Key lifting : Multi-key Fully Homomorphic Encryption in plain model without noise flooding

Xiaokang Dai[1,2] Wenyuan Wu[✉,2] and Yong Feng[2]

[1] University of Chinese Academy of Sciences, Beijing, 100049 China
[2] Chongqing Key Laboratory of Automated Reasoning and Cognition,Chongqing Institute of Green and Intelligent Technology, Chongqing, 400714, China
daixiaokang@cigit.ac.cn wuwenyuan@cigit.ac.cn yongfeng@cigit.ac.cn

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 schemes in applications is huge ciphertext expansion cost especially in data intensive scenario. For example, a boolean circuit with input length N, multiplication depth L, security parameter λ , the number of additional encryptions introduced to achieve ciphertext expansion is $O(N\lambda^6L^4)$.

In this paper we present a framework to solve this problem that we call Key-Lifting Multi-key Fully Homomorphic Encryption (KL-MKFHE). With this key lifting procedure, the number of encryptions for a local user is pulled back to $O(N)$ as single-key fully homomorphic encryption(FHE). In addition, the current MKFHE often needs to introduce noise flooding technology in the encryption or distributed decryption stage to ensure security. This results in extremely large q. In response to this problem, we propose an optimized proof method based on Rényi divergence, which removes the noise flooding technology in the encryption phase. On the other hand, in the distributed decryption phase, we prove that as long as encryption scheme is leakage-resilient, the partial decryption does not need to introduce noise flooding technique, the semantic security of fresh ciphertext can also be guaranteed, which greatly reducing the size of modulus q and the computational overhead of the entire scheme.

Moreover, we also consider RLWE for efficiency in practice. Due to the structural properties of polynomial rings, such LWE-based scheme based on Leftover hash lemma(LHL) cannot be trivially transplanted to RLWE-based scheme. We give a RLWE-based KL-MKFHE under Random Oracle Model(ROM) by introducing a bit commitment protocol.

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. [41], within a year of publishing of the RSA scheme [42]. The first truly fully homomorphic scheme was proposed by Gentry in his doctoral dissertation [21] in 2009. Based on Gentry's ideas, a series of FHE schemes have been proposed [22] [44] [10] [20] [23] [15] [14], 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.

Threshold fully holomorphic encryption(Th-FHE**).** After giving the first fully homomorphic encryption scheme, also, for the situation of multiple participants, Gentry [21] gave the corresponding strategy : first, all participants executed a secure multi-party computation protocol to obtain a common public key which all data were encrypted by, and then ciphertext evaluation was performed. After the evaluation was completed, all participants executed another secure MPC protocol to obtain the result. Obviously, the threshold was initially added to FHE only to support multiple users, while the later Th-FHE was more concerned with the flexibility of the access strategy in order to cope with different application scenarios.

In addition, there are two main ways to initialize the common public key of Th-FHE. First, assuming that there is a central authority, which generates the common public key, and disperses the private key (using Secret Sharing scheme) to each participant [25] [8]. Encryption and evaluation of data are all under the common public key, when decryption is required, the set of participants that satisfy the access control structure obtains the result through a round of interactive decryption. Boneh *et al* [8] further proposed the concept of Universal Thresholdizer, which for any fully homomorphic encryption scheme, it can be converted into a threshold fully homomorphic encryption supporting monotonic access control structure in a black-box manner.

The second method is for the parties to generate the common public key in a distributed manner, where there is no central authority. For example, Myers *et al* [36] added a threshold functionality to the integer homomorphic scheme [19], and used a distributed manner to generate the common public key and private key, without a central setup. Although adopting black box method for the construction process, the distributed key generation process was quite complicated, which consists of three steps, firstly generating the private key, then the private key of the squeezed circuit, and finally the common public key. These three processes all need to repeatedly invoke the distributed bit generation, the comparison, and the multiplication protocols. Based on the key homomorphic property, Asharov *et al* [5] generated the common public key through two rounds of interaction in a distributed manner, and the common private key was the sum of the individual private keys. In order to match the public and private keys and ensure the security of the private key, a common reference string(CRS) needed to be introduced, and decryption required everyone to provide the private key, which was actually a (n-n)Th-FHE. Damgård *et al* [18] introduced homomorphic encryption in order to optimize the preprocessing stage (such preprocessing was typically based on the classic circuit randomization technique of Beaver [7], it can be done by evaluating in parallel many small circuits of small multiplicative depth), and, a common reference string also needs to be introduced.

Multi-key Fully Homomorphic Encryption(MKFHE**).** To deal with privacy of multiple data providers, López-Alt *et al* [26] proposed the concept of MKFHE and constructed the first MKFHE scheme based on modified-NTRU [43]. Conceptually, it was an enhancement of the FHE on functionality that allowed data provider to encrypt data independent from other participants, its key generation and data encryption were done locally. To get the evaluated result, all participants were 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 [16] constructed a LWE-based MKFHE. This scheme defined the common private key as the concatenation of all private keys, and constructed a masking scheme to converts the ciphertext under individual public key to common public key by introducing CRS and circular-LWE assumptions, which only supports single-hop computation. In 2016, Mukherjee and Wich [35], Perkert and shiehian [38], Brakerski and Perlman [12] constructed MKFHE scheme based on GSW respectively. Mukherjee and Wich [35] simplified the mask scheme of [16], and focused on constructing a two-round MPC protocol. Different methods in [38] and [12] were put forward delicately to constructing a multi-hop MKFHE. Brakerski and Perlman [12] introduced bootstrapping to realize ciphertext expansion, thereby realizing the multi-hop function. Perkert and shiehian [38] realized multi-hop function through ingenious construction. It is worth mentioning that all MKFHE schemes constructed based on the LWE requires a ciphertext expansion procedure.

1.1 Motivation

We note that the biggest difference between Th-FHE and MKFHE in form is that MKFHE allows participants to encrypt data with their own public keys, and does not require interaction during the initialization phase, while Th-FHE needs to introduce a dealer or generate the common key pair in a distributed manner. Conceptually, it is clear that MKFHE is more concise, and a series of work [11] [35] [4] showed that MKFHE was an excellent base tool for building round-optimal MPC. However, despite looking attractive MKFHE actual construction involves some cumbersome operations and some unavoidable assumptions. Below we describe some details of the MKFHE scheme, and give our goal in the last paragraph of this subsection.

Ciphertext expansion is expensive : Although the MKFHE based on LWE can use LHL to remove CRS. In order to convert the ciphertext under different keys to the ciphertext under a same key(ciphertext expansion procedure), participants and the computing server need to do a lot of preparatory work. For ciphertext expansion, it is necessary to encrypt the random matrix $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$ of each ciphertext. 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 $O(N\lambda^6 L^4)$, in contrast to $O(N)$ for single-key FHE. For computing-sensitive participants, this is a lot of overhead.

CRS looks inevitable : Due to compact structure on polynomial ring and some interesting parallel algorithm such as SIMD, it is generally believed that FHE scheme based on RLWE is more efficient than FHE based on LWE. This is the reason why most current MKFHE schemes, such as [13] [34] are constructed based on RLWE.

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 [24] which can be used to get rid of CRS as [11] does. However, regularity lemma [29] over polynomial rings do not have corresponding properties, as [17] 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, it seems to be more difficult to remove CRS for RLWE-based MKFHE.

Noise flooding leads to extremely large module q : As far as we know so far, whether it is MKFHE or Th-FHE, a great noise needs to be introduced in the distributed decryption stage to cover up the partial decryption result, otherwise, private key may be leaked. In order to make the result of partial decryption simulatable, assuming that the noise accumulated after the evaluation is $\mathbf{e}_{eval}$ and the private key is $\mathbf{s}$, the flooding noise e_{sm} must satisfy $\langle \mathbf{e}_{eval}, \mathbf{s} \rangle / e_{sm} = \mathsf{negl}(\lambda)$. At this time, in order to ensure the correctness of the decryption result, module q needs to satisfy $q \geq 4e_{sm}$. Thus noise flooding results in a q that is exponentially larger than the q in a single-key FHE.

Therefore, MKFHE as a general framework, although conceptually attractive, is not suitable for some specific scenarios. Especially in the era of mobile Internet, data providers often do not trust others, and sometimes it is difficult to convince them there is a dealer or the randomness of common reference string generated by a third party. At the same time, it is unreasonable to require the data provider to do $O(N\lambda^6 L^4)$ such a large number of encryption on personal terminal.

Our goal : In response to the above problems, we propose our goal: we consider *trust-sensitive* and *computationally-sensitive* scenario with multi-users.

- Without CRS : we **do not assume** the existence of a dealer or a common reference string
- Data providers does **as many encryptions as the single-key homomorphic scheme**($O(N)$ for the circuit with input length N).
- $q = 2^{O(L)} B_\chi$ of **the same size as the single-key homomorphic scheme**, while $q = 2^{O(\lambda L)} B_\chi$ for those schemes introduced noise flooding.

1.2 Related works

Except sum type of key structure [5], concatenation structure were studied in [16] [38] [35] [12] [13] together with CRS. Ananth *et al* [3] removed CRS from a higher dimension, instead of using LHL or regularity lemma, they based on Multiparty Homomorphic Encryption and modified the initialization method of its root node to achieve this purpose, more details please refer to [3]. Brakerski *et al* [11] was the first scheme using leakage resilient property of LHL to get rid of CRS, which had the concatenation common private key structure, and ciphertext expansion was essential. All of the above schemes introduced noise flooding technology in distributed decryption phase.

We present a comparison of some properties in related work in Table 1.

Scheme	Key structure	CRS	Noise flooding	Interaction(setup phase)
THFHE [5]	Sum	✓	✓	✓
MKFHE [13]	Concatenation	✓	✓	×
MKFHE [35]	Concatenation	✓	✓	×
MKFHE [11]	Concatenation	✓	✓	✓
Scheme#1	Sum	×	×	✓
Scheme#2	Sum	ROM	✓	✓

Table 1. ✓ indicates that the corresponding operation or assumption needs to be introduced, or × indicates that it is not required.

1.3 Our Results

For *trust-sensitive* and *computationally-sensitive* scenario, we 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 [30] 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, and construct the corresponding KL-MKFHE based on ROM.

We give a brief introduction to the new concept and two scheme below and explain how we remove noise flooding in the encryption and distributed decryption phase respectively.

The concept of KL-MKFHE : Different from previous definition [35], we abandon ciphertext expansion procedure, instead, introducing a key lifting procedure which 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 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 two round interaction is allowed in Key lifting procedure.

For comparing with MKFHE, we describe the procedure of MKFHE and KL-MKFHE in Fig 1, more detailed definitions, please refer to Section 3. Here we feel compelled to explain that we are not proposing a new definition for the purpose of grandstanding or bells and whistles. The definitions of MKFHE and Th-FHE are good and suitable for many application scenarios. But as we mentioned in the previous subsection, the current schemes do not fit the scenario very well. Strictly classified by definition, the schemes(*Scheme#1*) that we proposed are neither MKFHE (we introduce interactions during initialization) nor Th-FHE (each party uses a different key to encrypt data). That's why we introduced the concept of KL-MKFHE.

Optimized Proof Method Based on Rényi Divergence : In computing tasks involving multiple parties, interactively generated keys often have some good properties, such as supporting homomorphic operations. However, since other parties are involved in the key generation process, the distribution of keys may not be as required. In order to ensure the security of the ciphertext encrypted with this key, a traditional approach which has been used in [5] [11] is to assume that the input provided by others satisfies a certain linear relationship, and to introduce large noise during encryption. However, the large noise will lead to the decrease of the efficiency of the whole scheme. Based on the result of [6, Theorem 4.2], we need to introduce neither above assumptions nor large encryption noise. However, we introduce a non-standard assumption, which we believe holds and can be used elsewhere.

We believe that this Rényi divergence-based proof method provides an alternative idea for those proofs that need to introduce strong assumptions and large noise to ensure security. More details please refer to Section 4.5.

Leakage resistance implies a smaller q : We noticed that, in the distributed decryption phase, introducing large noise to cover up the information of the private key is essentially to ensure the security of the plaintext. But adding noise is just one way to achieve it. In particular, we observe that if the encryption scheme is leakage resistant, the same purpose can be achieved by just increasing the significant bits of private key appropriately. We proved Theorem 1 in Section 4.6

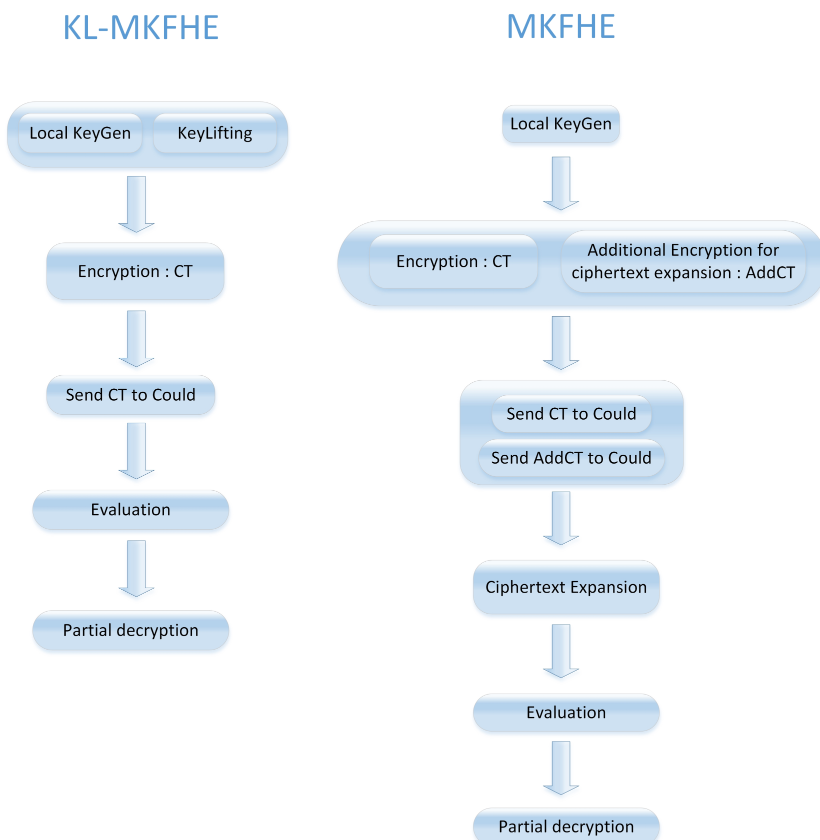

Fig. 1. The procedure of MKFHE and KL-MKFHE

Theorem 1 *If there is a multi-key homomorphic encryption scheme that is leakage resilient, then the semantic security of the initial ciphertext can be guaranteed without introducing noise flooding in the distributed decryption stage.*

Assuming that the output length of the circuit to be evaluated is W, without noise flooding, the information of private key leaked in the partial decryption results is $W \log q$ bits. As long as our encryption scheme is leakage resistant, we only need to add $W \log q$ bits to the length of the original $\mathbf{s}$, which can also ensure the security of the plaintext. Thus, the previous $q = 2^{O(\lambda L)} B_\chi$ in [5] [35] [13] can be reduced to $q = 2^{O(L)} B_\chi$ in our scheme. Refer to Section 4.6 for a detailed discussion.

***Scheme#1*: LWE-based KL-MKFHE under plain model :**
The security of *Scheme#1* is based on the LWE assumption. The common private key is the sum of the private keys of all participants. We note that previous MKFHE or Th-FHE schemes [33] [5] adopt this key structure are all based on the CRS model. Without CRS, 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.

We give a comparison with schemes [11] [38] [5] in Table 2. For detailed security and parameters, please refer to Section 4.

Scheme	Space			Time	Interaction(setup phase)	CRS
	PubKey + EvalKey	Ciphertext	Module q	Extra encryption		
MKFHE [38]	$\tilde{O}(\lambda^6 L^4(k + N\lambda^3 L^2))$	$\tilde{O}(Nk^2\lambda^6 L^4)$	$2^{O(\lambda L)} B_\chi$	$\tilde{O}(N\lambda^{14} L^9)$	×	✓
MKFHE [11]	$\tilde{O}(k^4\lambda^{15} L^{11})$	$\tilde{O}(Nk^4\lambda^8 L^6)$	$2^{O(\lambda L)} B_\chi$	$\tilde{O}(Nk^3\lambda^{15} L^{10})$	2 rounds	×
Th-FHE [5]	$\tilde{O}(\lambda^6 L^4)$	$\tilde{O}(N\lambda^6 L^4)$	$2^{O(\lambda L)} B_\chi$	×	1 rounds	✓
Scheme#1	PubKey + EvalKey : $\tilde{O}((k\lambda L + W)\lambda L^3)$ Ciphertext : $\tilde{O}(N(k\lambda L + W)^2 L^4)$ Module q : $2^{O(L)} B_\chi$			×	2 rounds	✓

Table 2. The notation $\tilde{O}$ hides logarithmic factors. Space column denotes the bit size of public, evaluation key and ciphertext; the Extra encryption column denotes the number of multiplication operations over $\mathbb{Z}_q$; λ denotes the security parameter, k participants number, B_χ the initial LWE noise; N, L, W denotes the input length, depth, and output length of the circuit respectively.

Remark : In [38] [11] [5], n represents the dimension of the LWE problem, in order to compare under the same security level, we replace n with expression in terms of λ and L. To achieve 2^λ security against known lattice attacks, one must have $n = \Omega(\lambda \log q/B_\chi)$. For our parameter settings $q = 2^{O(L)} B_\chi$, thus we would have $n = \Omega(\lambda L)$, while $n = \Omega(\lambda^2 L)$ for the previous scheme with noise flooding.

***Scheme#2*: RLWE-based KL-MKFHE under ROM :**
Same as the scheme in [13], *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 the number of participant k, so the ciphertext relinearization algorithm pulls the ciphertext after tensor product back to initial dimension by one shot, in addition, the "one shot algorithm" introduces less noise. We note that, as we mention before, regularity lemma on polynomial ring : $\mathbb{Z}(x)/x^d + 1$ does not enjoy the leakage resilient property, we have to introduce smudging noise in partial decryption phase as other RLWE-based MKFHE.

We compared with [13] in terms of memory and computational overhead, the results are shown in Table 3.

Scheme	Space		Time		Interaction(Setup phase)	CRS
	Evalkey	Ciphertext	Relinear	Mult		
MKFHE [13]	$\tilde{O}(kd)$	$\tilde{O}(kd)$	$\tilde{O}(k^2d)$	$\tilde{O}(k^2d)$	-	Yes
Scheme#2	$\tilde{O}(kd)$	$\tilde{O}(d)$	$O(1)$	$\tilde{O}(d)$	2 rounds	ROM

Table 3. The notation $\tilde{O}$ hides logarithmic factors, k denotes the number of participants; d denotes the degree of the RLWE problem. The Evalkey and Ciphertext columns denote the asymptotic storage overhead, dominated by k and d. The Relinear and Mult columns denotes the number of scalar operation over $\mathbb{Z}_q$.

2 Preliminaries

2.1 Notation:

We give the definitions of the relevant notations in Table 4. Let $\mathsf{negl}(\lambda)$ a negligible function parameterized by λ. Vectors are represented by lowercase bold

λ	security parameter	n	dimension of LWE problem
k	participants number	d	degree of RLWE problem
N	circuit input length	q	module base
L	circuit multiplicative depth		
W	circuit output length		

Table 4.

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 $\{1, \ldots, k\}$. If X is a distribution, then $a \leftarrow X$ denotes that value a is chosen according to the distribution X, or a finite set, then $a \leftarrow U(X)$ denotes that the value of a is uniformly sampled from X. For two distribution X, Y, we use $X \overset{\mathsf{stat}}{\approx} Y$ to represent X and Y are statistically indistinguishable ,while $X \overset{\mathsf{comp}}{\approx} Y$ are computationally indistinguishable.

In order to decompose elements in $\mathbb{Z}_q$ into binary, we review the Gadget matrix [31] [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 $\mathbf{g} = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g} \in \mathbb{Z}_q^{m \times ml}$, it satisfies $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.

2.2 Some background in probability theory

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

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

Lemma 1 (Smudging lemma [5]) *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 indistinguishable from that of $e_2 + e_1$ as long as $B_1/B_2 = \mathsf{negl}(\lambda)$.*

Theorem 2 (Rewritten version of [27, Theorem 5.3.2]) *Let $0 \leq k \leq n$, $f(n,k) = \sum_{i=0}^{k} \binom{n}{i}$, the sum of the first k binomial coefficients $f(n,k)$ is bounded by :*

$$f(n,k) \leq 2^{n-1} \exp\left(\frac{n-2k-2}{4(1+k-n)}\right)$$

The Rènyi divergence (in [6]) : For any two discrete probability distributions P and Q such that $\mathsf{Supp}(P) \subseteq \mathsf{Supp}(Q)$ and $a \in (1, +\infty)$, we define the The Rènyi divergence of order a by :

$$R_a(P||Q) = \left(\sum_{x \in \mathsf{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}}\right)^{\frac{1}{a-1}}$$

We omit the a subscript when $a = 2$. We define the The Rènyi divergence of order 1 and $+\infty$ by :

$$R_1(P||Q) = \exp\left(\sum_{x \in \mathsf{Supp}(P)} P(x) \log \frac{P(x)}{Q(x)}\right)$$

$$R_\infty(P||Q) = \max_{x \in \mathsf{Supp}(P)} \frac{P(x)}{Q(x)}.$$

The definitions are extended in the natural way to continuous distributions. The divergence R_1 is the (exponential of) the Kullback-Leibler divergence.

Theorem 3 ([6, Theorem 4.2]) *Let Φ, Φ' denote two distribution with $\mathsf{Supp}(\Phi) \subseteq \mathsf{Supp}(\Phi')$, and $D_0(r)$ and $D_1(r)$ denote two distributions determined by some parameter $r \in \mathsf{Supp}(\Phi')$. Let P, P' be two decision problems defined as follows :*

- *Problem P: distinguish whether input x is sampled from distribution X_0 or X_1, where*

$$X_0 = \{x : r \hookleftarrow \Phi, x \hookleftarrow D_0(r)\}, \qquad X_1 = \{x : r \hookleftarrow \Phi, x \hookleftarrow D_1(r)\}.$$

- *Problem P': distinguish whether input x is sampled from distribution X'_0 or X'_1, where*

$$X'_0 = \{x : r \hookleftarrow \Phi', x \hookleftarrow D_0(r)\}, \qquad X'_1 = \{x : r \hookleftarrow \Phi', x \hookleftarrow D_1(r)\}.$$

Assume that $D_0(\cdot)$ and $D_1(\cdot)$ satisfy the following public sampleability property: there exists a sampling algorithm S with run-time T_S such that for all (r, b), given any sample x from $D_b(r)$:

- *$S(0, x)$ outputs a fresh sample distributed as $D_0(r)$ over the randomness of S,*
- *$S(1, x)$ outputs a fresh sample distributed as $D_1(r)$ over the randomness of S.*

Then, given a T-time distinguisher $\mathcal{A}$ for problem P with advantage ϵ, we can construct a distinguisher $\mathcal{A}'$ for problem P' with run-time and distinguishing advantage, respectively, bounded from above and below by (for any $a \in (1, +\infty]$):

$$\frac{64}{\epsilon^2} \log\left(\frac{8R_a(\Phi||\Phi')}{\epsilon^{a/(a-1)+1}}\right) \cdot (T_S + T) \qquad \text{and} \qquad \frac{\epsilon}{4 \cdot R_a(\Phi||\Phi')} \cdot \left(\frac{\epsilon}{2}\right)^{\frac{a}{a-1}}.$$

2.3 Gaussian distribution on Lattice

Definition 2 *Let $\rho_s(\mathbf{x}) = \exp(-\pi||\mathbf{x}/s||^2)$ be a Gaussian function scaled by a factor of $s > 0$. Let $\Lambda \subset \mathbb{R}^n$ be a lattice, and $\mathbf{c} \in \mathbb{R}^n$. The discrete Gaussian distribution $D_{\Lambda+\mathbf{c},s}$ with support $\Lambda + \mathbf{c}$ is defined as :*

$$D_{\Lambda+\mathbf{c},s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x})}{\rho_s(\Lambda + \mathbf{x})}$$

Smoothing parameter : We recall the definition of the smoothing parameter from [32].

Definition 3 *For a lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $r > 0$ such that $\rho_{1/r}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \epsilon$.*

Lemma 2 (Special case of [32, Lemma 3.3]) *For any* $\epsilon > 0$,

$$\eta_\epsilon(\mathbb{Z}^n) \leq \sqrt{\ln(2n(1+1/\epsilon))/\pi}.$$

In particular, for any $\omega(\sqrt{\log n})$ *function, there is a negligible* $\epsilon = \epsilon(n)$ *such that* $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$.

Lemma 3 (Simplified version of [37, Theorem 3.1]) *Let* $\epsilon > 0, r_1, r_2 > 0$ *be two Gaussian parameters, and* $\Lambda \subset \mathbb{Z}^m$ *be a lattice. If* $\frac{r_1 r_2}{\sqrt{r_1^2+r_2^2}} \geq \eta_\epsilon(\Lambda)$, *then*

$$\Delta(\mathbf{y}_1 + \mathbf{y}_2, \mathbf{y}') \leq 8\epsilon$$

where $\mathbf{y}_1 \leftarrow \mathcal{D}_{\Lambda,r_1}$, $\mathbf{y}_2 \leftarrow \mathcal{D}_{\Lambda,r_2}$, *and* $\mathbf{y}' \leftarrow \mathcal{D}_{\Lambda,\sqrt{r_1^2+r_2^2}}$.

Lemma 4 (in [1]) *Let* χ *denote the Gaussian distribution with standard deviation* σ *and mean zero. Then, for all* $C > 0$, *it holds that:*

$$\Pr[e \leftarrow \chi : |e| > C \cdot \sigma] \leq \frac{2}{C\sqrt{2\pi}} \exp\{-\frac{C^2}{2}\}.$$

2.4 The Learning With Error(LWE) Problem

The Learning With Error problem was introduced by Regev [40].

Definition 4 (Decision-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* $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n}) \quad \mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e}$, *where* $\quad \mathbf{s} \leftarrow U(\mathbb{Z}_q^n) \quad \mathbf{e} \leftarrow \chi^m$

Regev [40] proved that for certain module q and Gaussian error distributions χ, the Decision-$\mathsf{LWE}_{n,q,\chi}$ problem is true as long as certain worst case lattice problems are hard to solve using a quantum algorithm. It leads to the Decision-$\mathsf{LWE}_{n,q,\chi}$ assumption $\mathcal{D}_0 \overset{\mathsf{comp}}{\approx} \mathcal{D}_1$.

2.5 The Ring Learning With Error(RLWE) Problem

Lyubaskevsky, Peikert and Regev defines the Decision-RLWE problem in [28] as follows:

Definition 5 (Decision-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 Decision-*$\mathsf{RLWE}_{d,q,\chi}$ *problem is to distinguish the following distribution:*

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

A reduction was given in [28] from the $\mathsf{RLWE}_{d,q,\chi}$ problem to the $\mathsf{Gap\text{-}SVP}$ problem on an ideal lattice, which is now generally considered to be intractable. Specially, Lyubashevsky *et al* [28] indicated that The $\mathsf{RLWE}_{d,q,\chi}$ problem is also infeasible when s is sampled from noise distribution χ. In homomorphic encryption, this property is especially popular, because the low-norm s introduces less noise during homomorphic computation.

2.6 Dual-GSW(DGSW) Encryption scheme

The DGSW scheme [11] 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 bounded by B_χ, module $q = \mathsf{poly}(n) \cdot B_\chi$, 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 $u \in \{0, 1\}$, 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 $\mathbf{E} \leftarrow \chi^{m \times w}$, 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 $\mathbf{w} = (0, \ldots, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, $v = \langle \mathbf{tC}, \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, 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 [11]. 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 [11] mentioned, DGSW scheme makes it more convenient to use the leakage resilient property of LHL to remove CRS.

2.7 Multi-Key Fully Homomorphic Encryption

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 6 *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 parameter* pp. *We assume that all algorithm take* pp *as input.*
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp}, \mathsf{crs})$ *: Input* pp*, common reference string* crs *(generated by a third party or random oracle), output a key pair for participant i.*
- $c_i \leftarrow \mathsf{Enc}(\mathsf{pk}_i, u_i)$ *: Input pk_i and plaintext u_i, output ciphertext c_i.*
- $v_i \leftarrow \mathsf{Enc}(\mathsf{pk}_i, r_i)$*: Input pk_i and the random r_i used in ciphertext c_i, output auxiliary ciphertext v_i.*
- $\bar{c}_i \leftarrow \mathsf{Expand}(\{\mathsf{pk}_i\}_{i\in[k]}, v_i, c_i)$*:Input the ciphertext c_i of participant i, the public key set $\{\mathsf{pk}_i\}_{i\in[k]}$ of all participants, auxiliary ciphertext v_i, output expanded ciphertext $\bar{c}_i$ which is under $f(\mathsf{sk}_i, \dots \mathsf{sk}_k)$ whose structure is undefined.*
- $\bar{c}_{eval} \leftarrow \mathsf{Eval}(\mathcal{S}, \mathcal{C})$*:Input circuit $\mathcal{C}$, the set of all ciphertext $\mathcal{S} = \{\bar{c}_i\}_{i\in[N]}$ 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 $\bar{c}_{eval}$, private key function $f(\mathsf{sk}_1 \dots \mathsf{sk}_k)$, output u (This is usually a distributed process).*

Remark : Although the definition of MKFHE in [26] does not contain auxiliary ciphertext v_i and ciphertext expansion procedure, in fact, the works [35] [39] [16] include this procedure to support homomorphic operations. This procedure seems to be essential, and we list it here for comparison with KL-MKFHE. The common private key depends on $\{\mathsf{sk}_i\}_{i\in[k]}$, f is a certain function, which is not unique, for example, it can be the concatenation of all keys or the sum of all keys.

Properties implicated 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)

In order to cope with *computationally-sensitive* and *trust-sensitive scenarios*, we avoid expensive ciphertext expansion procedure and introduce a relatively simple **Key lifting** procedure to replace it. 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 7 *A KL-MKFHE scheme is a tuple of probabilistic polynomial time algorithm* $(\mathsf{Init}, \mathsf{Gen}, \mathsf{KeyLifting}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$*, which can be divided into two phases, online phase:* $\mathsf{KeyLifting}$ *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.
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp})$*:Input public parameter* pp*, output the key pair of participant* i
- $\{\mathsf{hk}_i\}_{i\in[k]} \leftarrow \mathsf{KeyLifting}(\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]})$*: Input key pair* $\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]}$ *of all participants, output the hybrid key* $\{\mathsf{hk}_i\}_{i\in[k]}$ *of all* i*. (online phase : two round interaction)*
- $c_i \leftarrow \mathsf{Enc}(\mathsf{hk}_i, u_i)$*: Input plaintext* 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* $\hat{c}$
- $u \leftarrow \mathsf{Dec}(\hat{c}, f(\mathsf{sk}_1 \ldots \mathsf{sk}_k))$*: Input evaluated ciphertext* $\hat{c}$*,* $f(\mathsf{sk}_1 \ldots \mathsf{sk}_k)$*, output* $\mathcal{C}(u_i)_{i\in[N]}$*.(online phase :one round interaction)*

Remark : KL-MKFHE does not need ciphertext expansion procedure, indeed, the input 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 KL-MKFHE is locally computationally compact if the participants do the same number of encryptions as the single-key FHE scheme.*

Two round interaction : *Only two 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 KL-MKFHE scheme is correct if for a given security parameter* λ*, circuit depth* L*, participants* k*, we have the following*

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

probability is negligible, where $\mathcal{C}$ *is a circuit with input length* N *and depth length less than or equal to* L*. A KL-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 without noise flooding

Our first scheme is based on DGSW, please refer to Section 2.6 for details. In this section, we first introduce the key lifting process, then describe the entire scheme, and finally give parameter analysis and security proof.

4.1 Key lifting procedure

Following the definition of KL-MKFHE, the hybrid keys $\{\mathsf{hk}_i\}_{i\in[k]}$ which are obtained by $\mathsf{KeyLifting}(\cdot)$ algorithm are different from each other. Each participant encrypts his own plaintext u_i by hk_i and get $\mathbf{C}_i$. The ciphertexts $\{\mathbf{C}_{i\in[N]}\}$ can be used to evaluation without extra computation by claim 1. 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 $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{s}_i \leftarrow U\{0,1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i \mod q$. Assuming there is a broadcast channel, all participants are engaged in the following two interaction :
  - First round : i broadcasts $\mathbf{A}_i$ and receives all $\{\mathbf{A}_j\}_{j\in[k]\setminus i}$.
  - Second round : i generates and broadcasts $\{\mathbf{b}_{i,j}\}_{j\in[k]}$, where $\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j \mod q$

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

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

Claim 1 *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 :*

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

we have(omit small error) :

$$\bar{\mathbf{t}}\mathbf{C}_i \approx u_i\bar{\mathbf{t}}\mathbf{G}, \quad \bar{\mathbf{t}}\mathbf{C}_j \approx u_j\bar{\mathbf{t}}\mathbf{G}$$
$$\bar{\mathbf{t}}(\mathbf{C}_i + \mathbf{C}_j) \approx (u_i + u_j)\bar{\mathbf{t}}\mathbf{G}, \quad \bar{\mathbf{t}}\mathbf{C}_i\mathbf{G}^{-1}(\mathbf{C}_j) \approx (u_iu_j)\bar{\mathbf{t}}\mathbf{G}$$

Proof. According to the construction of $\mathsf{KeyLifting}(\cdot)$ we have :

$$\bar{\mathbf{t}}\mathbf{C}_i = \left(\sum_{i=1}^{k} -\mathbf{s}_i, 1\right)\left[\begin{pmatrix}\mathbf{A}_i\\ \sum_{j=1}^{k}\mathbf{b}_{j,i}\end{pmatrix} + \mathbf{E}_i + u_i\mathbf{G}\right] = \bar{\mathbf{t}}\mathbf{E}_i + u_i\bar{\mathbf{t}}\mathbf{G} \approx u_i\bar{\mathbf{t}}\mathbf{G}.$$

Similarly, $\bar{\mathbf{t}}\mathbf{C}_j \approx u_j\bar{\mathbf{t}}\mathbf{G}$, and $\bar{\mathbf{t}}(\mathbf{C}_i + \mathbf{C}_j) \approx (u_i + u_j)\bar{\mathbf{t}}\mathbf{G}$

$$\bar{\mathbf{t}}\mathbf{C}_i\mathbf{G}^{-1}(\mathbf{C}_j) \approx u_i\bar{\mathbf{t}}\mathbf{G}\mathbf{G}^{-1}(\mathbf{C}_j) \approx u_i\bar{\mathbf{t}}\mathbf{C}_j \approx (u_iu_j)\bar{\mathbf{t}}\mathbf{G}$$

■

Therefore, although $\mathbf{C}_i$ and $\mathbf{C}_j$ are encrypted by different hybrid keys, they correspond to the same decryption key $\bar{\mathbf{t}}$ and support homomorphic evaluation without extra modification.

There are two main security issues about $\mathsf{KeyLifting}(\cdot)$. First, semi-malicious adversary may generates matrix $\mathbf{A}$ with trapdoor, then $\mathbf{s}$ is leaked. Second, semi-malicious adversary j may generate $\mathbf{b}_{j,i}$ adaptively after seeing $\mathbf{b}_{i,i}$, then the hybrid key $\mathbf{b}_i$ of participant i may not distributed as requirement.

We note that as long as our encryption scheme is leakage-resistant, properly lengthening private key $\mathbf{s}$ can guarantee the semantic security of the scheme even if part of $\mathbf{s}$ is leaked. For the second problem, the general solution is to assume that $\mathbf{b}_{j,i}$ generated by adversary j satisfies the linear relationship $\mathbf{b}_{j,i} = \mathbf{s}_j\mathbf{A}_i$, $\mathbf{s}_j \in \{0,1\}^{m-1}$, and introduce a large noise in the encryption process to ensure security. Large encryption noise leads to large modulus q, which further leads to high computational and communication overhead. In order to alleviate this problem, we proposed an optimized proof method based on Rényi divergence which do not need the above assumptions, and do not need to introduce large noise in the encryption process, greatly reducing q. For more details, please refer to Section 4.5 .

4.2 The entire scheme

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

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L, 1^W)$: Let λ be security parameter, L circuit depth, W circuit output length, lattice dimension $n = n(\lambda, L)$, noise distribution χ over $\mathbb{Z}$, $e \leftarrow \chi$, where $|e|$ is bounded by B_χ with overwhelming probability, modulus $q = 2^{O(L)}B_\chi$, $k = \mathsf{poly}(\lambda)$, $m = (kn + W)\log q + \lambda$, 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 i, where $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{s}_i \leftarrow U\{0,1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i \mod q$.
- $\mathsf{hk}_i \leftarrow \mathsf{KeyLifting}(\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]})$: 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 $u_i \in \{0,1\}$, output ciphertext $\mathbf{C}_i = \begin{pmatrix}\mathbf{A}_i \\ \mathbf{b}_i\end{pmatrix}\mathbf{R} + \mathbf{E} + u_i\mathbf{G}$, where $\mathbf{R} \leftarrow U(\mathbb{Z}_q^{n\times ml})$, $l = \lceil \log q \rceil$, $\mathbf{E} \leftarrow \chi^{m\times ml}$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g}$ is a gadget matrix.

– $\mathbf{C}^{(L)} \leftarrow \mathsf{Eval}(S, \mathcal{C})$: Input the ciphertext set $S = \{\mathbf{C}_i\}_{i\in[N]}$ which are encrypted by hybrid key $\{\mathsf{hk}_i\}_{i\in[k]}$, circuit $\mathcal{C}$ with input length N, depth L, output $\mathbf{C}^{(L)}$.

Homomorphic addition and multiplication

Let $\mathbf{C}_i$, $\mathbf{C}_j$ be ciphertext under hybrid key hk_i and hk_j respectively, by claim 1, we have the following results.

– $\mathbf{C}_{\mathsf{add}} \leftarrow \mathsf{Add}(\mathbf{C}_i, \mathbf{C}_j)$: Input ciphertext $\mathbf{C}_i$, $\mathbf{C}_j$, output $\mathbf{C}_{\mathsf{add}} = \mathbf{C}_i + \mathbf{C}_j$, which $\bar{\mathbf{t}}\mathbf{C}_{\mathsf{add}} \approx (u_i + u_j)\bar{\mathbf{t}}\mathbf{G}$
– $\mathbf{C}_{\mathsf{mult}} \leftarrow \mathsf{Mult}(\mathbf{C}_i, \mathbf{C}_j)$: Input ciphertext $\mathbf{C}_i,\ \mathbf{C}_j$, output $\mathbf{C}_{\mathsf{mult}} = \mathbf{C}_i\mathbf{G}^{-1}(\mathbf{C}_j)$, which $\bar{\mathbf{t}}\mathbf{C}_{\mathsf{mult}} \approx u_iu_j\bar{\mathbf{t}}\mathbf{G}$

Distributed decryption Similar to [35], the decryption procedure is a distributed procedure :

– $\gamma_i \leftarrow \mathsf{LocalDec}(\mathbf{C}^{(L)}, \mathbf{s}_i)$: Input $\mathbf{C}^{(L)}$, let $\mathbf{C}^{(L)} = \begin{pmatrix} \mathbf{C}_{up} \\ \mathbf{c}_{low} \end{pmatrix}$, where $\mathbf{C}_{up}$ is the first $m-1$ rows of $\mathbf{C}^{(L)}$, and $\mathbf{c}_{low}$ is last row of $\mathbf{C}^{(L)}$. i computes $\gamma_i = \langle -\mathbf{s}_i,\ \mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T)\rangle$, where $\mathbf{w} = (0, \dots, 0, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, then i broadcast γ_i
– $u_L \leftarrow \mathsf{FinalDec}(\{\gamma_i\}_{i\in[k]})$: After receiving $\{\gamma_i\}_{i\in[k]}$, let $\gamma = \sum_{i=1}^k \gamma_i + \langle \mathbf{c}_{low},\ \mathbf{G}^{-1}(\mathbf{w}^T)\rangle$, output $u_L = \lceil \frac{\gamma}{q/2} \rfloor$

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 i uses hybrid key hk_i to encrypt $\mathbf{s}_i$ to obtain evaluation key evk_i. Because evk_i and $\mathbf{C}^{(L)}$ are both ciphertexts under $\mathbf{t} = (-\sum_{i=1}^k \mathbf{s}_i, 1)$, homomorphic evaluation of the decryption circuit could be executed directly as $\mathbf{C}^{(L)}$ 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 $\mathbf{C}^{(L)}$ is the ciphertext under $\mathbf{t}$, but $\{\mathsf{evk}_i\}_{i\in[k]}$ are the ciphertext under $\{\mathbf{t}_i\}_{i\in[k]}$. In order to expand $\{\mathsf{evk}_i\}_{i\in[k]} \rightarrow \{\widehat{\mathsf{evk}_i}\}_{i\in[k]}$, participant i needs to encrypt the random matrix of the ciphertext corresponding to evk_i. The extra encryption of i need to done locally is $O(\lambda^9 L^6)$.

4.4 Correctness analysis

To illustrate the correctness of *Scheme#1*, we first study the accumulation of noise. For fresh ciphertext $\mathbf{C} = \begin{pmatrix} \mathbf{A}_i \\ \mathbf{b}_i \end{pmatrix}\mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} + u\mathbf{G}$ under $\bar{\mathbf{t}}$, it holds that

$\bar{\mathbf{t}}\mathbf{C} = \mathbf{e}_1 - \mathbf{s}\mathbf{E}_0 + u\bar{\mathbf{t}}\mathbf{G}$. Let $\mathbf{e}_{init} = \mathbf{e}_1 - \mathbf{s}\mathbf{E}_0$, after L depth circuit evaluation :

$$\bar{\mathbf{t}}\mathbf{C}^{(L)} = \mathbf{e}_L + u_L\bar{\mathbf{t}}\mathbf{G} \tag{1}$$

According to the noise analysis of GSW in [23], the noise $\mathbf{e}_L$ in $\mathbf{C}^{(L)}$ is bounded by $(ml)^L\mathbf{e}_{init}$. By the distributed decryption of *Scheme#1* we have :

$$\begin{aligned}\gamma &= \sum_{i=1}^{k}\gamma_i + \langle \mathbf{c}_{low}, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle = \langle \sum_{i=1}^{k} -\mathbf{s}_i, \mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T)\rangle + \langle \mathbf{c}_{low}, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle \\ &= \bar{\mathbf{t}}\mathbf{C}^{(L)}\mathbf{G}^{-1}(\mathbf{w}^T) = \langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + u_L\lceil\frac{q}{2}\rceil\end{aligned}$$

In order to decrypt correctly, it requires $\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle < \frac{q}{4}$. For *Scheme#1*'s parameter settings, we have :

$$\begin{aligned}\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle &\le l \cdot ||\mathbf{e}_L||_\infty \\ &\le l \cdot (ml)^L \cdot ||\mathbf{e}_{init}||_\infty \\ &\le l \cdot (ml)^L \cdot (km+1)B_\chi\end{aligned}$$

Thus, $\log(\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle) = \tilde{O}(L)$. For those $q = 2^{O(L)}B_\chi \ge 4\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T\rangle$, requirements are fulfilled.

4.5 Semantic Security of Encryption against Semi-Malicious Adversary

We assume that the adversary is semi-malicious, that is to say, he can generate parameters adaptively and does not need to strictly execute the steps of the protocol. For a more formal definition, please refer to [5]. First, we note that DGSW is leakage-resilient, and second, we prove *Scheme#1*'s semantic security.

DGSW is leakage-resilient

The DGSW scheme and GSW scheme is similar to Dual-Regev scheme and Regev scheme resp. It is leakage-resilient [11]. Here, for completeness, we present it. Let χ be LWE noise distribution bounded by B_χ, χ' a distribution over $\mathbb{Z}$ bounded by $B_{\chi'}$, satisfying $B_\chi/B_{\chi'} = \mathsf{negl}(\lambda)$.

Lemma 5 (in [11]) *Let* $\mathbf{A}_i \in \mathbb{Z}_q^{(m-1)\times n}$ *be uniform, and let* $\mathbf{A}_j$ *for all* $j \neq i$ *be chosen by a rushing adversary after seeing* $\mathbf{A}_i$*. Let* $\mathbf{s}_i \leftarrow \{0,1\}^{m-1}$ *and* $\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j$*. Let* $\mathbf{r} \in \mathbb{Z}_q^n$ *be uniform,* $\mathbf{e} \leftarrow \chi^{m-1}$*,* $e' \leftarrow \chi'$*. Then under the* LWE *assumption, the vector* $\mathbf{c} = \mathbf{A}_i\mathbf{r} + \mathbf{e}$ *and number* $c' = \langle \mathbf{b}_{i,i}, \mathbf{r}\rangle + e'$ *are (jointly) pseudorandom, even given the* $\mathbf{b}_{i,j}$*'s for all* $j \in [k]$ *and the view of the adversary that generated the* $\mathbf{A}_j$*'s.*

The semantic security of *Scheme#1* For a honest player i, he generates $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1)\times n})$, $\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j$ as the protocol specification, but a semi-malicious adversary may generates it adaptively. For arbitrary $\mathbf{A}_i$, the leakage-resilient property of DGSW guarantees the semantic security. Here, we deal with what happens when $\mathbf{b}_{i,j}$ generated adaptively. As mentioned above, semi-malicious adversary i may generate $\mathbf{b}_{i,j}$ adaptively after seeing $\{\mathbf{b}_{t,j}\}_{t\in[k]/i}$ in Keylifting procedure, then the hybrid key $\mathsf{hk}_j = (\mathbf{A}_j, \mathbf{b}_j = \sum_i^k \mathbf{b}_{i,j})$ of participants j may not distributed as requirement.

A common approach to guarantee the semantic security is to assume that $\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j$ and satisfies $\mathbf{s}_i \in \{0,1\}^{m-1}$, and add a large noise to the encryption process. However, the introduced large noise result in a large module q, which increasing the computation overhead of the whole scheme. Below, we describe this common approach, and subsequently, we present a solution for Rényi divergence-based optimization

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

Claim 2 *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 $\mathbf{C}_{Scheme\#1}$ by DGSW.$\mathsf{Enc}(\mathsf{pk}_{\mathsf{DGSW}}, u)$:

1. First, Challenger generates $\mathsf{pk}_{\mathsf{DGSW}}$ like the step 1 of above Game, sets :

$$\mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, 0) = \begin{pmatrix} \mathbf{A} \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}$$

and sends $\mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, 0)$ to $\mathcal{A}$.

2. After receiving $\mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, 0)$, $\mathcal{A}$ generates $\{\mathbf{s}_i\}_{i\in[k]/1}$, set $\mathbf{s}' = \sum_{i=2}^{k} \mathbf{s}_i$. We have :

$$\mathbf{s}'\mathbf{C}_0 = \mathbf{s}'(\mathbf{A}\mathbf{R} + \mathbf{E}_0) = \sum_{i=2}^{k} \mathbf{b}_i\mathbf{R} + \mathbf{s}'\mathbf{E}_0$$

$$\begin{aligned} \mathbf{C}' &= \mathsf{DGSW.Enc}(\mathsf{pk}_{\mathsf{DGSW}}, 0) + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A} \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0 \end{pmatrix} \end{aligned} \tag{2}$$

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

■

Remark: When $||\mathbf{e}_1||_\infty$ is bounded by $2^\lambda B_\chi$, according to the correct analysis in Section 4.4, the initial noise $\mathbf{e}_{init} = \mathbf{e}_1 - \mathbf{s}\mathbf{E}_0$ is bounded by $(2^\lambda + km)B_\chi$. After L-level evaluation, $\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle$ is bounded by $l \cdot (ml)^L \cdot (2^\lambda + km)B_\chi$, $\log(\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle) = \tilde{O}(\lambda + L)$. Thus result in a $q = 2^{O(\lambda+L)}B_\chi$

Rényi divergence-based optimization : In above common approach, we showed how to construct $\mathbf{C}' = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0 \end{pmatrix}$ with a given GSW ciphertext and $\{\mathbf{s}\}_{i\in[k]/1}$, $\mathbf{s}_i \in \{0,1\}^{m-1}$ which generated by the adversary. Next, we show that for each such $\mathbf{C}'$, it is a sample from some distribution. For random $\mathbf{R} \in \mathbb{Z}_q^{n\times ml}$, without loss of generality, assuming $\frac{ml}{n} = g$, we can divide $\mathbf{R}$ into g square matrices :

$$\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_g)$$

where $\mathbf{R}_i \in \mathbb{Z}_q^{n\times n}$. Similarly, for $\mathbf{E}_0 \in \mathbb{Z}_q^{(m-1)\times ml}$, $\mathbf{e}_1 \in \mathbb{Z}_q^{ml}$:

$$\begin{aligned} \mathbf{E}_0 &= (\mathbf{E}_{0,1}, \mathbf{E}_{0,2}, \cdots, \mathbf{E}_{0,g}) \\ \mathbf{e}_1 &= (\mathbf{e}_{1,1}, \mathbf{e}_{1,2}, \cdots, \mathbf{e}_{1,g}) \end{aligned}$$

where $\mathbf{E}_{0,i} \in \mathbb{Z}_q^{(m-1)\times n}$, $\mathbf{e}_{1,i} \in \mathbb{Z}_q^{n}$. Then, for the last row $\mathbf{c}' = \mathbf{b}\mathbf{R} + \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0$ of $\mathbf{C}'$, it can be expressed as:

$$\mathbf{c}' = (\mathbf{b}\mathbf{R}_1 + \mathbf{s}'\mathbf{E}_{0,1} + \mathbf{e}_{1,1}, \mathbf{b}\mathbf{R}_2 + \mathbf{s}'\mathbf{E}_{0,2} + \mathbf{e}_{1,2}, \cdots, \mathbf{b}\mathbf{R}_g + \mathbf{s}'\mathbf{E}_{0,g} + \mathbf{e}_{1,g})$$

Let $\{\mathbf{v}_i \in \mathbb{Z}_q^n\}_{i\in[g]}$ be the solution of equation :

$$\{\mathbf{v}_i\mathbf{R}_i = \mathbf{s}'\mathbf{E}_{0,i}\}_{i\in[g]}$$

Obviously, if $\mathbf{R}_i$ is random over $\mathbb{Z}_q^{n\times n}$, then $\mathbf{v}_i$ has a unique solution. Define set V :

$$V = \{\mathbf{0}^{1\times ml}, \quad (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_g)\}$$

where $\mathbf{0}^{1\times ml}$ is a zero vector of length ml. Define the distribution $\mathbf{d} \leftarrow \mathcal{D}(V)$ over set V :

$$\mathbf{d} \leftarrow \mathcal{D}(V) \quad \Pr(\mathbf{d} = \mathbf{0}^{1\times ml}) = p \quad \Pr(\mathbf{d} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_g)) = 1 - p$$

Let Φ be the distribution of hybrid key of *Scheme#1* followed by $\mathbf{0}^{1\times ml}$:

$$(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}) \leftarrow \Phi$$

which determined by $\mathsf{KeyLifting}(\cdot)$ procedure. Let $\mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$ be the joint distribution of $(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$ and the last row of *Scheme#1*'s ciphertext $\begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}$ over the randomness $\mathbf{R}$, $\mathbf{E}_0$, $\mathbf{e}_1$:

$$(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}, \mathbf{bR} + \mathbf{e}_1) \leftarrow \mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$$

Let $\mathcal{D}_1(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$ be the joint distribution of $(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$ and $\mathbf{w} \leftarrow U(\mathbb{Z}_q^{ml})$:

$$(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}, \mathbf{w}) \leftarrow \mathcal{D}_1(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})$$

Let Φ' be the distribution of hybrid key of *Scheme#1* followed by $\mathbf{d}$ which is sampled from $\mathcal{D}(V)$:

$$(\mathbf{A}, \mathbf{b}, \mathbf{d}) \leftarrow \Phi'$$

Let P, P' be two decision problems defined as follows :

– Problem P : distinguish whether input x is sampled from distribution X_0 or X_1, where

$$X_0 = \{x : (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}, \mathbf{bR} + \mathbf{e}_1) \leftarrow \mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})\}.$$
$$X_1 = \{x : (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml}, \mathbf{w}) \leftarrow \mathcal{D}_1(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1\times ml})\}.$$

– Problem P' : distinguish whether input x is sampled from distribution X_0' or X_1', where

$$X_0' = \{x : (\mathbf{A}, \mathbf{b}, \mathbf{d}) \leftarrow \Phi',$$
$$x = (\mathbf{A}, \mathbf{b}, \mathbf{d}, (\mathbf{b} + \mathbf{d}_1)\mathbf{R}_1' + \mathbf{e}_1', (\mathbf{b} + \mathbf{d}_2)\mathbf{R}_2' + \mathbf{e}_2', \cdots, (\mathbf{b} + \mathbf{d}_g)\mathbf{R}_g' + \mathbf{e}_g') \leftarrow \mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{d})\}.$$
$$X_1' = \{x : (\mathbf{A}, \mathbf{b}, \mathbf{d}) \leftarrow \Phi', \quad x = (\mathbf{A}, \mathbf{b}, \mathbf{d}, \mathbf{w}) \leftarrow \mathcal{D}_1(\mathbf{A}, \mathbf{b}, \mathbf{d})\}.$$

where $\mathbf{R}_i' \leftarrow \mathbf{U}(\mathbb{Z}_q^{n\times n})$, $\mathbf{e}_i' \leftarrow \chi^n$, $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_g)$

Thus for every $\mathbf{c}'$ of the last row of *Scheme#1*'s ciphertext :

$$\mathbf{c}' = (\mathbf{bR}_1 + \mathbf{s}'\mathbf{E}_{0,1} + \mathbf{e}_{1,1}, \mathbf{bR}_2 + \mathbf{s}'\mathbf{E}_{0,2} + \mathbf{e}_{1,2}, \cdots, \mathbf{bR}_g + \mathbf{s}'\mathbf{E}_{0,g} + \mathbf{e}_{1,g})$$

it is a sample :

$$x = (\mathbf{b} + \mathbf{d}_1)\mathbf{R}'_1 + \mathbf{e}'_1, (\mathbf{b} + \mathbf{d}_2)\mathbf{R}'_2 + \mathbf{e}'_2, \cdots, (\mathbf{b} + \mathbf{d}_g)\mathbf{R}'_g + \mathbf{e}'_g$$

of X'_0 with $\mathbf{d} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_g)$, $\mathbf{R}'_i = \mathbf{R}_i$, $\mathbf{e}'_i = \mathbf{e}_{1,i}$.

We note that $\mathbf{c}'$ only forms part of sample of X'_0. The completed sample also contains $(\mathbf{A}, \mathbf{b}, \{\mathbf{v}_i\}_{i\in[g]})$, where $(\mathbf{A}, \mathbf{b})$ is public, $\{\mathbf{v}_i\}_{i\in[g]}$ are determined by $\{\mathbf{v}_i\mathbf{R}_i = \mathbf{s}'\mathbf{E}_{0,i}\}_{i\in[g]}$ where $\mathbf{R}_i$, $\mathbf{E}_{0,i}$ is generated by challenger, $\mathbf{s}'$ is generated by adversary.

Consider the following process :

- Challenger generates DGSW ciphetext and send it to adversary
- After receiving DGSW ciphertext, adversary adaptively generates $\mathbf{s}'$, $\mathbf{C}'$, and sends $\mathbf{s}'$ to the challenger
- Challenger computes $\{\mathbf{v}_i\}_{i\in[g]}$ and sends it to adversary.
- Adversary constructs a complete X'_0 sample from $\mathbf{C}'$ and $\{\mathbf{v}_i\}_{i\in[g]}$

Note that exposing $\{\mathbf{v}_i\}_{i\in[g]}$ to adversary will reveal the linear relationship between $\mathbf{R}_i$ and $\mathbf{E}_{0,i}$. We need to ensure that after the adversary gets $\{\mathbf{v}_i\}_{i\in[g]}$, the DGSW ciphertext is still indistinguishable.

Claim 3 *For a given DGSW ciphertext* $\mathbf{C} = \begin{pmatrix}\mathbf{A}\\ \mathbf{b}\end{pmatrix}\mathbf{R} + \begin{pmatrix}\mathbf{E}_0\\ \mathbf{e}_0\end{pmatrix}$ *the adversary adaptively chooses* $\mathbf{s}' \in \mathbb{Z}_q^{m-1}$*, and sends it to the challenger. The challenger solves the equation* $\{\mathbf{v}_i\mathbf{R}_i = \mathbf{s}'\mathbf{E}_{0,i}\}_{i\in[g]}$*, where* $\mathbf{R}_i$*and* $\mathbf{E}_{0,i}$ *are the i-th blocks of* $\mathbf{R}$ *and* $\mathbf{E}_0$*, respectively, and sends* $\{\mathbf{v}_i\}_{i\in[g]}$ *to the adversary. For the probabilistic polynomial time adversary, there is :*

$$(\mathbf{A}, \mathbf{b}, \mathbf{C}, \{\mathbf{v}_i\}_{i\in[g]}) \overset{\mathsf{comp}}{\approx} (\mathbf{A}, \mathbf{b}, \mathbf{U}, \{\mathbf{v}_i\}_{i\in[g]})$$

Unfortunately, there is no way for us to give a proof of Claim 3 here, but after our cursory attempts, we think this problem is hard. Let's take a look at what is the distribution of $\{\mathbf{v}_i\}_{i\in[g]}$. For $\mathbf{v}_i\mathbf{R}_i = \mathbf{s}'\mathbf{E}_{0,i}$,thus $\mathbf{v}_i = \mathbf{s}'\mathbf{E}_{0,i}\mathbf{R}_i^{-1}$. Because $\mathbf{R}_i$ is uniform over $\mathbb{Z}_q^{n\times n}$ and $\mathbf{E}_{0,i}$ is discrete Gaussian over $\mathbb{Z}_q^{(m-1)\times n}$, so $\mathbf{E}_{0,i}\mathbf{R}_i^{-1}$ is uniform over $\mathbb{Z}_q^{(m-1)\times n}$ and unknown to adversary. Replacing $\mathbf{E}_{0,i}\mathbf{R}_i^{-1}$ with $\mathbf{U} \leftarrow U(\mathbf{Z}_q^{(m-1)\times n}$, we have :

$$\mathbf{v}_i = \mathbf{s}'\mathbf{U}$$

Therefore, when $\mathbf{s}' \neq \mathbf{0}$, $\mathbf{v}_i$ is uniform random over $\mathbb{Z}_q^n$, that is, $\mathbf{v}_i$ and $\mathbf{s}'$ are independent except at zero. Consider the following LWE sample, which is the first $m-1$ elements of the first column of the DGSW ciphertext :

$$\begin{aligned}
\langle \mathbf{a}_1, \mathbf{r}\rangle + e_1 &= c_1\\
\langle \mathbf{a}_2, \mathbf{r}\rangle + e_2 &= c_2\\
&\cdots\\
\langle \mathbf{a}_{m-1}, \mathbf{r}\rangle + e_{m-1} &= c_{m-1}
\end{aligned}$$

where $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$, $\mathbf{r}$ is the first column of $\mathbf{R}$, $\{e_i\}_{i\in[m-1]}$ is the first column of $\mathbf{E}_{0,1}$. $\{c_i\}_{i\in[m-1]}$ is the first column of $\mathbf{C}$. One possible attack method we think of is: The adversary adaptively chooses $\mathbf{s}'$, so that $\mathbf{v}_1$ is a small linear combination of $\{\mathbf{a}_i\}_{i\in[m-1]}$. But we show that $\mathbf{v}_i$ and $\mathbf{s}'$ are independent, so this approach doesn't work. However, we cannot prove that there are no other attack methods, nor can we give a reduction of it to some hard problem. Here, we take it as a non-standard assumption for everyone to think about.

If one is willing to assume that Claim 3 is true, so far, we have completed the construction of X_0' samples: that is, for each given DGSW ciphertext, after getting $\{\mathbf{v}_i\}_{i\in[g]}$ from challenger, the adversary can convert it into a sample of X_0'. Our distributions $\mathcal{D}_0(\cdot)$ and $\mathcal{D}_1(\cdot)$ also satisfy the publicly sampleable property which required by Theorem 3 : there exists a sampling algorithm S with run-time T_S such that for all (α,β), given any sample x from $D_\alpha(\beta)$:

- $S(0,x)$ outputs a fresh sample distributed as $D_0(\beta)$ over the randomness of S,
- $S(1,x)$ outputs a fresh sample distributed as $D_1(\beta)$ over the randomness of S.

Then, by Theorem 3, if given a T- time distinguisher $\mathcal{A}$ for problem P with advantage ϵ, we can construct a distinguisher $\mathcal{A}'$ for problem P' with run-time and distinguishing advantage, respectively, bounded from above and below by(for any $a\in(1,+\infty]$) :

$$\frac{64}{\epsilon^2}\log\left(\frac{8R_a(\Phi||\Phi')}{\epsilon^{a/(a-1)+1}}\right)\cdot(T_S+T) \quad and \qquad \frac{\epsilon}{4\cdot R_a(\Phi||\Phi')}\cdot\left(\frac{\epsilon}{2}\right)^{\frac{a}{a-1}}.$$

For convenience, we take $R_\infty(\Phi||\Phi')$ analysis, let :

$$R_\infty(\Phi||\Phi')=\max_{Y\in\mathsf{Supp}(\Phi)}\frac{\Phi(Y)}{\Phi'(Y)}=\frac{\Phi(\mathbf{A}_0,\mathbf{b}_0,\mathbf{0}^{1\times ml})}{\Phi'(\mathbf{A}_0,\mathbf{b}_0,\mathbf{0}^{1\times ml})}.$$

$$=\frac{\Pr(\mathbf{A}=\mathbf{A}_0,\mathbf{b}=\mathbf{b}_0)}{\Pr(\mathbf{A}=\mathbf{A}_0,\mathbf{b}=\mathbf{b}_0,\mathbf{d}=\mathbf{0}^{1\times ml})} \tag{3}$$

Because $(\mathbf{A},\mathbf{b})$ and $\mathcal{D}(V)$ are independent, thus :

$$(3)=\frac{\Pr(\mathbf{A}=\mathbf{A}_0,\mathbf{b}=\mathbf{b}_0)}{\Pr(\mathbf{A}=\mathbf{A}_0,\mathbf{b}=\mathbf{b}_0)\Pr(\mathbf{d}=\mathbf{0}^{1\times ml})}=\frac{1}{p}$$

Then, given a T- time distinguisher $\mathcal{A}$ for problem P with advantage ϵ, we can construct a distinguisher $\mathcal{A}'$ for problem P' with run-time and distinguishing advantage, respectively, bounded from above and below by:

$$\frac{64}{\epsilon^2}\log\left(\frac{8}{p\cdot\epsilon^2}\right)\cdot(T_S+T) \quad and \qquad \frac{p\cdot\epsilon^2}{8}.$$

Remark : Under the semi-honest adversary model, $\{\mathbf{s}_{j\in[k]/1}\}$ is chosen uniformly random over $\{0,1\}^{m-1}$, so Φ is a uniform random distribution over $\mathbb{Z}_q^{m\times n}$, and the security is obvious. Under the semi-malicious adversary model, the common approach assumes $\mathbf{b}_{j,i} = \mathbf{s}_j\mathbf{A}_i$ and$\{\mathbf{s}_{j\in[k]/1}\} \in \{0,1\}^{m-1}$ is chosen adaptively, and introduces large noise in the encryption process to ensure security. In Rényi divergence-based optimization, we need to introduce neither above assumptions nor large encryption noise. However, we introduce a non-standard assumption, which we believe holds and can be used elsewhere.

We believe that this Rényi divergence-based proof method provides an alternative idea for those proofs that need to introduce strong assumptions and large noise to ensure security.

4.6 Noise flooding technology VS. Leakage resilient property in partial decryption

We note that the introduction of noise flooding in the partial decryption phase is essentially to guarantee the semantic security of fresh ciphertext, and noise flooding achieves this by masking the private key information in the partial decryption noise. For partial decryption to be simulatable, the magnitude of the noise introduced needs to be exponentially larger than the noise after the homomorphic evaluation. At the same time, as mentioned in [35], masking techniques based on noise flooding can only guarantee weak simulatable properties : input all private keys $\{\mathsf{sk}_j\}_{j\in[k]/i}$ except sk_i, evaluated result u_L, ciphertext $\mathbf{C}^{(L)}$, it can simulate the local decryption result γ_i, while for stronger security requirements : input any private key set $\{\mathsf{sk}_j\}_{j\in S}$ for any subset S of $[k]$, evaluated result u_{eval} and ciphertext $\mathbf{C}^{(L)}$, to simulate $\{\gamma_i\}_{i\in U,\ U=[k]-S}$, it don't know how to achieve it.

With noise flooding : To illustrate how our approach works, let's first look at how the noise flooding technique works. Let $\mathbf{C}^{(L)} = \begin{pmatrix} \mathbf{C}_{up} \\ \mathbf{c}_{low} \end{pmatrix}$ be the ciphertext after L-layer homomorphic multiplication. With a flooding noise $e_i'' \leftarrow U[-B_{smdg}, B_{smdg}]$, introduced in $\mathsf{LocalDec}(\cdot)$, we have :

$$\gamma_i = \langle -\mathbf{s}_i, \mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T)\rangle + e_i''$$

By Equation (1) and $\mathsf{FinalDec}(\cdot)$:

$$\gamma_i = u_L\lceil \frac{q}{2}\rceil + \langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + e_i'' - \langle \mathbf{c}_{low}, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + \langle \sum_{j\neq i}^{k} \mathbf{s}_j, \mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T)\rangle$$

For a simulator $\mathcal{S}$, input $\{\mathsf{sk}_j\}_{j\in[k]/i}$, evaluated result u_L, ciphertext $\mathbf{C}^{(L)}$, output simulated γ_i'

$$\gamma_i' = u_L\lceil \frac{q}{2}\rceil + e_i'' - \langle \mathbf{c}_{low}, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + \langle \sum_{j\neq i}^{k} \mathbf{s}_j, \mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T)\rangle$$

In order to make the partial decryption process simulatable, it requires :

$$\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle + e_i'' \overset{\mathsf{stat}}{\approx} e_i''$$

For the parameter settings in [35] : $B_{smdg} = 2^{L\lambda \log \lambda} B_\chi$, $q = 2^{\omega(L\lambda \log \lambda)} B_\chi$, obviously :

$$|\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle / e_i''| = \mathsf{negl}(\lambda)$$

$$\text{thus } \gamma_i \overset{\mathsf{stat}}{\approx} \gamma_i'.$$

In short, the noise e_i'' is introduced to cover up some information(private key $\mathbf{s}_i$ and the noise $\mathbf{E}_i$ in initial ciphertext) of participant i contained in $\mathbf{e}_L$(Noise after decrypting the ciphertext of level L, $\bar{\mathbf{t}}\mathbf{C}^{(L)} = \mathbf{e}_L + u_L \bar{\mathbf{t}}\mathbf{G}$) . Thus the partial decryption result of participant i can be simulated, providing other participants information.

Without noise flooding : Through the above analysis, we point out that as long as our encryption scheme is leakage-resilient and covers the initial noise $\{\mathbf{E}_i\}_{i\in[N]}$ in $\mathbf{e}_L$, there is no need to introduce noise flood in the partial decryption stage. To illustrate what information is contained in $\mathbf{e}_L$, let's look at how $\mathbf{e}_L$ is generated. For the initial ciphertext :

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

After performing a homomorphic multiplication operation, we obtain:

$$\begin{aligned} \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \left[\begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1 + \mathbf{E}_1 + u_1\mathbf{G}\right] \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + u_1 \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{b}_2 \end{pmatrix} \mathbf{R}_2 + u_1\mathbf{E}_2 + u_1u_2\mathbf{G} \\ &= \Pi_1 + \delta_1 + u_1u_2\mathbf{G} \end{aligned}$$

where :

$$\Pi_1 = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + u_1 \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{b}_2 \end{pmatrix} \mathbf{R}_2$$

$$\delta_1 = \mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + u_1\mathbf{E}_2$$

and $\bar{\mathbf{t}}\Pi_1 = 0$, δ_1 is the noise after the first homomorphic evaluation. For the ciphertexts $\mathbf{C}_3, \mathbf{C}_4$ of the same level, we have $\mathbf{C}_3\mathbf{G}^{-1}(\mathbf{C}_4) = \Pi_1' + \delta_1' + u_3u_4\mathbf{G}$, where Π_1', δ_1' and Π_1, δ_1 have the same structure.
Let $\mathbf{C}^{(2)}$, $\mathbf{C}^{(2)'}$ be the ciphertext at level 2 :

$$\mathbf{C}^{(2)} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2), \qquad \mathbf{C}^{(2)'} = \mathbf{C}_3\mathbf{G}^{-1}(\mathbf{C}_4)$$

$$\delta_2 = \delta_1\mathbf{G}^{-1}(\mathbf{C}^{(2)'}) + u_1u_2\delta_1'$$

we have $\mathbf{C}^{(2)}\mathbf{G}^{-1}(\mathbf{C}^{(2)'}) = \Pi_2 + \delta_2 + u_1u_2u_3u_4\mathbf{G}$. For the ciphertext at level L, we have :

$$\mathbf{C}^{(L)} = \mathbf{C}^{(L-1)}\mathbf{G}^{-1}(\mathbf{C}^{(L-1)'}) = \Pi_{L-1} + \delta_{L-1} + u_{L-1}u'_{L-1}\mathbf{G}$$
$$\delta_{L-1} = \delta_{L-2}\mathbf{G}^{-1}(\mathbf{C}^{(L-1)'}) + u_{L-1}\delta'_{L-2}$$

To find out what information δ_{L-1} contains, first, we observe $\delta_1 = \mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + u_1\mathbf{E}_2$.

Lemma 6 *For the DGSW ciphertext $\mathbf{C}_1$, $\mathbf{C}_2$, let $\mathbf{C}^{(2)} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$, the noise δ_1 obtained by decrypting $\mathbf{C}^{(2)}$ is dominated by the noise $\mathbf{E}_1$ in $\mathbf{C}_1$:*

$$\delta_1 \overset{\mathsf{stat}}{\approx} \mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) \tag{4}$$

To prove the above statement, we first prove that the distribution of the sum of multiple independent and identically distributed(*iid*) discrete Gaussian is close to discrete Gaussian. The work [37] has already proved the case of two discrete Gaussian summations, while we just generalize this result to the case of multiple summations

Lemma 7 *Let $\epsilon = 2^{-\lambda}$, $\sigma > \sqrt{2}\eta_\epsilon(\mathbb{Z})$, $m = (kn+W)l$, $l = \lceil \log q \rceil$, $\{y_i\}_{i\in[ml]} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$, $y' \leftarrow \mathcal{D}_{\mathbb{Z},\sqrt{ml}\sigma}$. we have :*

$$\Delta(\sum_{i=1}^{ml} y_i, y') \leq 8ml\epsilon.$$

Proof. Let $\{y_i^{(1)}\}_{i\in[ml/2]} \leftarrow \mathcal{D}_{\mathbb{Z},\sqrt{2}\delta}$, by lemma 3 :

$$\Delta(y_1 + y_2, y_1^{(1)}) < 8\epsilon$$
$$\Delta(y_3 + y_4, y_2^{(1)}) < 8\epsilon$$
$$\cdots$$
$$\Delta(y_{ml-1} + y_{ml}, y_{\frac{ml}{2}}^{(1)}) < 8\epsilon$$

By the additivity of statistical distances (we proved it in Appendix A) we have :

$$\Delta(\sum_{i=1}^{ml} y_i, \sum_{i=1}^{\frac{ml}{2}} y_i^{(1)}) < \frac{ml}{2} \cdot 8\epsilon.$$

Let $\{y_i^{(2)}\}_{i\in[ml/4]} \leftarrow \mathcal{D}_{\mathbb{Z},2\delta}$, again by lemma 3 :

$$\Delta(y_1^{(1)} + y_2^{(1)}, y_1^{(2)}) < 8\epsilon$$

thus :

$$\Delta(\sum_{i=1}^{\frac{ml}{2}} y_i^{(1)}, \sum_{i=1}^{\frac{ml}{4}} y_i^{(2)}) < \frac{ml}{4} \cdot 8\epsilon.$$

Iterating the above process, we have :

$$\Delta(\sum_{i=1}^{ml} y_i, y') \leq \frac{ml}{2} \cdot 8\epsilon + \frac{ml}{4} \cdot 8\epsilon +, \cdots, +8\epsilon = 8ml\epsilon.$$

we complete the proof.

■

Remark: We point out that the result here is certainly not sharp since we directly exploit the results of Lemma 3, but this result already satisfies our needs. For the case of summing multiple discrete Gaussian, if one follows the path of [37], a smaller statistical distance bound should be obtained.

Here, we prove Lemma 6:

Proof. First, according to the LWE assumption, replace $\mathbf{G}^{-1}(\mathbf{C}_2)$ with $\mathbf{M} \leftarrow U\{0,1\}^{ml \times ml}$. When $u_1 = 0$, it is proved. Assuming $u_1 = 1$, let $\delta_1(i,j)$, $\mathbf{E}_1\mathbf{M}(i,j)$ be the i-th row, j-th column element of δ_1, $\mathbf{E}_1\mathbf{M}$ respectively. We have :

$$\delta_1(1,1) = z_1e_1 + z_2e_2 + \cdots + z_{ml}e_{ml} + e_{ml+1}$$
$$\mathbf{E}_1\mathbf{M}(1,1) = z_1e_1 + z_2e_2 + \cdots + z_{ml}e_{ml}$$

where $\{z_i\}_{i\in[ml]}$ is the first column of $\mathbf{M}$, $\{e_i\}_{i\in[ml]} \leftarrow D_{\mathbb{Z},\sigma}$ is the first row of $\mathbf{E}_1$, $\mathbf{E}_2(1,1) = e_{ml+1} \leftarrow D_{\mathbb{Z},\sigma}$. Suppose, the number of 1s in $\{z_i\}_{i\in[ml]}$ is r. By lemma 7 we have :

$$\Delta(\delta_1(1,1), \mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma}) \leq 8(r+1)\epsilon.$$
$$\Delta(\mathbf{E}_1\mathbf{M}(1,1), \mathcal{D}_{\mathbb{Z},\sqrt{r}\sigma}) \leq 8r\epsilon$$

For our parameter setting, $8r\epsilon \leq 8ml\epsilon = \mathsf{poly}(\lambda) \cdot 2^{-\lambda} = \mathsf{negl}(\lambda)$. Thus :

$$\delta_1(1,1) \sim \mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma}$$
$$\mathbf{E}_1\mathbf{M}(1,1) \sim \mathcal{D}_{\mathbb{Z},\sqrt{r}\sigma}$$

Let :

$$\frac{\rho_{\sqrt{r+1}\sigma}(x)}{\rho_{\sqrt{r+1}\sigma}(\mathbb{Z})} = \frac{\rho_{\sqrt{r}\sigma}(x)}{\rho_{\sqrt{r}\sigma}(\mathbb{Z})}$$

the solution $x = \sqrt{r(r+1)\ln\frac{r+1}{r}}\sigma$. The statistical distance of $\delta_1(1,1)$ and $\mathbf{E}_1\mathbf{M}(1,1)$ is :

$$\Delta(\delta_1(1,1), \mathbf{E}_1\mathbf{M}(1,1)) = \sum_{-x}^{x} \mathcal{D}_{\mathbb{Z},\sqrt{r}\sigma} - \mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma}$$
$$= 2\sum_{-\infty}^{-x} \mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma} - \mathcal{D}_{\mathbb{Z},\sqrt{r}\sigma}$$
$$< 2\sum_{-\infty}^{-x} \mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma}$$

Let $C = \sqrt{r(r+1)\ln\frac{r+1}{r}}$, By the Lemma 4 in [1], We have :

$$\begin{aligned}2\sum_{-\infty}^{-x}\mathcal{D}_{\mathbb{Z},\sqrt{r+1}\sigma} &< \frac{2}{C\sqrt{2\pi}}\exp\{-\frac{C^2}{2}\} \\ &= \frac{2}{C\sqrt{2\pi}}\exp\{-\frac{1}{2}r(r+1)\ln\frac{r+1}{r}\} \\ &= \frac{2}{C\sqrt{2\pi}}\exp\{-\frac{r+1}{2}\}\end{aligned}$$

Generally, r is distributed like the summation of ml independent identically distributed 0-1 distribution, thus $r \sim B(ml, \frac{1}{2})$. By Theorem 2,

$$\begin{aligned}\Pr(r < \lambda) &\leq \frac{1}{2}\exp\left[\frac{(ml-2\lambda-2)^2}{4(1+\lambda-ml)}\right] \\ &\leq \frac{1}{2}\exp[-ml+3(\lambda+1)] \\ &= \mathsf{negl}(\lambda)\end{aligned}$$

Thus, the statistical distance of $\delta_1(1,1)$ and $\mathbf{E}_1\mathbf{M}(1,1)$:

$$\Delta(\delta_1(1,1), \mathbf{E}_1\mathbf{M}(1,1)) < \frac{2}{C\sqrt{2\pi}}\exp\{-\frac{\lambda+1}{2}\} = \mathsf{negl}(\lambda).$$

We completed the proof, for other item of $\delta_1(i,j)$ and $\mathbf{E}_1\mathbf{M}(i,j))$ the statement also holds.

■

Theorem 1 *If there is a multi-key homomorphic encryption scheme that is leakage resilient, then the semantic security of the initial ciphertext can be guaranteed without introducing noise flooding in the distributed decryption stage.*

Proof. According to the results we proved above, the noise $\mathbf{E}_2$ of the right ciphertext $\mathbf{C}_2$ in the ciphertext $\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$ is masked by the noise $\mathbf{E}_1$ in the left ciphertext $\mathbf{C}_1$. Similarly, the noise $\mathbf{E}_4$ of $\mathbf{C}_4$ in $\mathbf{C}_3\mathbf{G}^{-1}(\mathbf{C}_4)$ is masked by the noise $\mathbf{E}_3$ of $\mathbf{C}_3$ on the leftside. For the noise $\delta_2 = \delta_1\mathbf{G}^{-1}(\mathbf{C}^{(2)'}) + u_1u_2\delta_1'$ of the third level, δ_1' is masked by δ_1, and similarly the noise $\delta_{L-1} = \delta_{L-2}\mathbf{G}^{-1}(\mathbf{C}^{(L-2)'}) + u_{L-2}\delta_{L-2}'$ of the L-th level, δ_{L-2}' is masked by δ_{L-2}. We illustrate this continuous process in Figure 2.

If the circuit with input length N and depth L, as long as $L > \log N$, then the noise δ_{L-1} of the ciphertext $\mathbf{C}^{(L)}$ of the L-th level only contains the information of noise $\mathbf{E}_t(t \in [N])$ in a certain initial ciphertext. At this point, we only need to left-multiply $\mathbf{C}^{(L)}$ by a ciphertext $Enc(1)$ whose plaintext is 1, and let $\mathbf{C}_{clear} = Enc(1)\mathbf{G}^{-1}(\mathbf{C}^{(L)})$. Thus, the noise δ_{clear} in $\mathbf{C}_{clear}$ does not contain any information about the noise $\{\mathbf{E}_i\}_{i\in[N]}$ in the initial ciphertext $\{\mathbf{C}_i\}_{i\in[N]}$. Decrypting $\mathbf{C}_{clear}$, we have :

$$\mathbf{t}\mathbf{C}_{clear}\mathbf{G}^{-1}(\mathbf{w}^T) = \mathbf{t}\delta_{clear}\mathbf{G}^{-1}(\mathbf{w}^T) + u_L\lceil\frac{q}{2}\rceil.$$

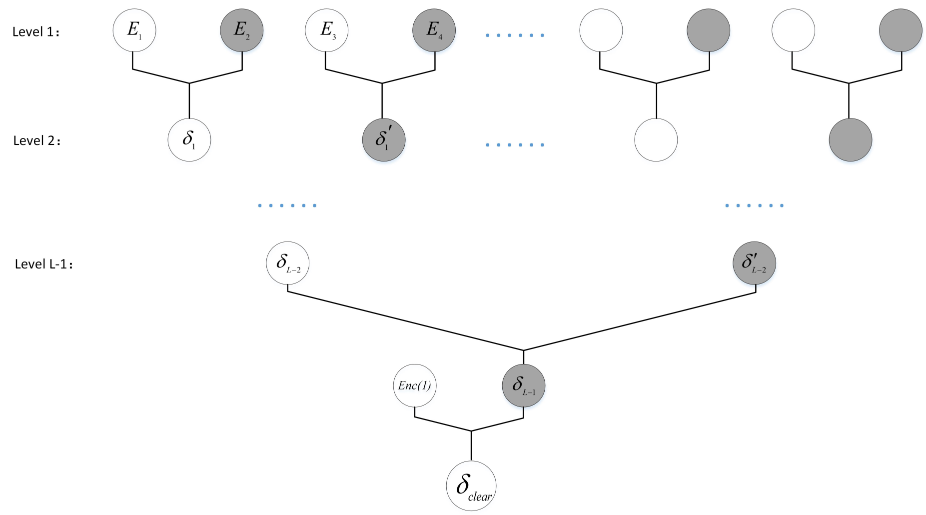

Fig. 2. Circuit

Let $\mathbf{e}_L = \mathbf{t}\delta_{clear}$, therefore, $\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T)\rangle \in \mathbb{Z}_q$ leaks participant i's private key $\mathbf{s}_i$ with at most $\log q$ bits. For a circuit with output length W, the entire partial decryption leaks $W \log q$ bits of $\mathbf{s}_i$. Because *Scheme#1* is leakage-resilient, as long as we set the key length reasonably $m = (kn + W) \log q + \lambda$, the initial ciphertext $\{\mathbf{C}_i\}_{i\in[N]}$ is semantically secure.

■

Remark : We point out that the asymmetric nature of noise in GSW ciphertext has been noted in [9] before us, but their aims and results are completely different from ours. Their purpose is to preserve the privacy of the circuit, i.e. to ensure that the final decrypted noise is independent of the circuit $\mathcal{C}$, whereas our purpose is to be independent of the initial noise. They show a discrete Gaussian version of the leftover hash lemma, whereas we show that the statistical distances of the distributions $\sum_{i=1}^{m} e_i$ and $\sum_{i=1}^{m+1} e_i$ is exponentially close to zero with m.

Here, the reader might think that doing so would result in a key that is longer than using noise flooding. We point out that as long as the output length W of circuit satisfies $W < kn(\lambda - 1)$, the length of the private key will not be longer than when using noise flooding. For $m = (kn+W) \log q+\lambda$, $q = 2^{O(L)}B_\chi$, while with noise flooding $m' = kn \log q' + \lambda$, $q' = 2^{O(\lambda L)}B_\chi$. In order to make $m < m'$, only $W < kn(\lambda - 1)$ is required, thus for circuits with small output fields, our scheme does not lead to longer keys.

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, [17] 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 notice 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.

Claim 4 *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. 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, i discloses Φ_i.
4. After receiving $\{\Phi_j\}_{j\in[k]/i}$, 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 $\{b_j,\ \mathbf{h}_j\}_{j\in[k]/i}$, 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 $\mathbf{c}_{\mathsf{mult}}=(c_0,c_1,c_2,c_3)$.

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

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 algorithm, 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 $O(N)$, and by abandoning noise flooding, it compress q from $2^{O(\lambda L)}B_\chi$ to $2^{O(L)}B_\chi$, reducing the computational scale of the entire scheme. However, the key length depends on the number of participants and the amount of leakage, which limits the application of the scheme to some extent. Further work will focus on compressing the key length.

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. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology 9(3), 169–203 (2015)
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. Ananth, P., Jain, A., Jin, Z., Malavolta, G.: Unbounded multi-party computation from learning with errors. pp. 754–781. LNCS, Springer, Heidelberg (2021)
5. 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)
6. Bai, S., Lepoint, T., Roux-Langlois, A., Sakzad, A., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. Journal of Cryptology 31(2), 610–640 (Apr 2018)
7. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (Aug 1992)
8. Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M.R., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 565–596. Springer International Publishing, Cham (2018)

9. Bourse, F., Del Pino, R., Minelli, M., Wee, H.: Fhe circuit privacy almost for free. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016. pp. 62–89. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
10. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012. pp. 309–325. ACM (Jan 2012)
11. 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)
12. 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)
13. 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)
14. 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)
15. 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)
16. 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)
17. 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)
18. Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (Aug 2012)
19. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. pp. 24–43. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
20. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012), https://eprint.iacr.org/2012/144
21. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)
22. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009)
23. 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)
24. 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)
25. Jain, A., Rasmussen, P.M.R., Sahai, A.: Threshold fully homomorphic encryption. Cryptology ePrint Archive, Paper 2017/257 (2017), https://eprint.iacr.org/2017/257, https://eprint.iacr.org/2017/257

26. 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)
27. Lovász, L., Pelikán, J., Vesztergombi, K.: Discrete mathematics: elementary and beyond. Springer Science & Business Media (2003)
28. 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)
29. 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)
30. 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)
31. 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)
32. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing 37(1), 267–302 (2007)
33. Mouchet, C., Troncoso-Pastoriza, J., Hubaux, J.P.: Computing across trust boundaries using distributed homomorphic cryptography. Cryptology ePrint Archive, Paper 2019/961 (2019), https://eprint.iacr.org/2019/961, https://eprint.iacr.org/2019/961
34. 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)
35. 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)
36. Myers, S., Sergi, M., abhi shelat: Threshold fully homomorphic encryption and secure computation. Cryptology ePrint Archive, Paper 2011/454 (2011), https://eprint.iacr.org/2011/454, https://eprint.iacr.org/2011/454
37. Peikert, C.: An efficient and parallel gaussian sampler for lattices. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. pp. 80–97. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
38. Peikert, C., Shiehian, S.: Multi-key fhe from lwe, revisited. In: Theory of Cryptography Conference. pp. 217–238. Springer (2016)
39. Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. Cryptology ePrint Archive, Report 2016/196 (2016), https://eprint.iacr.org/2016/196
40. 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)
41. 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)
42. 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)
43. 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)

44. 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)

Appendix

A the additivity of statistical distances

Claim 5 *For discrete random variables X, Y, Z with measurable space E, the statistical distance $\Delta(X,Z)$, $\Delta(X,Y)$, $\Delta(Y,Z)$ satisfy: (triangular inequality)*

$$\Delta(X,Z) \le \Delta(X,Y) + \Delta(Y,Z).$$

Proof.

$$\Delta(X,Z) = \frac{1}{2}\sum_{k\in E} |(\Pr(X=k) - \Pr(Z=k))|$$

$$\le \frac{1}{2}\sum_{k\in E} (|\Pr(X=k) - \Pr(Y=k)| + |\Pr(Y=k) - \Pr(Z=k)|)$$

$$\le \Delta(X,Y) + \Delta(Y,Z).$$

■

Claim 6 *For discrete random variables X, Y, Z with measurable space E, if X, Y, Z are independent, then :*

$$\Delta(X+Y, Y+Z) \le \Delta(X,Z)$$

Proof.

$$\Delta(X+Y, Y+Z) = \frac{1}{2}\sum_{k\in E} |\Pr(X+Y=k) - \Pr(Z+Y=k)|$$

$$= \frac{1}{2}\sum_{k\in E} |\Pr(X=k-Y) - \Pr(Z=k-Y)|$$

$$= \frac{1}{2}\sum_{k\in E} |\sum_{b\in E} (\Pr(Y=b)\Pr(X=k-b) - \Pr(Y=b)\Pr(Z=k-b)|$$

$$= \frac{1}{2}\sum_{k\in E} |\sum_{b\in E} \Pr(Y=b)(\Pr(X=k-b) - \Pr(Z=k-b))|$$

$$\le \frac{1}{2}\sum_{k\in E}\sum_{b\in E} |\Pr(Y=b)(\Pr(X=k-b) - \Pr(Z=k-b))|$$

$$= \frac{1}{2}\sum_{b\in E} \Pr(Y=b) \sum_{k\in E} |\Pr(X=k-b) - \Pr(Z=k-b)|$$

$$\le \sum_{b\in E} \Pr(Y=b) \cdot \Delta(X,Z)$$

$$= \Delta(X,Z)$$

■

Claim 7 *For discrete random variables X, Y, Z, W with measurable space E, if X, Y, Z, W are independent, then :*

$$\Delta(X+Y, Z+W) \leq \Delta(X, Z) + \Delta(Y, W).$$

Proof. by Claim 5, We have :

$$\Delta(X+Y, Z+W) \leq \Delta(X+Y, Z+Y) + \Delta(Z+Y, Z+W)$$

then, by Claim 6, We have :

$$\Delta(X+Y, Z+Y) + \Delta(Z+Y, Z+W) \leq \Delta(X, Z) + \Delta(Y, W).$$

■