty := \|\mathsf{vec}(a)\|_\infty$. For the distribution χ over $\mathbb{Z}^n$, we define $\mathcal{R}(\chi) := \{a \in \mathcal{R} \mid \mathsf{vec}(a) \sim \chi\}$. The randomized Gaussian rounding $\lfloor\cdot\rceil_\sigma$ (Definition 3.16) naturally extends to vectors coefficient-wise. The coefficient matrix is useful for describing a product:

Fact 5.2. *For* $r, e \in \mathcal{R}$*, we have* $\mathsf{vec}(re) = \mathbf{Re} = \mathbf{Er}$*, where* $\mathbf{R} := \mathsf{mat}(r)$*,* $\mathbf{r} := \mathsf{vec}(r)$*,* $\mathbf{E} := \mathsf{mat}(e)$ *and* $\mathbf{e} := \mathsf{vec}(e)$*.*

From the structure of $\mathcal{R}$, we obtain the following useful facts:

Fact 5.3. *For any* $a \in \mathcal{R}$*,* $\|\mathsf{mat}(a)\|_{\mathrm{len}} = \|\mathsf{vec}(a)\|$*.*

Fact 5.4. *For any* $a \neq 0 \in \mathcal{R}$*,* $\mathsf{mat}(a)$ *is nonsingular.*

The module-LWE problem is defined as follows:

Definition 5.5 (Module-LWE). *Let* $k, m, q \in \mathbb{N}$ *and* χ *be a distribution over* $\mathcal{R}_q$*. We define the module-LWE distribution as follows:* $\mathsf{MLWE}(k, m, q, \chi) := \{(\mathbf{A}, \mathbf{b} := \mathbf{As} + \mathbf{e}) \mid \mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{m\times k}, \mathbf{s} \leftarrow \chi^k, \mathbf{e} \leftarrow \chi^m\}$*. The advantage of an algorithm* $\mathcal{A}$ *for solving* d-MLWE *is defined as* $\mathrm{Adv}^{\mathsf{d\text{-}MLWE}}_{\mathcal{A}} = |\Pr[\mathcal{A}(\mathbf{A}, \mathbf{As}+\mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u} \leftarrow \mathcal{U}(\mathcal{R}_q^k)) = 1]|$*.*

The smudging lemma (Lemma 3.15) is extended to multivariate Gaussian:

Corollary 5.6. *Let* $n = \mathrm{poly}(\kappa)$*,* $\mathbf{c} \in \mathbb{Z}^n$ *and* $\sigma = \Omega(\|\mathbf{c}\|_\infty 2^\kappa)$*. Then, we have* $D_{\mathbb{Z}^n,\sigma,\mathbf{c}} \approx_{\mathsf{s}} D_{\mathbb{Z}^n,\sigma}$*.*

5.2 Noise Analysis

We prove the counterpart of Lemma 4.3 in the MLWE setting, which is required for the proof of correctness and security of our ThFHE from MLWE.

Lemma 5.7 (Noise analysis). *Let χ_{mlwe} be a B_{mlwe}-bounded distribution over $\mathbb{Z}$ s.t. $\mathbf{s} := (s_1, \ldots, s_k) \leftarrow \mathcal{R}(\chi_{\mathsf{mlwe}}^n)^k$ satisfies $\min_{i\in[k]} \sigma_{\min}(\mathbf{S}_i) \geq \frac{1}{c}$ for some $c > 0$ with overwhelming probability, where $\mathbf{S}_i := \mathsf{mat}(s_i)$ (c.f. Definition 5.1). Let $\epsilon := \mathrm{negl}(\kappa)$, $p \in \mathbb{N}$, $\sigma_{\mathsf{flood}} \geq p \cdot \sqrt{2n} B_{\mathsf{mlwe}} \eta_\epsilon^+(\mathbb{Z}^n)$, $\sigma_0 \geq c \cdot \sqrt{2n} B_{\mathsf{mlwe}} \eta_\epsilon^+(\mathbb{Z}^n)$, $\sigma_1 \geq \eta_\epsilon(\mathbb{Z}^n)$. Let $\mathfrak{E} \leftarrow \mathcal{R}(D_{\mathbb{Z}^n, \sigma_{\mathsf{flood}}})$, $\mathbf{u} \in \mathcal{R}^k$, and $\mathbf{r} := \mathbf{u} - p\lfloor \frac{1}{p}\mathbf{u} \rceil_{\sigma_0}$, and define $\mathbf{\Sigma}_{\mathbf{s}} := \sum_{i\in[k]} \mathbf{\Sigma}_{s_i}$, where $\mathbf{\Sigma}_{s_i} := \mathsf{Gram}(s_i)$. Then, we have*

$$\mathfrak{e} := \lfloor \tfrac{1}{p}(\mathbf{r}\cdot\mathbf{s} + \mathfrak{E}) \rceil_{\sigma_1} \approx_{\mathsf{s}} \mathcal{R}(D_{\mathbb{Z}^n, \sqrt{\sigma_0^2 \mathbf{\Sigma}_{\mathbf{s}} + ((\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2)\mathbf{I}}}). \tag{21}$$

Further, for $B_{\mathsf{pub}} > \sqrt{kn^2 B_{\mathsf{mlwe}}^2 + 1}$ and $\zeta \leftarrow \mathcal{R}(D_{\mathbb{Z}^n, \sigma_0\sqrt{B_{\mathsf{pub}}^2 \mathbf{I} - \mathbf{\Sigma}_{\mathbf{s}}}})$, we have

$$\mathfrak{e} + \zeta \approx_{\mathsf{s}} \mathcal{R}(D_{\mathbb{Z}^n, \sigma_{\mathsf{dec}}}), \quad \text{where } \sigma_{\mathsf{dec}} := \sqrt{\sigma_0^2 B_{\mathsf{pub}}^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}. \tag{22}$$

Proof. For any $\mathbf{x} \in \mathcal{R}^k$, we denote the i-th elements of $\mathbf{x}$ by $x_i \in \mathcal{R}$. Furthermore, we denote the j-th coefficient of x_i by $x_{i,j} \in \mathbb{Z}$. Since $\lfloor \cdot \rceil_{\sigma_0}$ is performed element- and coefficient-wise, we have $r_{i,j} = u_{i,j} - p \cdot \lfloor \frac{1}{p} u_{i,j} \rceil_{\sigma_0}$. Thus, $r_{i,j} \sim D_{c_{i,j} + p\mathbb{Z}, p\sigma_0}$, where $c_{i,j} := p\{\frac{1}{p} u_{i,j}\}$. Note that $c_{i,j} \in p \cdot (\frac{1}{p}\mathbb{Z}) = \mathbb{Z}$ is an integer. Now, we also have $r_i \sim \mathcal{R}(D_{\mathbf{c}_i + p\mathbb{Z}^n, p\sigma_0})$, where $\mathbf{c}_i := (c_{i,1}, \ldots, c_{i,n}) \in \mathbb{Z}^n$. Thus, we have $r_i s_i \sim \mathcal{R}(\mathbf{S}_i \cdot D_{\mathbf{c}_i + p\mathbb{Z}^n, p\sigma_0})$ by Fact 5.2. Furthermore, we have

$$r_i s_i \sim \mathcal{R}(D_{A_i, p\sigma_0 \mathbf{S}_i}) \quad \text{where} \quad A_i := \mathbf{S}_i \cdot \mathbf{c}_i + p\mathbf{S}_i \cdot \mathbb{Z}^n,$$

for any $i \in [k]$, by Lemma 3.13 and Fact 5.4. Note that the coset A_i satisfies $A_i \subseteq \mathbb{Z}^n$ since $\mathbf{S}_i$ and $\mathbf{c}_i$ are an integer matrix and vector.

We now analyze the distribution of $\mathbf{r}^\intercal \mathbf{s} + \mathfrak{E} = \sum_{i=1}^n r_i s_i + \mathfrak{E} \in \mathcal{R}$. Let $\widehat{\sigma} := \min(\sigma_{\mathsf{flood}}/p, \sigma_0/c)$, then we have $\widehat{\sigma} \geq \sqrt{2n} B_{\mathsf{mlwe}} \eta_\epsilon^+(\mathbb{Z}^n)$ by the hypothesis. Hence, we have $\min(\sigma_{\mathsf{flood}}, \min_{i\in[k]} \sigma_{\min}(p\sigma_0 \mathbf{S}_i)) = p\widehat{\sigma} \geq \sqrt{2n} p B_{\mathsf{mlwe}} \eta_\epsilon^+(\mathbb{Z}^n) \geq \sqrt{2} \max_{i\in[k]} \|p\mathbf{S}_i\|_{\mathrm{len}} \eta_\epsilon^+(\mathbb{Z}^n)$, where we use the fact that $\sqrt{n}B \geq \max_{i\in[k]} \|\mathbf{S}_i\|_{\mathrm{len}}$ holds by Fact 5.3. Therefore, by the subsequent Lemma 5.9, we obtain

$$\mathbf{r}^\intercal \mathbf{s} + \mathfrak{E} \approx_{\mathsf{s}} \mathcal{R}(D_{\mathbb{Z}^n, \sqrt{\mathbf{\Sigma}}}), \text{ where } \mathbf{\Sigma} := p^2 \sigma_0^2 \mathbf{\Sigma}_{\mathbf{s}} + \sigma_{\mathsf{flood}}^2 \mathbf{I}.$$

Then, by Lemma 3.13 we also have $\mathfrak{e}' := \frac{1}{p}(\mathbf{r}^\intercal \mathbf{s} + \mathfrak{E}) \approx_{\mathsf{s}} \mathcal{R}(D_{\frac{1}{p}\mathbb{Z}^n, \frac{1}{p}\sqrt{\mathbf{\Sigma}}})$. Finally, we obtain Eq. (21), by Lemma 3.11, if the conditions $\sigma_1 \geq \eta_\epsilon(\mathbb{Z}^n)$ and $(p^2 \mathbf{\Sigma}^{-1} + \sigma_1^{-2})^{-1/2} \geq \eta_\epsilon(\frac{1}{p}\mathbb{Z}^n)$ are satisfied. By Fact 5.8, we have

$$\sigma_{\min}((p^2 \mathbf{\Sigma}^{-1} + \sigma_1^{-2})^{-1/2}) \geq \tfrac{1}{\sqrt{2}} \min\{\tfrac{1}{p}\sigma_{\min}(\sqrt{\mathbf{\Sigma}}), \sigma_1\} \geq \tfrac{1}{\sqrt{2}} \min\{\widehat{\sigma}, \sigma_1\}$$

and $\widehat{\sigma} \geq \sqrt{2n} B_{\mathsf{mlwe}} \eta_\epsilon(\mathbb{Z}^n) (\geq \eta_\epsilon(\frac{1}{p}\mathbb{Z}^n))$ holds by the hypothesis. Hence, $(p^2 \mathbf{\Sigma}^{-1} + \sigma_1^{-2})^{-1/2} \geq \eta_\epsilon(\frac{1}{p}\mathbb{Z}^n)$ holds by Fact 3.9. Thus, we only require $\sigma_1 \geq \eta_\epsilon(\mathbb{Z}^n)$ to obtain Eq. (21).

Next, we prove Eq. (22). To simplify the notation, let $\mathbf{\Sigma}_{\mathfrak{e}} := \frac{1}{p^2}\mathbf{\Sigma} + \sigma_1^2 \mathbf{I}$ and $\mathbf{\Sigma}_\zeta := B_{\mathsf{pub}}^2 \mathbf{I} - \mathbf{\Sigma}_{\mathbf{s}}$, then we have $\mathfrak{e} \approx_{\mathsf{s}} \mathcal{R}(D_{\mathbb{Z}^n, \sqrt{\mathbf{\Sigma}_{\mathfrak{e}}}})$ and $\zeta \sim \mathcal{R}(D_{\mathbb{Z}^n, \sigma_0\sqrt{\mathbf{\Sigma}_\zeta}})$. By

Corollary 3.12, $\mathfrak{e}+\zeta \approx_{\mathsf{s}} \mathcal{R}(D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_{\mathfrak{e}}+\sigma_0^2\mathbf{\Sigma}_\zeta}})$, i.e., Eq. (22), holds if $\sqrt{\mathbf{\Sigma}_{\mathfrak{e}}} \geq \eta_\epsilon(\mathbb{Z}^n)$ and $(\mathbf{\Sigma}_{\mathfrak{e}}^{-1} + (\sigma_0^2\mathbf{\Sigma}_\zeta)^{-1})^{-1/2} \geq \eta_\epsilon(\mathbb{Z}^n)$. By Fact 5.8 and the hypothesis, we have

$$\sigma_{\min}(\sqrt{\mathbf{\Sigma}_{\mathfrak{e}}}) \geq \sqrt{2}\min(\tfrac{1}{p}\sigma_{\min}(\sqrt{\mathbf{\Sigma}}), \sigma_1) \geq \sqrt{2}\min(\widehat{\sigma}, \sigma_1) \geq 2\sqrt{n}B_{\mathsf{mlwe}}\eta_\epsilon(\mathbb{Z}^n),$$

thus, $\sqrt{\mathbf{\Sigma}_{\mathfrak{e}}} \geq \eta_\epsilon(\mathbb{Z}^n)$ holds by Fact 3.9. Again by Fact 5.8, we have

$$\sigma_{\min}((\mathbf{\Sigma}_{\mathfrak{e}}^{-1} + (\sigma_0^2\mathbf{\Sigma}_\zeta)^{-1})^{-1/2}) \geq \tfrac{1}{\sqrt{2}}\min(\sigma_{\min}(\sqrt{\mathbf{\Sigma}_{\mathfrak{e}}}), \sigma_0\sigma_{\min}(\sqrt{\mathbf{\Sigma}_\zeta})).$$

Hence, we only need $\sigma_0\sigma_{\min}(\sqrt{\mathbf{\Sigma}_\zeta}) \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^n)$. Since $\sigma_0 \geq \sqrt{2}\eta_\epsilon(\mathbb{Z}^n)$ by the hypothesis, we only need $\sigma_{\min}(\sqrt{\mathbf{\Sigma}_\zeta}) \geq 1$. To prove this, it is sufficient to show $\mathbf{\Sigma}_\zeta - \mathbf{I} = (B_{\mathsf{pub}}^2 - 1)\mathbf{I} - \sum_{i\in[k]} \mathbf{S}_i\mathbf{S}_i^\intercal \succ 0$. Note that the absolute value of every elements of $\mathbf{S}_i\mathbf{S}_i^\intercal$ is $\leq n\cdot\|\mathsf{vec}(s_i)\|^2 \leq n^2B_{\mathsf{mlwe}}^2$ by Fact 5.3. Hence, $\mathbf{\Sigma}_\zeta - \mathbf{I}$ is strictly diagonally dominant since $(B_{\mathsf{pub}}^2 - 1) - kn^2B_{\mathsf{mlwe}}^2 > 0$ by the hypothesis. Further, it implies $\mathbf{\Sigma}_\zeta - \mathbf{I} \succ 0$ (c.f., [HJ85, Thm. 6.1.10]). Thus, we have $\sigma_{\min}(\sqrt{\mathbf{\Sigma}_\zeta}) = \sqrt{\lambda_{\min}((\mathbf{\Sigma}_\zeta - \mathbf{I}) + \mathbf{I})} \geq \sqrt{\lambda_{\min}(\mathbf{\Sigma}_\zeta - \mathbf{I}) + 1} \geq 1$. □

We complete the proof by describing the deferred Fact 5.8 and Lemma 5.9:

Fact 5.8. *For any $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2 \succ 0$, we have $\sigma_{\min}((\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1/2}) \geq \frac{1}{\sqrt{2}}\min\{\sigma_{\min}(\sqrt{\mathbf{\Sigma}_1}), \sigma_{\min}(\sqrt{\mathbf{\Sigma}_2})\}$.*

Lemma 5.9 (Adapted from [OT25, Lemma 5.4]). *Let $\epsilon = \mathrm{negl}(\kappa)$. Let $\mathbf{\Sigma}_0, \ldots, \mathbf{\Sigma}_k \in \mathbb{R}^{n\times n}$ be positive definite matrices. Let $A_1, \ldots, A_k \subseteq \mathbb{Z}^n$ be cosets of full-rank integer lattices $\mathcal{L}_1(\mathbf{B}_1), \ldots, \mathcal{L}_k(\mathbf{B}_k)$ with (nonsingular) basis $\mathbf{B}_1, \ldots, \mathbf{B}_k$. Let $\widehat{\sigma} := \min_{i\in\{0,\ldots,k\}} \sigma_{\min}(\sqrt{\mathbf{\Sigma}_i})$ and $\widehat{B} := \max_{i\in[k]}\|\mathbf{B}_i\|_{\mathrm{len}}$. If $\widehat{\sigma} \geq \sqrt{2}\widehat{B}\eta_\epsilon^+(\mathbb{Z}^n)$, we have*

$$\textstyle\sum_{i=1}^k D_{A_i,\sqrt{\mathbf{\Sigma}_i}} + D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_0}} \approx_{\mathsf{s}} D_{\mathbb{Z}^n,\sqrt{\sum_{i=0}^k \mathbf{\Sigma}_i}}. \tag{23}$$

Proof. The proof is almost identical to that of [OT25, Lemma 5.4]. We provide the proof in Appendix B for the completeness of this paper. □

5.3 Correctness

Since the syntax is identical, the correctness of our MLWE-based ThFHE (Algo. 14) is also defined as in Definition 4.4. We prove the correctness:

Theorem 5.10 (Correctness). *Assume parameters $\epsilon, p, \chi_{\mathsf{mlwe}}, B_{\mathsf{mlwe}}, B_{\mathsf{pub}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ are selected as in Lemma 5.7. Furthermore, assume $\sigma_{\mathsf{flood}} = \Omega(2^\kappa B_{\mathsf{eval}})$ (B_{eval} is defined as in Algo. 13), $Q = p\cdot q = \sigma_{\mathsf{flood}}\cdot\Omega(\sqrt{\kappa})$, and $q = \sigma_{\mathsf{dec}}\cdot\Omega(\sqrt{\kappa})$, where $\sigma_{\mathsf{dec}} := \sqrt{(\sigma_0 B_{\mathsf{pub}})^2 + (\sigma_{\mathsf{flood}}/p)^2 + \sigma_1^2}$. Then, mThFHE (Algo. 14) is correct.*

Proof. The proof is essentially identical to Theorem 4.5, except for that we now rely on the core analysis in the MLWE setting (=Lemma 5.7) and multivariate version of the smudging lemma (=Corollary 5.6), instead of Lemma 4.3 and Lemma 3.15, respectively. □

5.4 Security

The security of our mThFHE (Algo. 14) is almost identically defined as Definition 4.6, by replacing variables over $\mathbb{Z}_q^n$ with the corresponding variables over R_q^k. The security can be proved by MLWE assumption as follows:

Theorem 5.11 (Security). mThFHE *(Algo. 14) is simulation-secure under the assumption on the pseudorandomness of* PRF *and* d-MLWE$(n, m, Q, \chi_{\mathsf{lwe}})$*: For any PPT algorithms* $\mathcal{D}, \mathcal{A}$*, there exists PPT algorithms* Sim*,* $\mathcal{B}_{\mathsf{LWE}}$ *and* $\mathcal{B}_{\mathsf{PRF}}$ *s.t.*

$$\mathrm{Adv}^{\mathsf{ThFHE}}_{\mathcal{D},\mathcal{A}}(\kappa) < \mathrm{Adv}^{\mathsf{d\text{-}MLWE}(m,k,Q,\mathcal{R}(\chi^n_{\mathsf{mlwe}}))}_{\mathcal{B}_{\mathsf{LWE}}} + \mathrm{Adv}^{\mathsf{PRF}}_{\mathcal{B}_{\mathsf{PRF}}}(\kappa) + \mathrm{negl}(\kappa),$$

if $\epsilon, p, q, Q, B_{\mathsf{mlwe}}, B_{\mathsf{pub}}, \chi_{\mathsf{mlwe}}, \sigma_{\mathsf{flood}}, \sigma_0, \sigma_1$ *are selected as in Theorem 5.10.*

Proof. The proof is almost identical to that of Theorem 4.7, except for that we rely on Lemma 5.7 instead of Lemma 4.3. □

References

[ARS+15] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, M. Zohner. "Ciphers for MPC and FHE". *EUROCRYPT 2015*. 2015, pp. 430–454. https://doi.org/10.1007/978-3-662-46800-5_17.

[AJL+12] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, D. Wichs. "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE". *EUROCRYPT 2012*. 2012, pp. 483–501. https://doi.org/10.1007/978-3-642-29011-4_29.

[AJW11] G. Asharov, A. Jain, D. Wichs. *Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE*. ePrint 2011/613. 2011. https://eprint.iacr.org/2011/613. Full version of [AJL+12].

[BJMS20] S. Badrinarayanan, A. Jain, N. Manohar, A. Sahai. "Secure MPC: Laziness Leads to GOD". *ASIACRYPT 2020*. 2020, pp. 120–150. https://doi.org/10.1007/978-3-030-64840-4_5.

[BCK+23] Y. Bae, J. H. Cheon, J. Kim, J. H. Park, D. Stehlé. "HERMES: Efficient Ring Packing Using MLWE Ciphertexts and Application to Transciphering". *CRYPTO 2023*. 2023, pp. 37–69. https://doi.org/10.1007/978-3-031-38551-3_2.

[BGG+18] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, A. Sahai. "Threshold Cryptosystems from Threshold Fully Homomorphic Encryption". *CRYPTO 2018*. 2018, pp. 565–596. https://doi.org/10.1007/978-3-319-96884-1_19.

[BS23] K. Boudgoust, P. Scholl. "Simple Threshold (Fully Homomorphic) Encryption from LWE with Polynomial Modulus". *ASIACRYPT 2023*. 2023, pp. 371–404. https://doi.org/10.1007/978-981-99-8721-4_12.

[BS24] K. Boudgoust, P. Scholl. *Simple Threshold (Fully Homomorphic) Encryption From LWE With Polynomial Modulus*. ePrint 2023/016. 2024. https://eprint.iacr.org/2023/016. Version 2024-07-16.

[BGV12] Z. Brakerski, C. Gentry, V. Vaikuntanathan. "(Leveled) Fully Homomorphic Encryption without Bootstrapping". *ITCS 2012*. ACM, 2012, pp. 309–325. https://doi.org/10.1145/2090236.2090262.

[BP23] L. T. Brandão, R. Peralta. *NIST IR 8214C ipd: NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft)*. 2023. https://doi.org/10.6028/NIST.IR.8214C.ipd (visited on Dec. 11, 2023).

[CHK+18] J. H. Cheon, K. Han, A. Kim, M. Kim, Y. Song. "Bootstrapping for Approximate Homomorphic Encryption". *EUROCRYPT 2018*. 2018, pp. 360–384. https://doi.org/10.1007/978-3-319-78381-9_14.

[CKKS17] J. H. Cheon, A. Kim, M. Kim, Y. Song. "Homomorphic Encryption for Arithmetic of Approximate Numbers". *ASIACRYPT 2017*. 2017, pp. 409–437. https://doi.org/10.1007/978-3-319-70694-8_15.

[CGGI20] I. Chillotti, N. Gama, M. Georgieva, M. Izabachène. "TFHE: Fast Fully Homomorphic Encryption Over the Torus". *J. Cryptol.* 33.1 (2020), pp. 34–91. https://doi.org/10.1007/s00145-019-09319-x.

[DEG+18] C. Dobraunig, M. Eichlseder, L. Grassi, V. Lallemand, G. Leander, E. List, F. Mendel, C. Rechberger. "Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit". *CRYPTO 2018*. 2018, pp. 662–692. https://doi.org/10.1007/978-3-319-96884-1_22.

[GGHR14] S. Garg, C. Gentry, S. Halevi, M. Raykova. "Two-Round Secure MPC from Indistinguishability Obfuscation". *TCC 2014*. 2014, pp. 74–94. https://doi.org/10.1007/978-3-642-54242-8_4.

[GMPW20] N. Genise, D. Micciancio, C. Peikert, M. Walter. "Improved Discrete Gaussian and SubGaussian Analysis for Lattice Cryptography". *PKC 2020*. 2020, pp. 623–651. https://doi.org/10.1007/978-3-030-45374-9_21.

[Gen09] C. Gentry. "Fully Homomorphic Encryption Using Ideal Lattices". *STOC '09*. ACM, 2009, pp. 169–178. https://doi.org/10.1145/1536414.1536440.

[GPV08] C. Gentry, C. Peikert, V. Vaikuntanathan. "Trapdoors for Hard Lattices and New Cryptographic Constructions". *STOC '08*. ACM, 2008, pp. 197–206. https://doi.org/10.1145/1374376.1374407.

[GSW13] C. Gentry, A. Sahai, B. Waters. "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based". *CRYPTO 2013*. 2013, pp. 75–92. https://doi.org/10.1007/978-3-642-40041-4_5.

[GLS15] S. Dov Gordon, F.-H. Liu, E. Shi. "Constant-Round MPC with Fairness and Guarantee of Output Delivery". *CRYPTO 2015*. 2015, pp. 63–82. https://doi.org/10.1007/978-3-662-48000-7_4.

[HKL+22] J. Ha, S. Kim, B. Lee, J. Lee, M. Son. "Rubato: Noisy Ciphers for Approximate Homomorphic Encryption". *EUROCRYPT 2022*. 2022, pp. 581–610. https://doi.org/10.1007/978-3-031-06944-4_20.

[HJ85] R. A. Horn, C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. https://doi.org/10.1017/CBO9780511810817.

[LPR10] V. Lyubashevsky, C. Peikert, O. Regev. "On Ideal Lattices and Learning with Errors over Rings". *EUROCRYPT 2010*. 2010, pp. 1–23. https://doi.org/10.1007/978-3-642-13190-5_1.

[MP13] D. Micciancio, C. Peikert. "Hardness of SIS and LWE with Small Parameters". *CRYPTO 2013*. 2013, pp. 21–39. https://doi.org/10.1007/978-3-642-40041-4_2.

[MR07] D. Micciancio, O. Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". *SIAM J. Comput.* 37.1 (2007), pp. 267–302. https://doi.org/10.1137/S0097539705447360.

[MS23] D. Micciancio, A. Suhl. *Simulation-Secure Threshold PKE from LWE with Polynomial Modulus*. ePrint 2023/1728. 2023. https://eprint.iacr.org/2023/1728.

[NLV11] M. Naehrig, K. Lauter, V. Vaikuntanathan. "Can homomorphic encryption be practical?" *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*. CCSW '11. ACM, 2011, pp. 113–124. https://doi.org/10.1145/2046660.2046682.

[OT25] H. Okada, T. Takagi. "Gram Root Decomposition over the Polynomial Ring: Application to Sphericalization of Discrete Gaussian". *ICISSP 2025*. 2025.

[PS25] A. Passelègue, D. Stehlé. "Low Communication Threshold Fully Homomorphic Encryption". *ASIACRYPT 2024*. 2025, pp. 297–329. https://doi.org/10.1007/978-981-96-0875-1_10.

[Pei10] C. Peikert. "An Efficient and Parallel Gaussian Sampler for Lattices". *CRYPTO 2010*. 2010, pp. 80–97. https://doi.org/10.1007/978-3-642-14623-7_5.

[PKM+24a] R. del Pino, S. Katsumata, M. Maller, F. Mouhartem, T. Prest, M.-J. Saarinen. "Threshold Raccoon: Practical Threshold Signatures from Standard Lattice Assumptions". *EUROCRYPT 2024*. 2024, pp. 219–248. https://doi.org/10.1007/978-3-031-58723-8_8.

[PKM+24b] R. del Pino, S. Katsumata, M. Maller, F. Mouhartem, T. Prest, M.-J. Saarinen. *Threshold Raccoon: Practical Threshold Signatures from Standard Lattice Assumptions*. Cryptology ePrint Archive, Paper 2024/184. 2024. https://eprint.iacr.org/2024/184. Full version of [PKM+24a].

[Reg09] O. Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". *J. ACM* 56.6 (2009). https://doi.org/

10.1145/1568318.1568324. Preliminary version appeared in STOC '05.

[Sha79] A. Shamir. "How to share a secret". *Commun. ACM* 22.11 (1979), pp. 612–613. https://doi.org/10.1145/359168.359176.

A Definition of Pseudorandom Function

We rely on the multi-instance variant of the pseudorandom functions PRF given in [PKM+24a]. Here, we refer to the definition for the completeness.

Definition A.1 (Pseudorandom function). *Let κ be a security parameter and $n, l \in \mathbb{N}$. We say that a deterministic PPT algorithm* $\mathsf{PRF} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^l$ *is pseudorandom function if for any PPT algorithm $\mathcal{A}$ we have*

$$\mathrm{Adv}^{\mathsf{PRF}}_{\mathcal{A}}(\kappa) = |\Pr[\mathsf{Game}^{\mathsf{PRF}}_{\mathcal{A}}(\kappa) = 1] - \tfrac{1}{2}| = \mathrm{negl}(\kappa),$$

where $\mathsf{Game}^{\mathsf{PRF}}_{\mathcal{A}}$ *is defined as in Algo. 12.*

Algorithm 12: Security game of PRF.

$\underline{\mathsf{Game}^{\mathsf{PRF}}_{\mathcal{A}}(\kappa)}$:	$\underline{\mathsf{O}_{\mathsf{PRF}}(i, x \in \{0,1\}^n)}$:
1 $\mathsf{List}[\cdot] := \emptyset$	6 **if** $\mathsf{List}[i] = \emptyset$ **then**
2 $b \leftarrow \{0,1\}$	$\mathsf{seed} \leftarrow \{0,1\}^\kappa$, $\mathsf{List}[i] \leftarrow \mathsf{seed}$
3 $b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{PRF}}}(\kappa)$	7 $\mathsf{seed} \leftarrow \mathsf{List}[i]$
4 **if** $b = b'$ **then return** 1	8 $y_0 \leftarrow \mathcal{U}(\{0,1\}^l)$, $y_1 \leftarrow \mathsf{PRF}(\mathsf{seed}, x)$
5 **else return** 0	9 **return** y_b

B Proof of Lemma 5.9

Although the proof of Lemma 5.9 is almost identical to that of [OT25, Lemma 5.4], we provide the proof in this section for the completeness:

Proof. By using Corollary 3.12, we first show

$$D_{A_1,\sqrt{\mathbf{\Sigma}_1}} + D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_0}} \approx_{\mathsf{s}} D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_0+\mathbf{\Sigma}_1}}. \tag{24}$$

We obtain $\sqrt{\mathbf{\Sigma}_0} \geq \eta_\epsilon(\mathbb{Z}^n)$ by Fact 3.9 and the hypothesis $\sigma_{\min}(\sqrt{\mathbf{\Sigma}_0}) \geq \widehat{\sigma} \geq \sqrt{2}\widehat{B}\eta^+_\epsilon(\mathbb{Z}^n) \geq \eta^+_\epsilon(\mathbb{Z}^n)$. We have $\sqrt{(\mathbf{\Sigma}_0^{-1} + \mathbf{\Sigma}_1^{-1})^{-1}} \geq \eta_\epsilon(\mathcal{L}_1(\mathbf{B}_1))$ because

$$\begin{aligned}\sigma_{\min}(\sqrt{(\mathbf{\Sigma}_0^{-1} + \mathbf{\Sigma}_1^{-1})^{-1}}) &\geq \tfrac{1}{\sqrt{2}} \min\{\sigma_{\min}(\sqrt{\mathbf{\Sigma}_0}), \sigma_{\min}(\sqrt{\mathbf{\Sigma}_1})\} \\ &\geq \tfrac{1}{\sqrt{2}}\widehat{\sigma} \geq \|\mathbf{B}_1\|_{\mathrm{len}}\eta^+_\epsilon(\mathbb{Z}^n)\end{aligned}$$

by Fact 5.8 and the hypothesis $\widehat{\sigma} \geq \sqrt{2}\widehat{B}\eta^+_\epsilon(\mathbb{Z}^n) \geq \sqrt{2}\|\mathbf{B}_1\|_{\mathrm{len}}\eta^+_\epsilon(\mathbb{Z}^n)$. Therefore, we obtain Eq. (24) by Corollary 3.12 since $A_1 \subseteq \mathbb{Z}^n$. Next, we show

$$D_{A_2,\sqrt{\mathbf{\Sigma}_2}} + D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_0+\mathbf{\Sigma}_1}} \approx_{\mathsf{s}} D_{\mathbb{Z}^n,\sqrt{\mathbf{\Sigma}_0+\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}} \tag{25}$$

Algorithm 13: Module-LWE-based underlying FHE scheme $\mathsf{mFHE} := (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$

$\underline{\mathsf{PP}(1^\kappa, 1^N) \to \mathsf{pp}}$:
return $\mathsf{pp} := (n, k, m, Q, \chi_{\mathsf{mlwe}}, B_{\mathsf{eval}})$ ▷ χ_{mlwe}: distribution over $\mathbb{Z}$

$\underline{\mathsf{KeyGen}()}$:
$\mathsf{sk} := \mathbf{s} \leftarrow \mathcal{R}(\chi_{\mathsf{mlwe}}^n)^k$, $\mathsf{pk} := (\mathbf{A}, \mathbf{b} := \mathbf{As} + \mathbf{e}) \sim \mathsf{MLWE}(k, m, Q, \mathcal{R}(\chi_{\mathsf{mlwe}}^n))$
Generate the evaluation key evk ▷ We omit the details of evk
return $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}(\mathsf{pk}, \mu \in \mathcal{M} := \mathcal{R}_2)}$:
$\mathbf{r} \leftarrow \mathcal{R}(\chi_{\mathsf{mlwe}}^n)^m, \mathbf{f} \leftarrow \mathcal{R}(\chi_{\mathsf{mlwe}}^n)^k, f \leftarrow \mathcal{R}(\chi_{\mathsf{mlwe}}^n)$
return $\mathsf{ct} := (\mathbf{a}, b) := (\mathbf{r}^\intercal \mathbf{A} + \mathbf{f}, \mathbf{r}^\intercal \mathbf{b} + f + \lfloor \frac{Q}{2} \rfloor \cdot \mu) \in \mathcal{R}_Q^{k+1}$
Note: PP outputs pp s.t. the noise in the ciphertext $e_{\mathsf{ct}} := b - \mathbf{a}^\intercal \mathbf{s} - \lfloor \frac{Q}{2} \rfloor \cdot \mu$ satisfies $\|e_{\mathsf{ct}}\|_\infty < B_{\mathsf{eval}}$ with overwhelming probability

$\underline{\mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)}$:
return $\overline{\mathsf{ct}} := (\overline{\mathbf{a}}, \overline{b})$ s.t. $e_{\mathsf{eval}} := \overline{b} - \overline{\mathbf{a}}^\intercal \mathbf{s} - \lfloor \frac{Q}{2} \rfloor \cdot C(\mu_1, \ldots, \mu_l)$ satisfies $\|e_{\mathsf{eval}}\|_\infty < B_{\mathsf{eval}}$ for any C, where $\mu_1, \ldots, \mu_l$ are the plaintexts of $\mathsf{ct}_1, \ldots, \mathsf{ct}_l$

$\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ct} := (\mathbf{a}, b)) \to \overline{\mu} \in \mathcal{M}}$:
return $\overline{\mu} := \lfloor (b - \mathbf{a}^\intercal \mathbf{s}) / \lfloor \frac{Q}{2} \rfloor \rceil$

via Corollary 3.12 again. By Fact 3.9, Fact 5.8 and the hypothesis $\eta_\epsilon^+(\mathbb{Z}^n) \le \widehat{\sigma}$, we have $\sqrt{\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1} \ge \eta_\epsilon(\mathbb{Z}^n)$ because we have

$$\sigma_{\min}(\sqrt{\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1}) \ge \min\{\sigma_{\min}(\sqrt{\mathbf{\Sigma}_0}), \sigma_{\min}(\sqrt{\mathbf{\Sigma}_1})\} \ge \widehat{\sigma} \ge \eta_\epsilon^+(\mathbb{Z}^n).$$

Furthermore, we have $\sqrt{((\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}} \ge \eta_\epsilon(\mathcal{L}_1(\mathbf{B}_2))$ because

$$\begin{aligned}
\sigma_{\min}(\sqrt{((\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}}) &\ge \tfrac{1}{\sqrt{2}} \min\{\sigma_{\min}(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1), \sigma_{\min}(\mathbf{\Sigma}_2)\} \\
&\ge \tfrac{1}{\sqrt{2}} \min\{\sigma_{\min}(\mathbf{\Sigma}_0), \sigma_{\min}(\mathbf{\Sigma}_1), \sigma_{\min}(\mathbf{\Sigma}_2)\} \\
&\ge \tfrac{1}{\sqrt{2}} \widehat{\sigma} \ge \|\mathbf{B}_2\|_{\mathrm{len}} \eta_\epsilon^+(\mathbb{Z}^n)
\end{aligned}$$

holds by the hypothesis $\widehat{\sigma} \ge \sqrt{2}\widehat{B}\eta_\epsilon^+(\mathbb{Z}^n) \ge \sqrt{2}\|\mathbf{B}_2\|_{\mathrm{len}}\eta_\epsilon^+(\mathbb{Z}^n)$. Hence, we obtain Eq. (25) by Corollary 3.12. Repeating the above, we obtain Eq. (23). □

C Omitted Construction of our MLWE-based ThFHE

In this section, we extend our LWE-based threshold FHE presented in Section 4 into the construction based on module-LWE.

We first describe underlying FHE scheme based on MLWE in Algo. 13. The syntax of FHE is the same as Definition 4.1. Then, we present our MLWE-based threshold FHE scheme in Algo. 14.

Algorithm 14: Our threshold FHE from Module-LWE: $\mathsf{mThFHE} := (\mathsf{PP}, \mathsf{KeyGen}, \mathsf{Setup}, \mathsf{Enc}, \mathsf{MaskErr}, \mathsf{ServerDec}, \mathsf{PartDec}, \mathsf{FinDec})$.

$\underline{\mathsf{PP}(1^\kappa, 1^N)}$:

return $\mathsf{pp} := (T, n, k, m, p, q, Q = p \cdot q, \chi_{\mathsf{mlwe}}, \sigma_0, \sigma_1, \sigma_{\mathsf{flood}}, B_{\mathsf{pub}}, B_{\mathsf{eval}}, L_{\mathsf{Dec}})$

$\underline{\mathsf{KeyGen}(T, N)}$:

Let $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{mFHE.KeyGen}()$, and denote $\mathsf{sk} = \mathbf{s} = (s_1, \ldots, s_k)^\intercal$.

$\mathsf{err} := \boldsymbol{\zeta} := (\zeta^{(1)}, \ldots, \zeta^{(L_{\mathsf{Dec}})}) \leftarrow (\mathcal{R}(D_{\mathbb{Z}^n, \sigma_0 \sqrt{B_{\mathsf{pub}}^2 \mathbf{I} - \boldsymbol{\Sigma}_{\mathbf{s}}}}))^{L_{\mathsf{Dec}}}$, where $\boldsymbol{\Sigma}_{\mathbf{s}} := \sum_{i \in [k]} \mathsf{Gram}(s_i)$.

$\{\mathsf{sk}_i := \mathbf{s}_i, \mathsf{err}_i := \boldsymbol{\zeta}_i\}_{i \in [N]} \leftarrow \mathsf{Shamir}_{\mathcal{R}_q, T, N}.\mathsf{Share}((\mathsf{sk}, \mathsf{err}))$

for $(i, j) \in [N] \times [N]$ **do** $\mathsf{seed}_{i,j} \xleftarrow{\$} \{0,1\}^\kappa$

for $i \in [N]$ **do** $\mathsf{seedset}_i := \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j \in [N]}$

return $(\mathsf{evk}, \mathsf{pk}, \mathsf{sk}, \mathsf{err}, \{\mathsf{sk}_i, \mathsf{err}_i, \mathsf{seedset}_i\}_{i \in [N]})$

$\underline{\mathsf{Enc}(\mathsf{pk}, \mu \in \mathcal{M} := \mathcal{R}_2)}$:

return $\mathsf{ct} \leftarrow \mathsf{mFHE.Enc}(\mathsf{pk}, \mu)$ (defined in Algo. 13)

$\underline{\mathsf{Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)}$:

return $\mathsf{ct}_{\mathsf{eval}} \leftarrow \mathsf{mFHE.Eval}(\mathsf{evk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_l)$ (defined in Algo. 13)

$\underline{\mathsf{MaskErr}(\mathsf{err}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})}$:

$\mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} := \sum_{j \in \mathsf{act}} \mathsf{PRF}(\mathsf{seed}_{i,j}, \mathsf{sid}) \in \mathcal{R}_q$ ▷ Private row-sum mask of P_i

return $\mathsf{maskerr}_i^{(\mathsf{sid})} := \lambda_{\mathsf{act},i} \cdot \zeta_i^{(\mathsf{sid})} + \mathfrak{m}^{\mathsf{row}}_{\mathsf{act},i} \in \mathcal{R}_q$ ▷ Masked share of ζ

$\underline{\mathsf{ServerDec}(\mathsf{ct}, \mathsf{act}, \mathsf{sid}, \{\mathsf{maskerr}_i^{(\mathsf{sid})}\}_{i \in \mathsf{act}})}$:

$\mathsf{ct}_{\mathsf{fresh}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(0)$, $\mathfrak{E} \leftarrow \mathcal{R}(D_{\mathbb{Z}^n, \sigma_{\mathsf{flood}}})$ ▷ We use the discrete Gaussian for $\mathfrak{E}$

$\mathsf{ct}_{\mathsf{in}} := (\mathsf{ct}_{\mathsf{in},0}, \mathsf{ct}_{\mathsf{in},1}) := \mathsf{ct} + \mathsf{ct}_{\mathsf{fresh}} + (0, \mathfrak{E}) \in \mathcal{R}_Q^{k+1}$,

$\mathsf{ct}_{\mathsf{dec},0} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},0} \right\rceil_{\sigma_0} \in \mathcal{R}_q^k$

$\mathsf{ct}_{\mathsf{dec},1} := \left\lfloor \frac{1}{p} \cdot \mathsf{ct}_{\mathsf{in},1} \right\rceil_{\sigma_1} + \sum_{i \in \mathsf{act}} \mathsf{maskerr}_i^{(\mathsf{sid})} \in \mathcal{R}_q$

return $\mathsf{ct}_{\mathsf{dec}} := (\mathsf{ct}_{\mathsf{dec},0}, \mathsf{ct}_{\mathsf{dec},1}) := (\overline{\mathbf{a}}, \overline{b})$

$\underline{\mathsf{PartDec}(\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}), \mathsf{sk}_i, \mathsf{seedset}_i, \mathsf{act}, \mathsf{sid})}$:

$\mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} := \sum_{j \in \mathsf{act}} \mathsf{PRF}(\mathsf{seed}_{j,i}, \mathsf{sid}) \in \mathcal{R}_q$ ▷ Private column-sum mask of P_i

return $\mathsf{pd}_i := \lambda_{\mathsf{act},i} \cdot \overline{\mathbf{a}}^\intercal \mathbf{s}_i + \mathfrak{m}^{\mathsf{col}}_{\mathsf{act},i} \in \mathcal{R}_q$

$\underline{\mathsf{FinDec}(\mathsf{ct}_{\mathsf{dec}} := (\overline{\mathbf{a}}, \overline{b}), \{\mathsf{pd}_i\}_{i \in \mathsf{act}}, \mathsf{act}) \to \overline{\mu} \in \mathcal{M} \text{ or } \perp}$:

assert $\{|\mathsf{act}| \geq T\}$

return $\overline{\mu} := \lfloor (\overline{b} - \sum_{i \in \mathsf{act}} \mathsf{pd}_i) / \lfloor \frac{q}{2} \rfloor \rceil$