Key lifting : Multi-key Fully Homomorphic Encryption in plain model

Abstract. Multi-key Fully Homomorphic Encryption(MKFHE) based on Learning With Error(LWE) usually lifts ciphertexts of different users to new ciphertexts under a common public key to enable homomorphic evaluation. The main obstacle of current MKFHE schemese in applications is huge ciphertext expansion cost especially in data intensive scenario. For example, for an boolean circuit with input length N, multiplication depth L, security parameter λ , the number of additional encryptions introduced to obtain ciphertext expansion is $O(N\lambda^6L^4)$. In this paper we present a framework to slove this problem that we call Key-Lifting Multi-key Fully Homomorphic Encryption (KL-MKFHE). By introducing a key lifting procedure, the number of encryptio for a local user is pulled back to $O(N)$. Moreover, current MKFHE schemes are often based on Common Reference String model(CRS). In our LWE-based scheme, CRS is removed by using the leak resilient property of the left-over hash lemma(LHL). Due to the structural properties of polynomial rings, such LWE-based scheme cannot be trivially transplanted to RLWE-based scheme. We give a RLWE-based KL-MKFHE under Random Oracle Model(ROM) by introduing a bit commitment protocal.

Keywords: Multi-key homomorphic encryption · LWE · RLWE · Leakage resilient cryptography.

1 Introduction

Fully Homomorphic Encryption(FHE). The concept of FHE was proposed by Rivest et al. [39], within a year of publishing of the RSA scheme [40]. The first truly fully homomorphic scheme was proposed by Gentry in his doctoral dissertation [15]in 2009. Based on Gentry's ideas, a series of FHE schemes have been proposed [16] [42] [5] [14] [18] [10] [9], and their security and efficiency have been continuously improved. FHE is suitable to the problem of unilateral outsourcing computations. However in the case of multiple data providers, in order to support homomorphic evaluation, data must be encrypted by a common public key. Due to privacy of data, it is unreasonable to require participants to use other people's public keys to encrypt their own data.

Multiparty Computation (MPC). This problem was initialized by Yao in [43] [44], who considered two-party scenarios and gave a solution. Later, Goldreich, Micali and Wigderson extended the model to k participants with malicious adversaries in [27]. Compared to FHE, MPC is more mature, as [35] mentioned, "generic MPC is a fast moving field: In 2009 the first implementation of MPC for 2 parties with active security [37], was able to evaluate a circuit of 3×10^4 gates in 10^3 seconds. Only two years after, the same circuit is evaluated in less than 5 seconds [34]". Subsequently, for active attacks, the literature [22] [21] made a series of improvements.The first large-scale and practical application of multi-party computation (demonstrated on an actual auction problem) took place in Denmark in January 2008. However, MPC also has disadvantages: it suffers from high communication overhead and is vulnerable to attacks by corrupt participants, as [12] mentioned, "All the general protocols we have seen require a number of rounds that is linear in the depth of circuit. We do not know if this is inherent, in fact, we do not even know which functions can be computed with unconditional security and a constant number of rounds".

Multi-key Fully Homomorphic Encryption(MKFHE). To deal with privacy of multiple data providers, López-Alt et al. proposed the concept of MKFHE in [23] and construct the first MKFHE scheme based on modified-NTRU [41]. Conceptually, it is an enhancement of the FHE on function that allows data provider to encrypt data independent from other participants, its key generation and data encryption are done locally. To get the evaluated result, all participants are required to execute a round of threshold decryption protocol.

After López-Alt et al. proposed the concept of MKFHE, many schemes were proposed. In 2015, Clear and McGoldrick [11] constructed a GSW [18] LWE-based MKFHE. This scheme defined the total key as the concatenation of all keys, and constructed a masking scheme to converts the ciphertext under single key to total key by introducing CRS and circular LWE assumptions, which only supports single-hop computation. In 2016, Mukherjee and Wich [33], Perkert and shiehian [36], Brakerski and Perlman [7] constructed MKFHE scheme based on GSW respectively. [33] simplified the mask scheme of [11], and focused on constructing a two-round MPC protocol. The work of [36] and [7] was dedicated to constructing a multi-hop MKFHE, but their used different methods. [7] introduced bootstrapping to realize ciphertext expansion, thereby realizing the multi-hop function. [36] realized multi-hop function through ingenious construction. It is worth mentioning that all MKFHE schemes constructed based on the LWE scheme require a ciphertext expansion procedure.

1.1 Motivation

Ciphertext expansion is expensive: For MKFHE based on LWE, in order to support homomorphic evaluation, it is necessary to encrypt the random matrix $\mathbf{R} \in \mathbb{Z}_q^{m\times m}$ of each ciphertext to prepare for ciphertext expansion. For a boolean circuit with input length N, multiplication depth L, security parameter λ, $m = n\log q + \omega(\log \lambda)$, the additional encryption operation introduced is $O(N\lambda^6 L^4)$, which is $O(N)$ for single-key FHE. For computing-sensitive participants, this is a lot of overhead.

CRS can be removed: In many realistic scenarios, the data provider challenges the randomness of the common reference string, or challenges the fairness of a trusted third party. Leftover Hash Lemma (LHL) over integer ring $\mathbb{Z}$ enjoys the leakage resilient property : It can transform an average quality random sources into higher quality [19] which can be used to get rid of CRS.

Due to the more compact structure on polynomial ring and various efficient ring algorithms, it is generally believed that FHE scheme based on RLWE is more efficient than the homomorphic scheme based on LWE. This is the reason why most current MKFHE schemes, such as [8] [32] are constructed based on RLWE.

However, regularity lemma [25] over polynomial rings do not have corresponding properties, as [13]mentioned if the j-th Number theoretical Transfer(NTT) coordinate of each ring element in $\mathbf{x} = (x_1, \ldots, x_l)$ is leaked, then the j-th NTT coordinate of $a_{l+1} = \sum a_i x_i$ is defined, so a_{l+1} is very far from uniform, yet this is only a 1/n leakage rate. Therefore, to deal with this dilemma, we rely on random oracle model to present a RLWE-based KL-MKFHE without CRS by using bit commitment protocal.

1.2 Our Results

For trust-sensitive and data-sensitive scenario, we appropriately *tighten and loosen* the original definition of MKFHE, and introduce the concept of KL-MKFHE which is more suitable for such scenarios. Following this concept, we construct the first KL-MKFHE scheme based on LWE in the plain model.

Since regularity lemma [26] on rings has no corresponding leakage resilient properties, we cannot apply the LWE construction routine trivially to RLWE-based MKFHE, as a compromise, we introduce a round of bit commitment protocol to guarantee the independence of each participants, construct the corresponding KL-MKFHE based on ROM. We give a review of our definition and two scheme below.

The definition of KL-MKFHE :
Different from previous definition [33], we abandon ciphertext expansion procedure, instead, introducing a key lifting procedure which has the same function with ciphertext expansion, but at a lower cost. In addition to the properties that required by MKFHE, such as *Correctness, Compactness, semantic security, Simulatability of decryption*, KL-MKFHE should satisfy the following two additional properties :

- **Locally Computationally Compactness :** A leveled KL-MKFHE is locally computationally compact if the participants do the same number of encryptions as the single-key FHE scheme.
- **Low round complexity :** Only constant round interaction for arbitrarily many number of users is allowed in Key lifting procedure.

Scheme#1: LWE-based KL-MKFHE under plain model :

The security of Scheme#1 is based on the LWE assumption. The total private key is the sum of the private keys of all participants. We note that previous MKFHE schemes adopt this key structure are all based on the CRS model. Not only is the CRS removed, our solution is simpler and more efficient in construction : For a circuit with an input length N, our scheme requires local users to perform $O(N)$ encryption operations, while is $O(N\lambda^6 L^4)$ for those schemes that require ciphertext expansion.

However, in order to ensure the semantic security of encryption and make the threshold decryption procedure simulatable, we have to choose a larger smuging error to conceal the local decryption result. We bound the participants k by $\mathsf{poly}(\lambda)$, because a larger k will lead to a larger smuging error, which further leads to a larger q. Here, we choose $q = 2^{\lambda L} B_\chi$, the approximate factor of the GapSVP problem on lattice is $\tilde{O}(2^{\lambda L})$ for such q. For detailed security and parameters, please refer to Section4.

We give the efficiency comparison with the scheme [36] in Table1. Since we have no ciphertext expansion, our scheme has a much lower computational overhead.

Scheme	Space		Time	CRS
	PubKey + EvalKey	CT	EvalkeyGen	
[36]	$\tilde{O}(\lambda^6 L^4(k + N\lambda^3 L^2))$	$\tilde{O}(Nk^2\lambda^6 L^4)$	$\tilde{O}(N\lambda^{14} L^9)$	Yes
[6]	$\tilde{O}(k^4\lambda^{15} L^{11})$	$\tilde{O}(Nk^4\lambda^8 L^6)$	$\tilde{O}(Nk^3\lambda^{15} L^{10})$	Yes
Scheme#1	$\tilde{O}(k^2\lambda^6 L^4)$	$\tilde{O}(Nk^2\lambda^8 L^6)$	-	NO

Table 1. The notation $\tilde{O}$ hides logarithmic factors. The public key, evaluation key and ciphertext size are bits; the EvalkeyGen column denotes the number of multiplication operations over $\mathbb{Z}_q$; k denotes participants number; L denotes the circuit depth; λ is the security parameter.

Remark : We replaced n with λ. To achieve 2^λ security against known lattice attacks, one must have $n = \Omega(\lambda \log q/B_\chi)$, for our parameter settings $q = O(2^{\lambda L} B_\chi)$, thus we would like to be $n = \Omega(\lambda^2 L)$.

Scheme#2: RLWE-based KL-MKFHE under ROM :

Scheme#2 is based on circular RLWE. We introduce a bit commitment protocol to guarantee the randomness of each participant's public key. Due to the sum key structure, the dimension of $\mathbf{t} \otimes \mathbf{t}$ is independent of k, so the ciphertext relinearization algorithm pull the ciphertext after tensor product back to initial dimension by one shot, in addition, the "one shot algorithm" introduces less noise. We compared with [8] in terms of memory and computational overhead, the results are shown in Table2.

1.3 Related works

Unlike our scheme, [11] [36] [33] [7] [8] used the concatenation of all private key as the total key structure, and CRS are introduced. [3] removes CRS from a higher

Scheme	Space		Time		CRS
	Evalkey	CT	Relinear	Mult	
[8]	$\tilde{O}(kd)$	$\tilde{O}(kd)$	$\tilde{O}(k^2d)$	$\tilde{O}(k^2d)$	Yes
Scheme#2	$\tilde{O}(kd)$	$\tilde{O}(d)$	$O(1)$	$\tilde{O}(d)$	ROM

Table 2. The Evalkey and CT size are in bits, the Relinear and Mult columns denotes the number of scalar operations over $\mathbb{Z}_q$. The notation $\tilde{O}$ hides logarithmic factors, k denotes the number of participants; d denotes the dimension of the RLWE problem.

dimension, instead of using LHL or regularity lemma, they base on Multiparty Homomorphic Encryption(MHE) and modify the initialization method of its root node to achieve this purpose, more details please refer to [3]. [4] is the first scheme that introduce the summation of all private key as the total key, which is also under CRS. [6] is the first scheme using the leakage resilient property of LHL to get rid of the CRS, which has the concatenation total key structure, and ciphertext expansion is essential.

1.4 Overview of our construciton

Scheme#1 is based on DGSW scheme. we briefly review it first : let $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{(m\times n)})$, $\mathbf{s} \leftarrow \{0,1\}^m$, $\mathsf{pk} = (\mathbf{A}, \mathbf{b} = \mathbf{sA})$, $\mathsf{sk} = \mathbf{t} = (-\mathbf{s}, 1)$, plaintext $u \in \{0,1\}$, the DGSW ciphertext $\mathbf{C}$:

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R}+\mathbf{E}+u\mathbf{G},\ \mathbf{R} \leftarrow U(\mathbb{Z}_q^{(n\times m)}),\ \mathbf{E} \text{ is an noise matrix, } \mathbf{G} \text{ is a gadget matrix.}$$

Obviously, $\mathbf{tC} \approx u\mathbf{tG}$ (omit small noise)

Key Lifting procedure : Following the definition of KL-MKFHE, it requires the ciphertext encrypted by hybrid key hk which are outputted by $\mathsf{KeyLifting}(\cdot)$ and are different among participants, to support homomorphic evaluation without extra modification. We achieve this property by allowing two round interaction between participants.

For the convenience of explanation, we assume that there are only two participants $\mathsf{p}_1, \mathsf{p}_2$, naturally, the whole process can be extended to N participants.

- $\{\mathsf{hk}_1\} \leftarrow \mathsf{KeyLifting}(\{\mathsf{pk}_1, \mathsf{sk}_1\})$: input the DGSW key pair of p_1, where $\mathsf{pk}_1 = (\mathbf{A}_1, \mathbf{b}_{1,1})$, $\mathbf{b}_{1,1} = \mathbf{s}_1\mathbf{A}_1$, $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{s}_1 \leftarrow U\{0,1\}^{m-1}$. p_1, p_2 are engaged in the following two interaction
  - First round : p_1 broadcasts $(\mathbf{A}_1, \mathbf{b}_{1,1})$ and receives $\{\mathbf{A}_2, \mathbf{b}_{2,2}\}$ (from p_2).
  - Second round : p_1 generates and disclose $\mathbf{b}_{1,2}$, where $\mathbf{b}_{1,2} = \mathbf{s}_1\mathbf{A}_2$

After above two round interaction, p_1 receives $\mathbf{b}_{2,1}$(from p_2). Let $\mathbf{b}_1 = \mathbf{b}_{1,1}+\mathbf{b}_{2,1}$, p_1 output hybrid key $\mathsf{hk}_1 = (\mathbf{A}_1, \mathbf{b}_1)$, similarly, p_2 outputs hybrid key $\mathsf{hk}_2 = (\mathbf{A}_2, \mathbf{b}_2)$.

After the **Key Lifting procedure** is completed, p_1 and p_2 get the corresponding hybrid keys hk_1, hk_2. In short, what the key lifting procedure does is convert the DGSW key pair of p_1 and p_2 into the hybrid keys hk_1, hk_2, which are used to encrypt their data. Let $\bar{\mathbf{t}} = (-\mathbf{s}, 1)$, $\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2$, for ciphertext $\mathbf{C}_1$, $\mathbf{C}_2$ encrypted by hybrid key hk_1, hk_2 respectively :

$$\mathbf{C}_1 = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1 + \mathbf{E}_1 + u_1\mathbf{G}, \qquad \mathbf{C}_2 = \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{b}_2 \end{pmatrix} \mathbf{R}_2 + \mathbf{E}_2 + u_2\mathbf{G},$$

obviously we have $\bar{\mathbf{t}}\mathbf{C}_1 \approx u_1\bar{\mathbf{t}}\mathbf{G}$, $\bar{\mathbf{t}}\mathbf{C}_2 \approx u_2\bar{\mathbf{t}}\mathbf{G}$ (omit small error). Therefore, although $\mathbf{C}_1$ and $\mathbf{C}_2$ are encrypted by different hybrid keys, they correspond to the same decryption key $\bar{\mathbf{t}}$. As we'll point out later, however, this structure will drew some security concern. We remedy this problem by increasing the noise bounds in the last row of the noise matrix $\mathbf{E}$. we discuss the security of Scheme#1 in Section 4.5

Scheme#2: Compared with [8], Scheme#2 has one more round of bit commitment protocol and adopts the sum key structure. Due to the sum key structure, the Relinear algorithm can pull the ciphertext after the tensor product back to initial dimension by one shot. For a more details, please refer to section 5.

2 Preliminaries

2.1 Notation:

In this work, λ denotes security parameter, $\mathsf{negl}(\lambda)$ denotes the negligible function parameterized by λ, vectors are represented by lowercase bold letters such as $\mathbf{v}$, unless otherwise specified, vectors are row vectors by default, and matrices are represented by uppercase bold letters such as $\mathbf{M}$, $[k]$ denotes the set of integers $\{1, \dots, k\}$. If X is a distribution, then $a \leftarrow X$ denotes that value a according to the distribution X. If X is a finite set, then $a \leftarrow U(X)$ denotes that the value of a is uniformly sampled from X. For two distribution X, Y parameterized by λ, we use $X \overset{\mathsf{stat}}{\approx} Y$ to represent X and Y are statistically indistinguishable. Similarly, $X \overset{\mathsf{comp}}{\approx} Y$ means that there are computationally indistinguishable.

In order to decompose elements in $\mathbb{Z}_q$ into binary, we review the Gadget matrix [29] [2] here, let $\mathbf{G}^{-1}(\cdot)$ be the computable function that for any

$$\mathbf{M} \in \mathbb{Z}_q^{m \times n}, \text{ We have } \mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{ml \times n}, \text{ where } l = \lceil \log q \rceil$$

Let $\mathbf{g} = (1, 2, \dots, 2^{l-1}) \in \mathbb{Z}_q^l$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g} \in \mathbb{Z}_q^{m \times ml}$, it satisfies $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.

Definition 1. *A distribution ensemble $\{\mathcal{D}_n\}_{n \in [N]}$ supported over integer, is called B-bounded if :*

$$\Pr_{e \leftarrow \mathcal{D}_n} [\, |e| > B \,] = \mathsf{negl}(n).$$

In order to prove the security of our scheme under plain model and enable the simulatability of threshold decryption, we need the following lemma which is introduced by [4]:

Lemma 2 (in [4]). *Let $B_1 = B_1(\lambda)$, and $B_2 = B_2(\lambda)$ be positive integers and let $e_1 \in [-B_1, B_1]$ be a fixed integer, let $e_2 \in [-B_2, B_2]$ be chosen uniformly at random, Then the distribution of e_2 is statistically indistinguisable from that of $e_2 + e_1$ as long as $B_1/B_2 = \mathsf{negl}(\lambda)$.*

2.2 The Small Integer Solution(SIS) Problem

The Small Integer Solution(SIS) problem was introduced by Ajtai in the seminal work [1] which presented a family of one-way function based on SIS problem. Subsequent series of works [28] [31] [17] [30] have made efforts to reduce the size of q, the definition below comes from [31]:

Definition 3 (in [31]). *The small integer solution problem $\mathsf{SIS}_{m,n,q,\beta}$ (in the ℓ_∞ norm) is : given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a real β, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m/\{\mathbf{0}\}$ such that $\mathbf{Az} = \mathbf{0} \mod q$ and $||\mathbf{z}||_\infty < \beta$*

[30] proved that solving the $\mathsf{SIS}_{m,n,q,\beta}$ problem is at least as hard as approximating lattice problems in the worst case on lattices :

Theorem 4 (in [30]). *Let n and $m = \mathsf{poly}(n)$ be integers, let β be reals, let $Z = \{\mathbf{z} \in \mathbb{Z}^m : ||\mathbf{z}||_\infty < \beta\}$, and let $q > \beta \cdot n^\delta$ for some constant $\delta > 0$. Then solving (on the average, with non-negligible probability) $\mathsf{SIS}_{m,n,q,\beta}$ with parameters $m, n, q\beta$ and solution set $Z/\{\mathbf{0}\}$ is at least as hard as approximating lattice problems in the worst case on n dimensional lattices to within $\gamma = \tilde{O}(\beta\sqrt{n})$.*

2.3 The Learning With Error(LWE) Problem

The Learning With Error problem was introduced by Regev [38].

Definition 5 (LWE). *Let λ be security parameter, for parameters $n = n(\lambda)$ be an integer dimension, $q = q(\lambda) > 2$ be an integer, and a distribution $\chi = \chi(\lambda)$ over $\mathbb{Z}$, the $\mathsf{LWE}_{n,q,\chi}$ problem is to distinguish the following distribution:*

- *$\mathcal{D}_0$: the jointly distribution $(\mathbf{A}, \mathbf{z}) \in (\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n)$ is sampled by $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n}) \quad \mathbf{z} \leftarrow U(\mathbb{Z}_q^n)$*
- *$\mathcal{D}_1$: the jointly distribution $(\mathbf{A}, \mathbf{b}) \in (\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n)$ is computed by $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n}) \quad \mathbf{b} = \mathbf{sA} + \mathbf{e}$, where $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n) \quad \mathbf{e} \leftarrow \chi^m$*

The $\mathsf{LWE}_{n,q,\chi}$ assumption assuming that $\mathcal{D}_0 \overset{\mathsf{comp}}{\approx} \mathcal{D}_1$. Regev [38] proved that for certain moduli q and Gaussian error distributions χ the $\mathsf{LWE}_{n,q,\chi}$ problem is true as long as certain worst case lattice problems are hard to solve using a quantum algorithm.

2.4 The Ring Learning With Error(RLWE) Problem

Lyubaskevsky, Peikert and Regev defines The RLWE problem in [24] as follows:

Definition 6 (RLWE). *Let λ be a security parameter. For parameters $d = d(\lambda)$, where d is a power of 2, $q = q(\lambda) > 2$,and a distribution $\chi = \chi(\lambda)$ over $R = \mathbb{Z}[x]/x^d + 1$, let $R_q = R/qR$, the $\mathsf{RLWE}_{d,q,\chi}$ problem is to distinguish the following distribution:*

- *$\mathcal{D}_0$: the jointly distribution $(a, z) \in R_q^2$ is sampled by $(a, z) \leftarrow U(R_q^2)$.*
- *$\mathcal{D}_1$: the jointly distribution $(a, b) \in R_q^2$ is computed by $a \leftarrow U(R_q)$, $b = as+e$, where $s \leftarrow U(R_q)$, $e \leftarrow \chi$.*

[24] gave a reduction from the $\mathsf{RLWE}_{d,q,\chi}$ problem to the Gap-SVP problem on an ideal lattice, which is now generally considered to be intractable. Specially, [24] indicated that The $\mathsf{RLWE}_{n,q,\chi}$ problem is also infeasible when s is sampled from nosie distribution χ. In homomorphic encryption, this property is especially popular, because the low-norm s introduces less noise during homomorphic computation.

2.5 Dual-GSW(DGSW) Encryption scheme

The DGSW scheme [6] and GSW scheme is similar to Dual-Regev scheme and Regev scheme resp. which is defined as follows:

- $\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$: For a given security parameter λ, circuit depth L, choose a appropriate lattice dimension $n = n(\lambda, L)$, $m = n \log q + \omega(\lambda)$, a discrete Gaussian distribution $\chi = \chi(\lambda, L)$ over $\mathbb{Z}$, which is bouned by B_χ, module $q = \mathsf{poly}(n) \cdot B_\chi$ satisfying the $\mathsf{LWE}_{n,q,\chi,B_\chi}$ problem, Output $\mathsf{pp} = (n, m, q, \chi, B_\chi)$ as the initial parameters.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$: Let $\mathsf{sk} = \mathbf{t} = (-\mathbf{s}, 1)$, $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, where $\mathbf{s} \leftarrow U\{0,1\}^{m-1}$, $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m-1\times n})$, $\mathbf{b} = \mathbf{sA} \mod q$
- $\mathbf{C} \leftarrow \mathsf{Enc}(\mathsf{pk}, u)$: Input public key pk and plaintext u, choose a random matrix $\mathbf{R} \leftarrow U(\mathbb{Z}_q^{n\times w})$, $w = ml$, $l = \lceil \log q \rceil$ and an error matrix $\mathbf{E} \leftarrow \chi^{n\times w}$, Output the ciphertext :
$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \mathbf{E} + u\mathbf{G}, \text{ where } \mathbf{G} \text{ is a gadget Matrix.}$$
- $u \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathbf{C})$: Input private key sk, ciphertext $\mathbf{C}$, let $\mathbf{w} = (0, \dots, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, $v = \langle \mathbf{tC}, \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, output $u' = \lceil \frac{v}{q/2} \rfloor$.

Homomorphic addition and multiplication : For ciphertext $\mathbf{C}_1$, $\mathbf{C}_2 \in \mathbb{Z}_q^{m\times w}$, let $\mathbf{C}_{\mathsf{add}} = \mathbf{C}_1 + \mathbf{C}_2$, $\mathbf{C}_{\mathsf{mult}} = \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2)$. It is easy to verify that $\mathbf{C}_{\mathsf{add}}$ and $\mathbf{C}_{\mathsf{mult}}$ are ciphertext of $u_1 + u_2$ and $u_1 u_2$, respectively.

For the security and correctness of the DGSW scheme, please refer to [6]. Compared with the GSW scheme, DGSW scheme has bigger ciphertext, which is

$O(n^2 \log^3 q)$, while $O(n^2 \log q)$ for GSW scheme. As [6] mentioned, DGSW scheme makes it more convenient to use the leakage resilient property of LHL to remove CRS.

2.6 Multi-Key Fully Homomorphic Encryption(MKFHE)

We review the definition of MKFHE in detail here, the main purpose of which is to compare with the definition of KL-MKFHE we proposed later.

Definition 7. *Let λ be the security parameter, L be the circuit depth, and k be the number of participants. A Leveled multi-key fully homomorphic encryption scheme consists of a tuple of efficient probabilistic polynomial time algorithms* MKFHE=(Init, Gen, Enc, Expand, Eval, Dec) *which defines as follows.*

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L)$ *: Input security parameter λ, circuit depth L, output system paremeter* pp*. We assume that all algorithm take* pp *as input.*
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{id})$ *: Input* pp*, identity* id*, output a key pair for participant* p_i*.*
- $c_i \leftarrow \mathsf{Enc}(\mathsf{pk}_i, u_i)$ *: Input* pk_i *and* u_i*, output ciphertext* c_i*.*
- $\bar{c}_i \leftarrow \mathsf{Expand}(\mathsf{pk}, c_i)$*:Input the ciphertext* c_i *of participant* p_i*, the public key set* $\mathsf{pk} = \{\mathsf{pk}_i\}_{i\in[k]}$ *of all participants, output expanded ciphertext* $\bar{c}_i$ *which is under* $f(\mathsf{sk}_i, \dots \mathsf{sk}_k)$ *whose structure is undefined.*
- $\bar{c}_{eval} \leftarrow \mathsf{Eval}(\bar{c}, \mathcal{C})$*:Input circuit* $\mathcal{C}$*, the set of all ciphertext* $\bar{c} = \{\bar{c}_1 \dots \bar{c}_N\}$ *while* N *is the input length of circuit* $\mathcal{C}$*, output evaluated ciphertext* $\bar{c}_{eval}$
- $u \leftarrow \mathsf{Dec}(\bar{c}_{eval}, f(\mathsf{sk}_1 \dots \mathsf{sk}_k))$ *: Input evaluated ciphertext* $\bar{c}_{eval}$*, total private key function* $f(\mathsf{sk}_1 \dots \mathsf{sk}_k)$*, output* u

Remark :

1. The $\mathsf{Expand}(\cdot)$ algorithm is not necessary. For example, in the RLWE-based MKFHE scheme, the ciphertext expansion procedure is trivial, but in the LWE-based MKFHE scheme, the ciphertext expansion is a complicated and time-consuming process.
2. The ciphertext structure function $f(\mathsf{sk}_1, \dots, \mathsf{sk}_k)$ represents an organization form, or a certain function, which is not unique, for example, it can be the concatenation of all keys or the sum of all keys.

Properties implicited in the definition of MKFHE: For the above definition, each participant is required in key generation and encryption phase independently to generates their own keys and completes the encryption operation without interaction between participants. These two phases are similar to single-key homomorphic encryption, the computational overhead is independent of k and only related to λ and L, only in the decryption phase, interaction is involved when participants perform a round of decryption protocol.

3 Key Lifting Multi-key Fully Homomorphic Encryption(KL-MKFHE) scheme

3.1 The definition of KL-MKFHE

In order to cope with computationally-sensitive and trust-sensitive scenarios, we appropriately *tighten and loosen* the definition of MKFHE, we abandon ciphertext expansion procedure and introduce a **Key lifting** procedure. In addition, a tighter bound is required on the amount of local computation, as a compromise, we allow a small amount of interaction during **Key lifting**.

Definition 8. *A leveled* KL-MKFHE *scheme is a tuple of probabilistic polynomial time algorithm* (Init, Gen, KeyLifting, Enc, Eval, Dec)*, which can be divided into two phases, online phase:* KeyLifting *and* Dec*, where interaction is allowed between participants, but the rounds should be constant, local phase :* Init*,* Gen*,* Enc*, and* Eval*, whose operations do not involve interaction. These five algorithms are described as follows:*

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L)$*:Input security parameter* λ*, circuit depth* L*, output public parameters* pp.
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{id})$*:Input public parameter* pp*, identity* id*, output the key pair of participant* p_i
- $\mathsf{hk}_i \leftarrow \mathsf{KeyLifting}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{id})$*: Input key pair of participants* p_i*, output the hybrid key* hk_i *of* p_i*..*
- $c_i \leftarrow \mathsf{Enc}(\mathsf{hk}_i, u_i)$*: Input* u_i *and* hk_i*, output ciphertext* c_i
- $\hat{c} \leftarrow \mathsf{Eval}(\mathcal{C}, S)$*: Input circuit* $\mathcal{C}$*, ciphertext set* $S = \{c_i\}_{i\in[N]}$ *, output ciphertext* $\hat{c}$
- $u \leftarrow \mathsf{Dec}(\hat{c}, f(\mathsf{sk}_1 \dots \mathsf{sk}_k))$*: Input evaluated ciphertext* $\hat{c}$*,* $f(\mathsf{sk}_1 \dots \mathsf{sk}_k)$*, output* u*.*

Remark : KL-MKFHE does not have ciphertext expansion procedure, indeed, the inputed ciphertext set S in $\mathsf{Eval}(\cdot)$ is encrypted by participants under their own hybrid key hk_i which are different among participants, however, the resulting ciphertext c_i supports homomorphic evaluation without extra modification.

we require KL-MKFHE to satisfy the following properties:

Locally Computationally Compactness : *A leveled* KL-MKFHE *is locally computationally compact if the participants do the same number of encryptions as the single-key FHE scheme.*

Low round complexity : *Only constant round interaction is allow in* $\mathsf{KeyLifting}(\cdot)$ *procedure.*

IND-CPA security of encryption : *Let* λ *be the security parameter,* $L = \mathsf{poly}(\lambda)$ *is the circuit depth, for any probabilistic polynomial time adversary* $\mathcal{A}$*, he can distinguish the following two distributions with negligible advantage.*

$$\Pr\left[\,\mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 1)) - \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0)) \neq 0\,\right] = \mathsf{negl}(\lambda).$$

Correctness and Compactness : *A leveled* $\mathsf{KL\text{-}MKFHE}$ *scheme is correct if for a given security parameter* λ*, circuit depth* L*, participants* k*, we have the following*

$$\Pr\left[\,\mathsf{Dec}(f(\mathsf{sk}_1 \dots \mathsf{sk}_k), \hat{c}) \neq \mathcal{C}(u_1 \dots u_N)\,\right] = \mathsf{negl}(\lambda).$$

probability is negligible, where $\mathcal{C}$ *is a circuit with input length* N *and depth length less than* L*. A leveled* $\mathsf{KL\text{-}MKFHE}$ *scheme is compact, if the size* $\hat{c}$ *of evaluated ciphertext is bounded by* $\mathsf{poly}(\lambda, L, k)$*, but independent of circuit size.*

4 Scheme#1 : a KL-MKFHE scheme based on DGSW in plain model

4.1 Key lifting procedure

Following the definition of $\mathsf{KL\text{-}MKFHE}$, it requires the ciphertext encrypted by hybrid key hk which is outputted by $\mathsf{KeyLifting}(\cdot)$ algorithm is different among participants, to support homomorphic evaluation without extra modification. We achieve this property by allowing two round interaction between participants.

Key Lifting

- $\{\mathsf{hk}_i\}_{i\in[k]} \leftarrow \mathsf{KeyLifting}(\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]})$: Input the DGSW key pair $\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]}$ of all participants, where $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{s}_i \leftarrow U\{0,1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i \mod q$. All participants are engaged in the following two interaction :
  - First round : p_i broadcasts $(\mathbf{A}_i, \mathbf{b}_{i,i})$ and receives all $\{\mathbf{A}_j, \mathbf{b}_{j,j}\}_{j\in[k]}$.
  - Second round : p_i generates and disclose $\{\mathbf{b}_{i,j}\}_{j\in[k]}$, where $\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j \mod q$

After above two round interaction, p_i receives $\{\mathbf{b}_{j,i}\}_{j\in[k]}$

$$\text{let } \mathbf{b}_i = \sum_{j=1}^{k} \mathbf{b}_{j,i},\ \ \mathsf{p}_i \text{ output hybrid key } \mathsf{hk}_i = (\mathbf{A}_i, \mathbf{b}_i)$$

Let $\bar{\mathbf{t}} = (-\mathbf{s}, 1)$, $\mathbf{s} = \sum_{i=1}^{k} \mathbf{s}_i$, for ciphertext $\mathbf{C}_i$, $\mathbf{C}_j$ encrypted by hybrid key hk_i, hk_j respectively :

$$\mathbf{C}_i = \begin{pmatrix} \mathbf{A}_i \\ \mathbf{b}_i \end{pmatrix} \mathbf{R}_1 + \mathbf{E}_1 + u_i\mathbf{G}, \qquad \mathbf{C}_j = \begin{pmatrix} \mathbf{A}_j \\ \mathbf{b}_j \end{pmatrix} \mathbf{R}_2 + \mathbf{E}_2 + u_j\mathbf{G},$$

obviously we have $\bar{\mathbf{t}}\mathbf{C}_i \approx u_i\bar{\mathbf{t}}\mathbf{G}$, $\bar{\mathbf{t}}\mathbf{C}_j \approx u_j\bar{\mathbf{t}}\mathbf{G}$(omit small error). Therefore, although $\mathbf{C}_i$ and $\mathbf{C}_j$ are encrypted by different hybrid keys, they correspond to the same decryption key $\bar{\mathbf{t}}$. As we'll point out later, however, this structure will drew some security concern. We remedy this problem by increasing the noise bounds in the last row of the noise matrix $\mathbf{E}$. we discuss the security of the scheme in Section 4.5

4.2 The entire scheme

Scheme#1 is based on the DGSW scheme, containing the following five algorithm (Init, Gen, KeyLifting, Enc, Eval, Dec)

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L)$: Let λ be security parameter, L be circuit depth, lattice dimension $n = n(\lambda, L)$, noise distribution χ over $\mathbb{Z}$, and $e \leftarrow \chi$, where $|e|$ is bounded by B_χ, modulus $q = 2^{\lambda L} B_\chi$, $k = \mathsf{poly}(\lambda)$, $m = kn \log q + \lambda$, suitable choosing above parameters to make $\mathsf{LWE}_{n,m,q,B_\chi}$ is infeasible. Output $\mathsf{pp} = (k, n, m, q, \chi, B_\chi)$
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp})$: Input pp, output the DGSW key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$ of participants p_i, where $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{s}_i \leftarrow U\{0,1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i \mathbf{A}_i \mod q$.
- $\mathsf{hk}_i \leftarrow$ **Key Lifting** : All participants are engaged in the **Key lifting procedure 4.1**, output the hybrid key hk_i.
- $\mathbf{C}_i \leftarrow \mathsf{Enc}(\mathsf{hk}_i, u_i)$: Input hybrid key hk_i, plaintext u_i, output ciphertext $\mathbf{C}_i = \begin{pmatrix} \mathbf{A}_i \\ \mathbf{b}_i \end{pmatrix} \mathbf{R} + \mathbf{E} + u_i \mathbf{G}$, where $\mathbf{R} \leftarrow \chi^{n \times ml}$, $l = \lceil \log q \rceil$, $\mathbf{E} = \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}$, $\mathbf{E}_0 \leftarrow \chi^{(m-1)\times ml}$, $\mathbf{e}_1 \leftarrow \chi'^{\, ml}$, χ' is a distribution over $\mathbb{Z}$, satisfying $|\mathbf{e}_1|$ is bounded by $2^{\lambda^{\epsilon_1}} B_\chi$, $\epsilon_1 \in (0, \frac{1}{2})$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g}$ is a gadget matrix.
- $\hat{\mathbf{C}} \leftarrow \mathsf{Eval}(S, \mathcal{C})$: Input the ciphertext $S = \{\mathbf{C}_i\}_{i \in [N]}$ which are encrypted by hybrid key $\{\mathsf{hk}_i\}_{i \in [k]}$, circuit $\mathcal{C}$ with input length N, output $\hat{\mathbf{C}}$.

Homomorphic addition and multiplication

- $\mathbf{C}_{\mathsf{add}} \leftarrow \mathsf{Add}(\mathbf{C}_1, \mathbf{C}_2)$: Input ciphertext $\mathbf{C}_1$, $\mathbf{C}_2$, output $\mathbf{C}_{\mathsf{add}} = \mathbf{C}_1 + \mathbf{C}_2$, Obviously $\bar{\mathbf{t}}\mathbf{C}_{\mathsf{add}} \approx (u_1 + u_2)\bar{\mathbf{t}}\mathbf{G}$
- $\mathbf{C}_{\mathsf{mult}} \leftarrow \mathsf{Mult}(\mathbf{C}_1, \mathbf{C}_2)$: Input ciphertext $\mathbf{C}_1,\ \mathbf{C}_2$, output $\mathbf{C}_{\mathsf{mult}} = \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2)$, Obviously $\bar{\mathbf{t}}\mathbf{C}_{\mathsf{mult}} \approx u_1 u_2 \bar{\mathbf{t}}\mathbf{G}$

Distributed decryption Similar to [33], the decryption procedure is a distributed procedure :

- LocalDec: Input $\hat{\mathbf{C}}$, let $\hat{\mathbf{C}} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}$, where $\mathbf{C}_0 \in \mathbb{Z}_q^{(m-1)\times ml}$, $\mathbf{c}_1 \in \mathbb{Z}_q^{ml}$, p_i computes $\beta_i = \langle \mathbf{s}_i,\ \mathbf{C}_0 \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, and set $\gamma_i = \beta_i + e_i''$, where $\mathbf{w} = (0, \ldots, 0, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, $e_i'' \leftarrow \chi''$ is a distribution over $\mathbb{Z}$, satisfying $|e_i''| < 2^{L\lambda^{\epsilon_2}} B_\chi,\ \epsilon_2 \in (\frac{1}{2}, 1)$, then p_i broadcast γ_i
- FinalDec: After received $\{\gamma_i\}_{i \in [k]}$, let $\gamma = \sum_{i=1}^k \gamma_i + \langle \mathbf{c}_1,\ \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, output $u = \lceil \frac{\gamma}{q/2} \rfloor$

4.3 Bootstrapping

In order to eliminate the dependence on the circuit depth to achieve fully homomorphism, we need to use Gentry's bootstrapping technology. It is worth noting that the bootstrapping procedure of Scheme#1 is the same as single-key homomorphic scheme: After **Key lifting** procedure, participant p_i uses hybrid key hk_i to encrypt s_i to obtain evaluation key evk_i. Because evk_i and $\hat{\mathbf{C}}$ are both ciphertexts under $\bar{\mathbf{t}} = (-\sum_{i=1}^{k} \mathbf{s}_i, 1)$, homomorphic evaluation of the decryption circuit could be executed directly as $\hat{\mathbf{C}}$ are need to be refresh. Therefore, in order to evaluate any depth circuit, we only need to set the initial parameters to satisfy the homomorphic evaluation of the decryption circuit.

However, for those MKFHE schemes that requires ciphertext expansion, additional ciphertext expansion is required, for the reason that $\hat{\mathbf{C}}$ is the ciphertext under $\bar{\mathbf{t}}$, but $\{\mathsf{evk}_i\}_{i\in[k]}$ are the ciphertext under $\{\mathbf{t}_i\}_{i\in[k]}$. This is another large amount of computational overhead, because in order to expand $\{\mathsf{evk}_i\}_{i\in[k]}$, participant p_i needs to encrypt the random matrix of the ciphertext corresponding to evk_i.

4.4 Correctness analysis

To illustrate the correctness of Scheme#1, we first study the accumulation of noise:

$$\text{Let } \mathbf{s} = \sum_{i=1}^{k} \mathbf{s}_i,\ \mathbf{t} = (-\mathbf{s}, 1),\ \text{for fresh ciphertext } \mathbf{C} = \begin{pmatrix} \mathbf{A}_i \\ \mathbf{b}_i \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} + u\mathbf{G}$$

we have $\mathbf{tC} = \mathbf{e}_1 + \mathbf{sE}_0 + u\mathbf{tG}$, let $\mathbf{e}_{\mathsf{init}} = \mathbf{e}_1 + \mathbf{sE}_0$, Obviously $|\mathbf{e}_{\mathsf{init}}| < (2^{\lambda^{\epsilon_1}} + km)B_\chi$.

After L depth circuit evaluation, let $\mathbf{e}_L = (ml)^L \mathbf{e}_{\mathsf{init}}$. According to the **distributed encryption** of Scheme#1 we have :

$$\gamma = \sum_{i=i}^{k} \beta_i + \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{w}^T) = \langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + \sum_{i=1}^{k} e_i'' + u\lfloor \frac{q}{2} \rceil \tag{1}$$

Let $e_{\mathsf{final}} = \langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + \sum_{i=1}^{k} e_i''$. In order to decrypt correctly, it requires $e_{\mathsf{final}} < \frac{q}{4}$. For our parameter settings, obviously $|e_i''| > \langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle$, for taking the logarithm of both sides:

$$\log e_i'' = \lambda^{\epsilon_2} L$$
$$\log\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle = \log(knL(\lambda)^{(2L+1)}(2^{\lambda^{\epsilon_2}} + k^2 n\lambda L))B_\chi = O(L + \lambda^{\epsilon_2})$$

thus $e_{\mathsf{final}} < \frac{q}{4}$.

4.5 Semantic Security of Encryption against Semi-Malicious Adversary

For a honest player p_i, he generates $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m\times n)})$, $\mathbf{s}_i \leftarrow \{0,1\}^m$ as the protocol specification, but a semi-malicious adversary may generates it arbitrarily and adaptively. For arbitrary $\mathbf{A}_i$, Brakerski et al. [6] proved that the DGSW scheme is leakage resilient. We deal with what happens when $\mathbf{s}_i$ changes. Let B_{sis} be the bound keeping the $\mathsf{SIS}_{m,n,q,B_{\mathsf{sis}}}$ problem hard, according to Theorem 4, if $B_{\mathsf{sis}} \ll q^{n/m}$, the problem is vacuously hard, most likely, such solutions do not exists, if $B_{\mathsf{sis}} \gg \gamma^m \cdot q^{n/m}$, this is an instance of approx-SVP with exponential approximation factor γ, which can be solved by LLL [20], somewhere in between these bounds is where cryptography takes place, typically for $B_{\mathsf{sis}} = q^{n/m} \cdot \mathsf{poly}(\lambda)$. For our parameter Settings $B_{\mathsf{sis}} = q^{n/m} \cdot \mathsf{poly}(\lambda) = \mathsf{poly}(\lambda)$. We complete the simulation by constructing a reduction from Scheme#1 to the DGSW scheme. Consider the following Game:

1. Challenger generates $\mathsf{pk}_{\mathsf{DGSW}} = (\mathbf{A}, \mathbf{b}_1)$ where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{b}_1 = \mathbf{s}_1\mathbf{A}$, $\mathbf{s}_1 \leftarrow U\{0,1\}^m$ and send $\mathsf{pk}_{\mathsf{DGSW}}$ to adversary $\mathcal{A}$
2. $\mathcal{A}$ adaptively chooses $\{\mathbf{s}_i\}_{i\in[k]/1}$ where $|\mathbf{s}_i| < B_{\mathsf{sis}}$ and generates $\{\mathbf{b}_i\}_{i\in[k]/1}$, $\mathbf{b}_i = \mathbf{s}_i\mathbf{A}$, choose a bit $u \in \{0,1\}$ and set $\mathsf{hk}_{\mathsf{Scheme\#1}} = (\mathbf{A}, \mathbf{b})$, where $\mathbf{b} = \sum_{i=1}^{k} \mathbf{b}_i$, then send $\mathsf{hk}_{\mathsf{Scheme\#1}}$ and u to Challenger.
3. Challenger choose a bit $\alpha \in \{0,1\}$, if $\alpha = 0$, set $\mathbf{C}_{\mathsf{Scheme\#1}} \leftarrow \mathsf{Enc}(\mathsf{hk}_{\mathsf{Scheme\#1}}, u)$, otherwise $\mathbf{C}_{\mathsf{Scheme\#1}} \leftarrow U(\mathbb{Z}_q^{m\times ml})$, and send $\mathbf{C}_{\mathsf{Scheme\#1}}$ to $\mathcal{A}$
4. After receiving $\mathbf{C}_{\mathsf{Scheme\#1}}$, $\mathcal{A}$ output bit $\bar{\alpha}$, if $\bar{\alpha} = \alpha$, then $\mathcal{A}$ wins.

Lemma 9. *Let* $\mathsf{Adv} = |Pr[\bar{\alpha} = \alpha] - \frac{1}{2}|$ *denote* $\mathcal{A}$*'s advantage in winning the game, If* $\mathcal{A}$ *can win the game with advantage* Adv*, then* $\mathcal{A}$ *can distinguish between the ciphertext distribution of* DGSW *and the uniform random distribution with the same advantage.*

Proof. We construct $\mathbf{C}_{\mathsf{Scheme\#1}}$ by $\mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, u)$:

1. First, Challenger generates $\mathsf{pk}_{\mathsf{DGSW}}$ like the step 1 of above Game, set $\mathbf{C}_{\mathsf{DGSW}} = \mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, u)$ send the both to $\mathcal{A}$.
2. $\mathcal{A}$ generates $\{\mathbf{s}_i\}_{i\in[k]/1}$, let $\mathbf{s}' = \sum_{i=2}^{k} \mathbf{s}_i$, $\mathbf{C}_{\mathsf{DGSW}} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}$, $\mathbf{C}' = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 + \mathbf{c}'_1 \end{pmatrix}$, where $\mathbf{c}'_1 = \mathbf{s}'\mathbf{C}_0$, obviously $\mathbf{C}' = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0 \end{pmatrix}$.

For our parameter settings $|\mathbf{e}_1| < 2^{\lambda^{\epsilon_1}} B_\chi$, $|\mathbf{s}'\mathbf{E}_0| < kmB_\chi B_{\mathsf{sis}}$, thus $\mathbf{e}_1/\mathbf{s}'\mathbf{E}_0 = \mathsf{negl}(\lambda)$, we have $\mathbf{C}' \overset{\mathsf{stat}}{\approx} \mathbf{C}_{\mathsf{Scheme\#1}}$, if $\mathcal{A}$ can distinguish between $\mathbf{C}_{\mathsf{Scheme\#1}}$ and uniform random distribution by advantage Adv, then he can distinguish between $\mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, u)$ and the uniform random distribution with the same advantage.

Remark: we require k to be bounded by $\mathsf{poly}(\lambda)$, because if a larger k is introduced, it will lead to a larger smudging error, which further leads to a larger q. For our choice of $q = 2^{\lambda L} B_\chi$, the corresponding approximation factor of the SVP problem is $\tilde{O}(2^{\lambda L})$

4.6 Simulatability of distributed decryption procedure

Similar to [33], we get a weak simulation of the distributed decryption procedure : input all private keys $\{\mathsf{sk}_j\}_{j\in[k]/i}$ except sk_i, evaluated result u_{eval}, ciphertext $\hat{\mathbf{C}}$, we can simulate the local decryption result γ_i. For stronger security requirements : input any private key set $\{\mathsf{sk}_j\}_{j\in S}$, S is any subset of $[k]$, evaluated result u_{eval} and ciphertext $\hat{\mathbf{C}}$, to simulate $\{\gamma_i\}_{i\in U,\ U=[k]-S}$, we don't know how to achieve it.

According to Equantion 1 we have $\gamma = \sum_{i=1}^{k} \gamma_i + \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{w}^T)$

$$\text{thus } \gamma_i = u_{\mathsf{eval}}\lfloor\frac{q}{2}\rceil + e_{\mathsf{final}} + \sum_{i=1}^{k} e_i'' + \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{w}^T) - \sum_{j\neq i}^{k} \gamma_j$$

For a simulator $\mathcal{S}$, input $\{\mathsf{sk}_j\}_{j\in[k]/i}$, evaluated result u_{eval}, ciphertext $\hat{\mathbf{C}}$, output simulated γ_i'

$$\gamma_i' = u_{\mathsf{eval}}\lfloor\frac{q}{2}\rceil + \sum_{i=1}^{k} e_i'' + \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{w}^T) - \sum_{j\neq i}^{k} \gamma_j.$$

For our parameter settings, we have :

$$|\sum_{i=1}^{k} e_i''| < k\cdot 2^{L\lambda^{\epsilon_2}}B_\chi$$
$$e_{\mathsf{final}} < kn(L\lambda)^{(2L+1)}(2^{\lambda^{\epsilon_2}} + k^2 nL\lambda)B_\chi = 2^{O(L\lambda^{\epsilon_1})}B_\chi$$
$$\text{thus } |e_{\mathsf{final}}/\sum_{i=1}^{k} e_i''| = k\cdot 2^{-\omega(L\lambda^{\epsilon_2} - L\lambda^{\epsilon_1})} = \mathsf{negl}(\lambda)$$

we have $\gamma_i \overset{\mathsf{stat}}{\approx} \gamma_i'$.

5 Scheme#2 : KL-MKFHE based on RLWE in ROM

It is regrettable that general polynomial ring $R : \mathbb{Z}[x]/f(x)$ cannot enjoy the leak resilient property of the LHL on the integer ring $\mathbb{Z}$. This means that we cannot transplant the above construction process trivially to RLWE-based FHE. Indeed, [13] pointed out that for $\mathbf{x} = (x_1, \ldots, x_l) \in R^l$, if the j-th NTT coordinate of each $x_{i,i\in[l]}$ is leaked, then the j-th NTT coordinate of $a_{l+1} = \sum_{i=1}^{l} a_i x_i$ is defined, thus a_{l+1} is far from random, although the leakage ratio is only 1/n. We also noticed a trivial solution : for $\mathbf{a}, \mathbf{s} \in R_q^l$, $b = \langle \mathbf{a}, \mathbf{s}\rangle \in R_q$, b leaks information about $\mathbf{s}$ at most $n\log q$ bits, therefore, as long as we set l long enough, for example, $l = l + n\log q$, then obviously b is close to uniformly random, but this will result in a extremely large key, thus it is not practical.

To ensure the independence of the $\{a_i\}_{i\in[k]}$ generated by each participant, we simply added a round of bit commitment protocol. Under the ROM, the cryptographic hash function is used to ensure the independence of $\{a_i\}_{i\in[k]}$. Let $H : \{0,1\}^\star \to \{0,1\}^\lambda$ be a cryptography hash function, $a_i \in R_q$, $H(a_i) = \delta_i$. For a given $\delta \in \{0,1\}^\lambda$, an adversary $\mathcal{A}$ sends a query $x \in \{0,1\}^\star$ to H, which happens to have probability $\Pr[\,H(x) = \delta\,] = \frac{1}{2^\lambda}$. Let Adv denotes the probability that $\mathcal{A}$ finds a collision after making $q_{\mathsf{ro}} = \mathsf{poly}(\lambda)$ queries, Obviously $\mathsf{Adv} = \mathsf{negl}(\lambda)$, we have the following result.

Lemma 10. *For a given $\delta \in \{0,1\}^\lambda$, k probabilistic polynomial time(ppt) adversary $\mathcal{A}$, Each $\mathcal{A}$ makes $q_{ro} = \mathsf{poly}(\lambda)$ queries to H, let $\overline{\mathsf{Adv}}$ denotes the probability of finding a collision, then:* $\overline{\mathsf{Adv}} = \mathsf{negl}(\lambda)$

For Scheme#2, we only describe its key generation and re-linearization procedure in detail, the encryption, evaluation and decryption algorithm is similar to other RLWE-based MKFHE schemes.

Key generation with bit commitment.

k participants perform the following steps to get their own public key and evaluation key

1. $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^L)$:Input security parameter λ, circuit depth L, output $\mathsf{pp} = (d, q, \chi, B_\chi)$, which χ is an noise distribution over $R : \mathbb{Z}[x]/x^d + 1$, satisfying $e \leftarrow \chi$, $||e||_\infty^{can}$ is bounded by B_χ, and $\mathsf{RLWE}_{d,q,\chi,B_\chi}$ is infeasible.
2. p_i generates $\Phi_i = \{a_i, \mathbf{d}_i, \mathbf{f}_i\}$ where $a_i \leftarrow U(R_q)$ is used for public key, $\mathbf{d}_i$, $\mathbf{f}_i \leftarrow U(R_q^l)$ for evaluation key, and commitment $\Psi_i = \{\delta_i, \epsilon_i, \zeta_i\}$, $\delta_i = H(a_i)$, $\epsilon_i = H(\mathbf{d}_i)$, $\zeta_i = H(\mathbf{f}_i)$, broadcast Ψ_i.
3. After all $\{\Psi_i\}_{i\in[k]}$ are public, p_i discloses Φ_i.
4. After receiving $\{\Phi_j\}_{j\in[k]/i}$, p_i broadcast $\{b_i,\ \mathbf{h}_i\}$, where $b_i = as_i + e_1$, $\mathbf{h}_i = \mathbf{d}s_i + \mathbf{e}_2$, $a = \sum_{i=1}^k a_i$, $\mathbf{d} = \sum_{i=1}^k \mathbf{d}_i$, $(s_i,\ e_1,\ \mathbf{e}_2) \leftarrow \chi^{l+2}$.
5. After receiving $\{b_j,\ \mathbf{h}_j\}_{j\in[k]/i}$, p_i output $\mathsf{pk}_i = (a, b)$ and evaluation key $\mathsf{evk}_i = (\mathbf{h}_i, \boldsymbol{\eta}_i, \boldsymbol{\theta}_i)$

$$b = \sum_{i=1}^k b_i \qquad \boldsymbol{\eta}_i = \mathbf{d}r_i + \mathbf{e}_3 + s_i\mathbf{g}$$

$$\boldsymbol{\theta}_i = \mathbf{f}s_i + \mathbf{e}_4 + r_i\mathbf{g} \qquad (r_i,\ \mathbf{e}_3,\ \mathbf{e}_4) \leftarrow \chi^{2l+1}$$

Re-linearization ciphertext

Multiplying two ciphertext $\mathbf{c}_1$, $\mathbf{c}_2 \in R_q^2$, which under the same private key $\mathbf{t} = (1, s)$, $s = \sum_{i=1}^k s_i$, resulting $\mathbf{c}_{\mathsf{mult}} = \mathbf{c}_1 \otimes \mathbf{c}_2 \in R_q^4$, where its corresponding private key is $\mathbf{t} \otimes \mathbf{t} = (1, s, s, s^2)$. In order to re-linearize $\mathbf{c}_{\mathsf{mult}}$, we need to construct the ciphertext of s^2 under $\mathbf{t}$. Let total evaluation key $\boldsymbol{\Pi} = (\boldsymbol{\eta}, \boldsymbol{\theta}, \mathbf{h})$.

$$\text{where } \boldsymbol{\eta} = \sum_{i=1}^k \boldsymbol{\eta}_i \quad \boldsymbol{\theta} = \sum_{i=1}^k \boldsymbol{\theta}_i \quad \mathbf{h} = \sum_{i=1}^k \mathbf{h}_i$$

Let $\mathbf{k} = (\mathbf{k}_0, \mathbf{k}_1),\ \mathbf{k}_0 = -\boldsymbol{\theta}\mathbf{g}^{-1}(\mathbf{h}) \in R_q^l,\ \mathbf{k}_1 = (\boldsymbol{\eta} + \mathbf{f}\mathbf{g}^{-1}(\mathbf{h})) \in R_q^l$, obviously $\mathbf{k}_0 + \mathbf{k}_1 s \approx s^2\mathbf{g}$ (omit small error). Let $\mathbf{c}_{\mathsf{mult}} = (c_0, c_1, c_2, c_3)$.

$$\begin{aligned}\langle \mathbf{c}_{mult}, \mathbf{t} \otimes \mathbf{t} \rangle &= c_0 + (c_1 + c_2)s + s^2 c_3 \\ &= c_0 + (c_1 + c_2)s + s^2 \mathbf{g}\mathbf{g}^{-1}(c_3) \\ &= c_0 + \mathbf{k}_0 \mathbf{g}^{-1}(c_3) + (c_1 + c_2 + \mathbf{k}_1 \mathbf{g}^{-1}(c_3))s.\end{aligned}$$

Let $\mathbf{c}_{\mathsf{linear}} = (c_0', c_1'),\ c_0' = c_0 + \mathbf{k}_0 \mathbf{g}^{-1}(c_3),\ c_1' = c_1 + c_2 + \mathbf{k}_1 \mathbf{g}^{-1}(c_3)$, output $\mathbf{c}_{\mathsf{linear}}$ as re-linearized ciphertext. The algorithm defines as follows:

- $\mathbf{c}_{\mathsf{linear}} \leftarrow \mathsf{Relinear}(\mathbf{c}_{\mathsf{mult}}, \{\mathsf{evk}_i\}_{i\in[k]})$: Input $\mathbf{c}_{\mathsf{mult}} \in R_q^4$, evaluation key $\{\mathsf{evk}_i\}_{i\in[k]}$, perform the following algorithem, output $\mathbf{c}_{\mathsf{linear}} = (c_0', c_1')$.

Ciphertext Relinearization

Input: $\mathbf{c}_{\mathsf{mult}} = (c_0, c_1, c_2, c_3) \in R_q^4,\ \{\mathsf{evk}_i\}_{i\in[k]} = \{\mathbf{h}_i,\ \boldsymbol{\eta}_i,\ \boldsymbol{\theta}_i\}_{i\in[k]}$

Output: $\mathbf{c}_{\mathsf{linear}} = (c_0', c_1') \in R_q^2$

1: $\boldsymbol{\eta} \leftarrow \sum_{i=1}^k \boldsymbol{\eta}_i,\ \boldsymbol{\theta} \leftarrow \sum_{i=1}^k \boldsymbol{\theta}_i,\ \mathbf{h} \leftarrow \sum_{i=1}^k \mathbf{h}_i$

2: $\mathbf{k}_0 \leftarrow -\boldsymbol{\theta}\mathbf{g}^{-1}(\mathbf{h}),\ \mathbf{k}_1 \leftarrow \boldsymbol{\eta} + \mathbf{f}\mathbf{g}^{-1}(\mathbf{h})$

3: $c_0' \leftarrow c_0 + \mathbf{k}_0 \mathbf{g}^{-1}(c_3),\ c_1' \leftarrow c_1 + c_2 + \mathbf{k}_1 \mathbf{g}^{-1}(c_3)$

4: **Output:** (c_0', c_1')

5: **End**.

Due to the sum structure of keys, the dimension of $\mathbf{t} \otimes \mathbf{t}$ is independent of participants k, thus above algorithm pulls the tensor product ciphertext back to initial dimension by one shot, and introduces less noise than those keys with concatenation structure.

6 Conclusions

For the LWE-based MKFHE in order to alleviate the overhead of the local participants, we proposed the concept of KL-MKFHE which introduced a **Key lifting** procedure, getting rid of expensive ciphertext expansion operation and construct a DGSW style KL-MKFHE under plain model. Our Scheme#1 is more friendly to local participants than previous scheme, for which the local encryption $O(N\lambda^6 L^4)$ is reduced to $O(N)$. However, to support semantic security and threshold decryption, module q is required to be $O(2^{\lambda L})$, such a large q results in high overhead of ciphertext evaluation. Reducing q while ensuring security is the future direction.

For the multi-key homomorphic scheme based on RLWE, although the computation overhead of the local participants is not large: to perform re-linearization, only one ring element needs to be encrypted, the common random string is always an insurmountable hurdle. We introduced bit commitment to ensure the independence of the $\{a_i\}_{i\in[k]}$ generated by each participant under ROM. Constructing RLWE-type MKFHE under plain model is the future direction.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC. pp. 99–108. ACM Press (May 1996)
2. Alperin-Sheriff, J., Peikert, C.: Faster bootstrapping with polynomial error. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 297–314. Springer, Heidelberg (Aug 2014)
3. Ananth, P., Jain, A., Jin, Z., Malavolta, G.: Multi-key fully-homomorphic encryption in the plain model. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 28–57. Springer, Heidelberg (Nov 2020)
4. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (Apr 2012)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012. pp. 309–325. ACM (Jan 2012)
6. Brakerski, Z., Halevi, S., Polychroniadou, A.: Four round secure computation without setup. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 645–677. Springer, Heidelberg (Nov 2017)
7. Brakerski, Z., Perlman, R.: Lattice-based fully dynamic multi-key FHE with short ciphertexts. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 190–213. Springer, Heidelberg (Aug 2016)
8. Chen, H., Dai, W., Kim, M., Song, Y.: Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 395–412. ACM Press (Nov 2019)
9. Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 409–437. Springer, Heidelberg (Dec 2017)
10. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 3–33. Springer, Heidelberg (Dec 2016)
11. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 630–656. Springer, Heidelberg (Aug 2015)
12. Cramer, R., Damgård, I.B., et al.: Secure multiparty computation. Cambridge University Press (2015)
13. Dachman-Soled, D., Gong, H., Kulkarni, M., Shahverdi, A.: Towards a ring analogue of the leftover hash lemma. Journal of Mathematical Cryptology 15(1), 87–110 (2021)
14. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012), https://eprint.iacr.org/2012/144
15. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)

16. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009)
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008)
18. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (Aug 2013)
19. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions (extended abstracts). In: 21st ACM STOC. pp. 12–24. ACM Press (May 1989)
20. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische annalen 261(ARTICLE), 515–534 (1982)
21. Lindell, Y.: Fast cut-and-choose based protocols for malicious and covert adversaries. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 1–17. Springer, Heidelberg (Aug 2013)
22. Lindell, Y., Pinkas, B.: An efficient protocol for secure two-party computation in the presence of malicious adversaries. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 52–78. Springer, Heidelberg (May 2007)
23. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Karloff, H.J., Pitassi, T. (eds.) 44th ACM STOC. pp. 1219–1234. ACM Press (May 2012)
24. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (May / Jun 2010)
25. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 35–54. Springer, Heidelberg (May 2013)
26. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 35–54. Springer (2013)
27. Micali, S., Goldreich, O., Wigderson, A.: How to play any mental game. In: Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC. pp. 218–229. ACM (1987)
28. Micciancio, D.: Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. SIAM Journal on Computing 34(1), 118–169 (2004)
29. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (Apr 2012)
30. Micciancio, D., Peikert, C.: Hardness of sis and lwe with small parameters. In: Annual Cryptology Conference. pp. 21–39. Springer (2013)
31. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: 45th FOCS. pp. 372–381. IEEE Computer Society Press (Oct 2004)
32. Mouchet, C., Troncoso-Pastoriza, J.R., Bossuat, J.P., Hubaux, J.P.: Multiparty homomorphic encryption from ring-learning-with-errors. PoPETs 2021(4), 291–311 (Oct 2021)
33. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 735–763. Springer, Heidelberg (May 2016)

34. Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 681–700. Springer, Heidelberg (Aug 2012)
35. Orlandi, C.: Is multiparty computation any good in practice? In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5848–5851. IEEE (2011)
36. Peikert, C., Shiehian, S.: Multi-key fhe from lwe, revisited. In: Theory of Cryptography Conference. pp. 217–238. Springer (2016)
37. Pinkas, B., Schneider, T., Smart, N.P., Williams, S.C.: Secure two-party computation is practical. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 250–267. Springer, Heidelberg (Dec 2009)
38. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005)
39. Rivest, R.L., Adleman, L., Dertouzos, M.L., et al.: On data banks and privacy homomorphisms. Foundations of secure computation 4(11), 169–180 (1978)
40. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
41. Stehlé, D., Steinfeld, R.: Making ntru as secure as worst-case problems over ideal lattices. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 27–47. Springer (2011)
42. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (May / Jun 2010)
43. Yao, A.C.C.: Protocols for secure computations (extended abstract). In: 23rd FOCS. pp. 160–164. IEEE Computer Society Press (Nov 1982)
44. Yao, A.C.C.: How to generate and exchange secrets (extended abstract). In: 27th FOCS. pp. 162–167. IEEE Computer Society Press (Oct 1986)