

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

Fullversion : <https://eprint.iacr.org/2022/055>

Xiaokang Dai^{1,2} Wenyuan Wu^{✉,1,2} and Yong Feng^{1,2}

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 the Learning With Error assumption (LWE), usually lifts ciphertexts of different users to new ciphertexts under a common public key to enable homomorphic evaluation. The efficiency of the current Multi-key Fully Homomorphic Encryption (MKFHE) scheme is mainly restricted by two aspects:

1. **Expensive ciphertext expansion operation:** In a boolean circuit with input length N , multiplication depth L , security parameter λ , the number of additional encryptions introduced to achieve ciphertext expansion is $O(N\lambda^6 L^4)$.
2. **Noise flooding technology resulting in a large modulus q :** In order to prove the security of the scheme, the noise flooding technology introduced in the encryption and distributed decryption stages will lead to a huge modulus $q = 2^{O(\lambda L)} B_x$, which corrodes the whole scheme and leads to sub-exponential approximation factors $\gamma = \tilde{O}(n \cdot 2^{\sqrt{nL}})$.

This paper solves the first problem by presenting a framework called Key-Lifting Multi-key Fully Homomorphic Encryption (KL-MKFHE). With this *key lifting* procedure, the number of encryptions for a local user is reduced to $O(N)$, similar to single-key fully homomorphic encryption (FHE). For the second problem, based on the Rényi divergence, we propose an optimized proof method that removes the noise flooding technology in the encryption phase. Additionally, in the distributed decryption phase, we prove that the asymmetric nature of the DGSW ciphertext ensures that the noise after decryption does not leak the noise in the initial ciphertext, as long as the depth of the circuit is sufficient. Thus, our initial ciphertext remains semantically secure even without noise flooding, provided the encryption scheme is leakage-resilient. This approach significantly reduces the size of the modulus q (with $\log q = O(L)$) and the computational overhead of the entire scheme.

Keywords: Multi-key homomorphic encryption · Rényi divergence · Noise flooding · Leakage resilient cryptography.

1 Introduction

Fully Homomorphic Encryption (FHE). The concept of FHE was proposed by Rivest et al. [34] within a year of publishing the RSA scheme [35]. Gentry proposed the first truly fully homomorphic scheme in his doctoral dissertation [17] 2009. Based on Gentry’s ideas, a series of FHE schemes have been proposed [18] [37] [8] [16] [19] [13] [12], and their security and efficiency have been continuously improved. FHE is suitable for the problem of unilaterally outsourcing computations. However, for multiple data providers, data must be encrypted by a common public key to support homomorphic evaluation. Due to privacy concerns, it is unreasonable to require parties to use other people’s public keys to encrypt their data.

Multi-key Fully Homomorphic Encryption (MKFHE). To deal with the privacy of multiple data providers, López-Alt et al. [21] proposed the concept of MKFHE and constructed the first MKFHE scheme based on the modified-NTRU [36]. Conceptually, it enhanced the functionality of FHE by allowing data providers to encrypt data independently from other parties. Key generation and data encryption is done locally. To obtain the evaluated result, all parties are required to execute of a round of threshold decryption protocol.

After López-Alt et al. proposed the concept of MKFHE, many schemes were developed. In 2015, Clear and McGoldrick [14] constructed a LWE-based MKFHE scheme. This scheme defined the common private key as concatenating all private keys. It constructed a masking scheme to convert ciphertext under the individual public keys to the common public key by introducing a Common reference string (CRS) and the circular-LWE assumptions. However, this scheme only supports single-hop computation. In 2016, Mukherjee and Wichs [28], Peikert and Shiehian [31], and Brakerski and Perlman [10] constructed MKFHE schemes based on GSW, respectively. Mukherjee and Wichs [28] simplified the masking scheme of [14] and focused on constructing a two-round MPC protocol. Different methods in [31] and [10] were proposed delicately to construct a multi-hop MKFHE. It is worth mentioning that all MKFHE schemes constructed based on LWE require a ciphertext expansion procedure.

1.1 Motivation

A series of work [5, 9, 28] showed that MKFHE was an excellent base tool for building round-optimal MPC. However, despite looking attractive, the construction of MKFHE involves some cumbersome operations and unavoidable assumptions. Below we describe some details of the MKFHE scheme and state our goal in the last paragraph of this subsection.

Ciphertext expansion is expensive: Although the MKFHE based on LWE can use the Leftover hash lemma (LHL) to remove CRS, to convert the ciphertext under different keys to the ciphertext under the same key (ciphertext expansion procedure), parties and the computing server need to do much 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 an input length of N , multiplication depth of L , security parameter of 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 parties, this is much overhead.

CRS looks inevitable: Due to the compact structure of the polynomial ring and some fascinating parallel algorithms such as SIMD, it is generally believed that FHE scheme based on RLWE is more efficient than FHE based on LWE. This is why most current MKFHE schemes, such as [11, 27], 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 [20] which can be used to get rid of CRS as [9] does. However, regularity lemma [23] over polynomial rings does not have corresponding properties, as [15] mentioned: if the j -th Number theoretical transfer (NTT) coordinate of each ring element in $\mathbf{x} = (x_1, \dots, x_l)$ 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 technology resulting in a large modulus q : As far as we know so far, whether it is MKFHE or Threshold fully homomorphic encryption (Th-FHE), such as [9] [28] [14] [10] [6], a great noise needs to be introduced in encryption phase or the distributed decryption phase to ensure security; otherwise, the private key may be leaked. 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 $\langle \mathbf{e}_{eval}, \mathbf{s} \rangle / e_{sm} = \text{negl}(\lambda)$. To ensure the decryption result's correctness, modulus q needs to satisfy $q \geq 4e_{sm}$. Thus noise flooding results in a q exponentially larger than the q in a single-key FHE. Typically, in [28], the flooding noise $e_{sm} = 2^{O(L\lambda \log \lambda)} B_\chi$, the modulus $q = 2^{\omega(L\lambda \log \lambda)} B_\chi$, and the corresponding approximation factor of GapSVP_γ is $\gamma = \tilde{O}(n \cdot 2^{\lambda L})$ (which is sub-exponential in n by replacing $\lambda = \sqrt{n/L}$)³.

Therefore, although conceptually attractive, MKFHE as a general framework is not suitable for some specific scenarios. Especially in the mobile Internet era, data providers often do not trust others, and sometimes it is not easy 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 the personal terminal.

³ To achieve 2^λ security against known lattice attacks, one must have $n = \Omega(\lambda \log q / B_\chi)$

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

- Without CRS : we **do not assume** the existence of a dealer or a common reference string
- Data providers do **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 [6], concatenation structures were studied in [14] [31] [28] [10] [11] together with CRS. Ananth et al. [4] 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. Brakerski et al. [9] was the first scheme using the 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.

Recently, the work [3] has proposed an alternative approach: instead of removing it, they proposed the concept of accountability of CRS, that is, the generator of CRS should be responsible for its randomness; otherwise, the challenging party can provide a publicly verifiable proof that certifies the authority's misbehaviour. This could be an effective means of balancing authority.

We compare some properties in related work in Table 1.

Table 1. Scheme property comparison

Scheme	Key structure	CRS	Noise flooding	Interaction(setup phase)
THFHE [6]	S	✓	✓	✓
MKFHE [11]	C	✓	✓	×
MKFHE [28]	C	✓	✓	×
MKFHE [9]	C	✓	✓	✓
Our scheme	S	×	×	✓

S" and "C" in the column of Key structure represent the sum or concatenated key structure, respectively. ✓ 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* scenarios, 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.

We briefly introduce the new concept and our scheme below and explain how we remove noise flooding in the encryption and distributed decryption phases, respectively.

The concept of KL-MKFHE : Different from previous definition [28], we abandon the ciphertext expansion procedure, instead, introducing a *key lifting* procedure which at a lower cost. Informally, the *key lifting* is an interactive protocol. The input is the key pair of all parties. After the protocol is executed, the "lifted" key pair is output, called the hybrid key, which has such properties:

- *Everyone's hybrid key is different.*
- *The ciphertext encrypted by different hybrid keys supports homomorphic evaluation.*

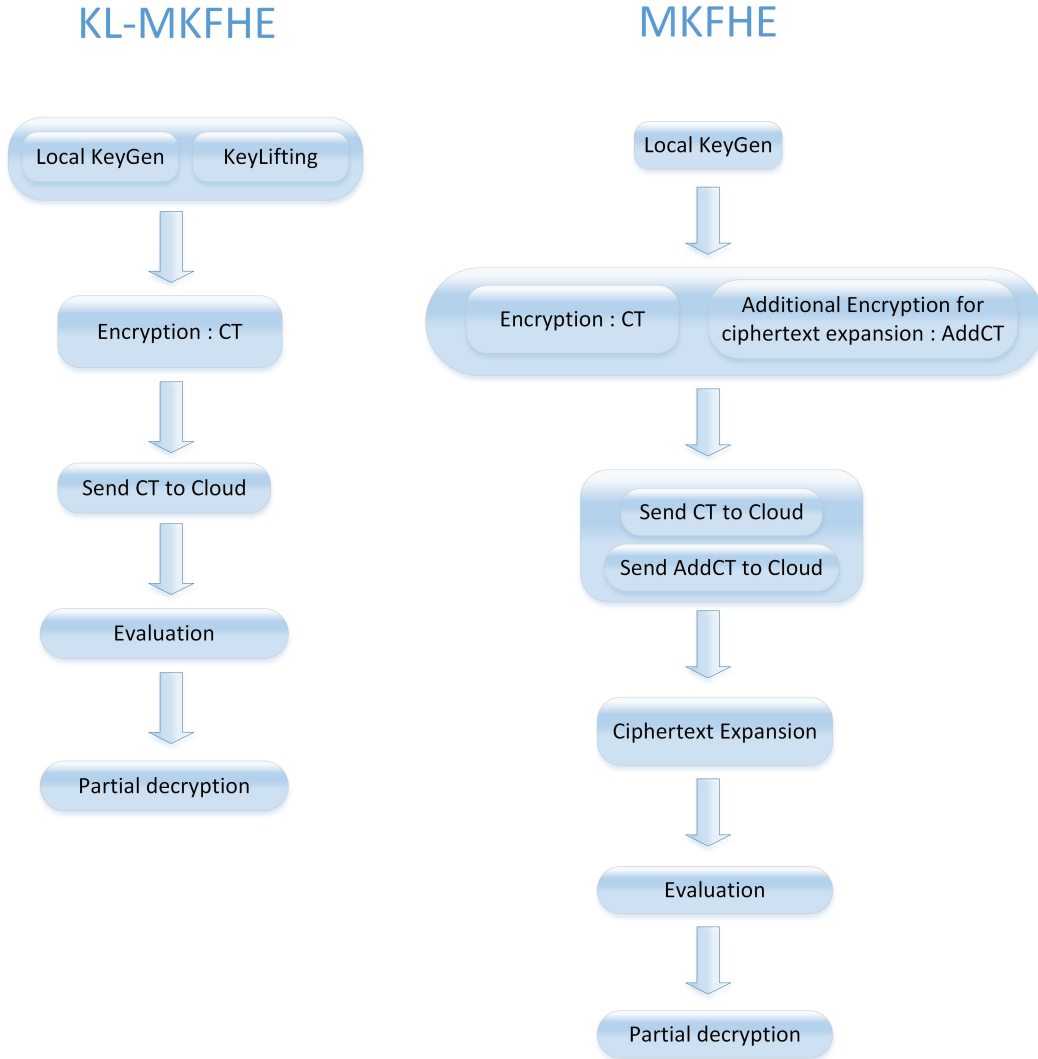
In addition to the properties that are required by MKFHE, such as *Correctness*, *Compactness*, *Semantic security*, KL-MKFHE should satisfy the following three additional properties :

- **Plain model :** *No trusted setup or Common Reference String*

- **Locally Computationally Compactness** : For a computational task corresponds to a Boolean circuit with an input length of N , a *KL-MKFHE* scheme is locally computationally compact if the parties do $O(N)$ encryptions as the single-key FHE scheme.
- **Low round complexity** : Only two round interaction is allowed in the key lifting procedure.

For comparison with MKFHE, we describe the procedure of MKFHE and KL-MKFHE in Fig 1.

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



Optimized security proof method based on Rényi divergence : In order to prove the security of a scheme, a routine is to construct an instance of the scheme from a well-known hard problem instance. Unfortunately, sometimes, this process does not go so smoothly. To make the constructed distribution statistically indistinguishable from the target distribution, you need to add noise distribution to smooth the gap between the two; this is where noise flooding comes into play. For example, [6] [9] adopted this method to prove security. Unfortunately, the added noise tends to be significant, reducing the scheme's efficiency.

Shi et al. [7] pointed out that Rényi divergence can also be used to distinguish problems: they proved that, under certain conditions, if there is an algorithm that can distinguish problem P , then there is an algorithm that can distinguish problem P' . Note that it does not require that the P

problem is indistinguishable from P' , which is where the Rényi divergence comes into play. Based on the result of [7, Theorem 4.2], our proof method is as follows :

1. Define the P problem as distinguishing our scheme's ciphertext from a uniform distribution.
2. Prove that for a given hard problem instance I , there exists a distribution \mathcal{D} , and a sample x of \mathcal{D} can be constructed from this instance I .
3. Define the P' problem as distinguishing \mathcal{D} from a uniform distribution

Thus, if there is an adversary who can distinguish the P problem, then he can distinguish the P' problem and can also distinguish the hard problem instance I from the uniform distribution.

We believe that this Rényi divergence-based proof method provides an alternative idea for those proofs that must introduce strong assumptions and large noise to ensure security. For example, we give the optimal proof method for the leakage-resilient of the DGSW scheme in Appendix C (without introducing large noise). For more details, please refer to Section 5.4.

Remark: We must note that it is not easy to construct a sample x of distribution \mathcal{D} from instance I . For this reason, we also introduce a new problem, called the "interactive LWE " problem. After some attempts, we, unfortunately, cannot prove or disprove it, but we hope it is hard. We would be grateful and admired if readers could prove or disprove it. See Section 4 for more details.

Leakage resistance implies a smaller q : We noticed that the distributed decryption of the MKFHE will leak the noise accumulated after the homomorphic evaluation and the decryptor's private key. In order to ensure security, previous MKFHE, such as [6] [11] [28] [9], will add some additional noise to the distributed decryption results to cover up this part of the information. Because we only care about the security of the initial ciphertext (note that the noise after the homomorphic evaluation will leak the privacy of the circuit), as long as it can be proved that the noise of distributed decryption is independent of the noise in the initial ciphertext, and our scheme is anti-leakage, then even without adding additional noise, the semantic security of the initial ciphertext can be guaranteed.

For the Dual GSW-like scheme, we noticed that the noise after its homomorphic multiplication is very regular: let $\mathbf{C}_{\text{mult}} = \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2)$, the noise in \mathbf{C}_{mult} hardly contains the noise in \mathbf{C}_2 . Assuming that the initial ciphertext are $\{\mathbf{C}_i\}_{i \in [N]}$, and the circuit multiplication depth is L , as long as $L \geq \log N$, then the noise in the ciphertext \mathbf{C}_L of the L -th layer only contains the noise of a certain initial ciphertext. At this point, just multiply \mathbf{C}_L by a ciphertext $Enc(1)$ whose plaintext is 1, let $\mathbf{C}_{\text{clear}} = Enc(1)\mathbf{G}^{-1}(\mathbf{C}_L)$, then the noise in $\mathbf{C}_{\text{clear}}$ does not contain any noise information in the initial ciphertext $\{\mathbf{C}_i\}_{i \in [N]}$. At this point, the distributed decryption of $\mathbf{C}_{\text{clear}}$ will only leak the decryptor's private key.

Suppose our scheme is anti-leakage and predicts the amount of private key leakage in the distributed decryption process in advance. In that case, we only need to cover this part of the leakage amount when the parameters are initialized. Even if no noise is added in the distributed decryption process, it can guarantee the semantic security of the initial ciphertext. The disadvantage is that the complexity of our scheme could be more circuit-dependent. However, there is no noise flooding in encryption and distributed decryption, so we can set $q = 2^{O(L)}B_\chi$ to be the same size as the single-key homomorphic scheme, where $q = 2^{O(\lambda L)}B_\chi$ in [6] [28] with noise flooding technology (Correspondingly, the approximation factor of $Gapsvp_\gamma$ is reduced to $\gamma = \tilde{O}(n \cdot 2^L)$). Refer to Section 5.5 for a detailed discussion.

Our scheme: LWE-based KL-MKFHE under plain model :

Our scheme is based on the LWE assumption. The common private key is the sum of the private keys of all parties. Previous MKFHE or Th-FHE schemes [26] [6] adopt this key, all based on the CRS model. For a circuit with an input length N , our scheme requires local users to perform $O(N)$ encryption operations, while it is $O(N\lambda^6L^4)$ for those schemes that require ciphertext expansion.

We give a comparison with schemes [9] [31] [6] in Table 2. Please refer to Section 5 for detailed security and parameters.

Table 2. Scheme complexity comparison

Scheme	Space			Time	Interaction(setup phase)	CRS
	PubKey + EvalKey	Ciphertext	Module q			
MKFHE [31]	$\tilde{O}(\lambda^6 L^4 (k + N\lambda^3 L^2))$	$\tilde{O}(Nk^2 \lambda^6 L^4)$	$2^{O(\lambda L)} B_\chi$	$\tilde{O}(N\lambda^{14} L^9)$	×	✓
MKFHE [9]	$\tilde{O}(k^4 \lambda^{15} L^{11})$	$\tilde{O}(Nk^4 \lambda^8 L^6)$	$2^{O(\lambda L)} B_\chi$	$\tilde{O}(Nk^3 \lambda^{15} L^{10})$	2 rounds	×
Th-FHE [6]	$\tilde{O}(\lambda^6 L^4)$	$\tilde{O}(N\lambda^6 L^4)$	$2^{O(\lambda L)} B_\chi$	×	1 rounds	✓
Our scheme	$\tilde{O}((k\lambda L + W)\lambda L^3)$	$\tilde{O}(N(k\lambda L + W)^2 L^4)$	$2^{O(L)} B_\chi$	×	2 rounds	×

The notation \tilde{O} hides logarithmic factors. The "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 parties number, B_χ the initial LWE noise; N, L, W denotes the input length, depth, and output length of the circuit respectively. In [31] [9] [6], n represents the dimension of the LWE problem, in order to compare under the same security level, we replace n with the 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.

1.4 Roadmap:

In Section 2, we define some symbols and list some commonly used definitions and results. In Section 3, we define the KL-MKFHE. In Section 4, we define a new problem. In Section 5, we construct the first KL-MKFHE scheme based on LWE.

2 Preliminaries

2.1 Notation:

We define the relevant notations in Table 3. Let $\text{negl}(\lambda)$ be a negligible function parameterized by λ .

Table 3.

λ	security parameter	n	dimension of LWE problem
k	number of parties	q	modulu base
N	circuit input length	W	circuit output length
L	circuit multiplicative depth		

Lowercase bold letters such as \mathbf{v} , unless otherwise specified, represent vectors. Vectors are row vectors by default, and matrices are represented by uppercase bold letters such as \mathbf{M} . $[k]$ denotes the set of integers $\{1, \dots, k\}$. If X is a distribution, then $a \leftarrow X$ denotes that value a 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 . Let $\Delta(X, Y)$ denote the statistical distance of X and Y . For two distributions X, Y , we use $X \stackrel{\text{stat}}{\approx} Y$ to represent X and Y are statistically indistinguishable, while $X \stackrel{\text{comp}}{\approx} Y$ are computationally indistinguishable.

In order to decompose elements in \mathbb{Z}_q into binary, we review the Gadget matrix [24] [2] here. Let $\mathbf{G}^{-1}(\cdot)$ be the computable function that for any $\mathbf{M} \in \mathbb{Z}_q^{m \times n}$, it holds that $\mathbf{G}^{-1}(\mathbf{M}) \in \{0, 1\}^{ml \times n}$, where $l = \lceil \log q \rceil$. Let $\mathbf{g} = (1, 2, \dots, 2^{l-1}) \in \mathbb{Z}_q^l$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g} \in \mathbb{Z}_q^{m \times ml}$, it satisfies $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.

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 :

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

Lemma 1 (Smudging lemma [6]) 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, Then the distribution of e_2 is statistically indistinguishable from that of $e_2 + e_1$ as long as $B_1/B_2 = \text{negl}(\lambda)$.

Theorem 1 ([22, Theorem 5.3.2]) *Let $0 \leq t \leq m$. Then the probability that out of $2m$ coin tosses, the number of heads is less than $m - t$ or large than $m + t$, is at most $e^{-t^2/(m+t)}$.*

The Rènyi divergence (in [7]) : For any two discrete probability distributions P and Q such that $\text{Supp}(P) \subseteq \text{Supp}(Q)$ where $\text{Supp}(P) = \{x : P(x) \neq 0\}$ and $a \in (1, +\infty)$, The Rènyi divergence of order a is defined by :

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

Omitting the a subscript when $a = 2$, defining the The Rènyi divergence of order 1 and $+\infty$ by :

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

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

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

Theorem 2 ([7, Theorem 4.2]) *Let Φ, Φ' denote two distribution with $\text{Supp}(\Phi) \subseteq \text{Supp}(\Phi')$, and $D_0(r)$ and $D_1(r)$ denote two distributions determined by some parameter $r \in \text{Supp}(\Phi')$. 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

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

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

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

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 $a \in (1, +\infty]$):

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

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 :*

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

Smoothing parameter : We recall the definition of the smoothing parameter from [25].

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 $\rho_{1/r}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$.*

Lemma 2 (Special case of [25, 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 [30, 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 ([1]) Let χ denote the Gaussian distribution with standard deviation σ and mean zero. Then, for all $C > 0$, it holds that:

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

2.4 The Learning With Error(LWE) Problem

The Learning With Error problem was introduced by Regev [33].

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 $\text{LWE}_{n,q,\chi}$ 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 $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$ $\mathbf{z} \leftarrow U(\mathbb{Z}_q^n)$
- \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 $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$ $\mathbf{b} = \mathbf{s}\mathbf{A} + \mathbf{e}$, where $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$ $\mathbf{e} \leftarrow \chi^m$

As shown in Regev [33] [29], the $\text{LWE}_{n,q,\chi}$ problem with χ being discrete Gaussian distribution with parameter $\sigma = \alpha q \geq 2\sqrt{n}$ is at least as hard as approximating the shortest independent vector problem(SIVP) to within a factor of $\gamma = \tilde{O}(n/\alpha)$ in *worst case* dimension n lattices. It leads to the Decision-LWE $_{n,q,\chi}$ assumption $\mathcal{D}_0 \stackrel{\text{comp}}{\approx} \mathcal{D}_1$.

2.5 Dual-GSW(DGSW) Encryption scheme

The DGSW scheme [9] and GSW scheme are similar to the Dual-Regev scheme and Regev scheme resp. Which is defined as follows:

- $\text{pp} \leftarrow \text{Gen}(1^\lambda, 1^L)$: For a given security parameter λ , circuit depth L , choose an 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 $q = \text{poly}(n) \cdot B_\chi$, Output $\text{pp} = (n, m, q, \chi, B_\chi)$ as the initial parameters.
- $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(\text{pp})$: Let $\text{sk} = \mathbf{t} = (-\mathbf{s}, 1)$, $\text{pk} = (\mathbf{A}, \mathbf{b})$, where $\mathbf{s} \leftarrow U(\{0, 1\}^{m-1})$, $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{(m-1) \times n})$, $\mathbf{b} = \mathbf{s}\mathbf{A} \pmod q$.
- $\mathbf{C} \leftarrow \text{Enc}(\text{pk}, u)$: 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 :

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \mathbf{E} + u\mathbf{G}$$

where \mathbf{G} is a gadget Matrix.

- $u \leftarrow \text{Dec}(\text{sk}, \mathbf{C})$: Input private key sk , ciphertext \mathbf{C} , let $\mathbf{w} = (0, \dots, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, $v = \langle \mathbf{t}\mathbf{C}, \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, output $u' = \lceil \frac{v}{q/2} \rceil$.

Leak resistance : Brakerski et al proved in [9] that DGSW is leak-resistant. Informally, even if part of the private key of the DGSW scheme is leaked, the DGSW ciphertext is still semantically secure. As Lemma 5 says :

Lemma 5 ([9]) *Let χ be LWE noise distribution bounded by B_χ , χ' a distribution over \mathbb{Z} bounded by $B_{\chi'}$, satisfying $B_\chi/B_{\chi'} = \text{negl}(\lambda)$. 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.*

Remark : Note that in the proof of [9], the condition for the establishment of Lemma 5 is $|\mathbf{e}/e'| = \text{negl}(\lambda)$. We point out that this condition is not required with our analytical method. We prove it in the Appendix C.

2.6 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 proposed later.

Definition 5 *Let λ be the security parameter, L be the circuit depth, and k be the number of parties. A levelled multi-key fully homomorphic encryption scheme consists of a tuple of efficient probabilistic polynomial time algorithms $\text{MKFHE}=(\text{Init}, \text{Gen}, \text{Enc}, \text{Expand}, \text{Eval}, \text{Dec})$ which defines as follows.*

- $\text{pp} \leftarrow \text{Init}(1^\lambda, 1^L)$: Input security parameter λ , circuit depth L , output system parameter pp . We assume that all algorithms take pp as input.
- $(\text{pk}_i, \text{sk}_i) \leftarrow \text{Gen}(\text{pp}, \text{crs})$: Input pp , common reference string crs (generated by a third party or random oracle), output a key pair for party i .
- $c_i \leftarrow \text{Enc}(\text{pk}_i, u_i)$: Input pk_i and plaintext u_i , output ciphertext c_i .
- $v_i \leftarrow \text{Enc}(\text{pk}_i, r_i)$: Input pk_i and the random r_i used in ciphertext c_i , output auxiliary ciphertext v_i .
- $\bar{c}_i \leftarrow \text{Expand}(\{\text{pk}_i\}_{i \in [k]}, v_i, c_i)$: Input the ciphertext c_i of party i , the public key set $\{\text{pk}_i\}_{i \in [k]}$ of all parties, auxiliary ciphertext v_i , output expanded ciphertext \bar{c}_i which is under $f(\text{sk}_i, \dots, \text{sk}_k)$ whose structure is undefined.
- $\bar{c}_{\text{eval}} \leftarrow \text{Eval}(\mathcal{S}, \mathcal{C})$: 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}
- $u \leftarrow \text{Dec}(\bar{c}_{\text{eval}}, f(\text{sk}_1 \dots \text{sk}_k))$: Input evaluated ciphertext \bar{c}_{eval} , private key function $f(\text{sk}_1 \dots \text{sk}_k)$, output u (This is usually a distributed process).

Remark : Although the definition of MKFHE in [21] does not contain auxiliary ciphertext v_i and ciphertext expansion procedure, in fact, the works [28] [32] [14] include this procedure to support homomorphic operations. This procedure seems essential; we list it here for comparison with KL-MKFHE. The common private key depends on $\{\text{sk}_i\}_{i \in [k]}$, 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 party is required in the key generation and encryption phase independently to generate their keys and complete the encryption operation without interaction between parties. 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 parties perform a round of decryption protocol.

3 Key Lifting Multi-key Fully Homomorphic Encryption

In order to cope with *computationally-sensitive* and *trust-sensitive scenarios*, we avoid expensive ciphertext expansion procedures and introduce a relatively simple *Key lifting* procedure to replace it. In addition, a tighter bound is required on the amount of local computation and parameter size; as a compromise, we allow a small amount of interaction during *Key lifting*.

Definition 6 A *KL-MKFHE* scheme is a tuple of probabilistic polynomial time algorithm (Init, Gen, KeyLifting, Enc, Eval, Dec), which can be divided into two phases, online phase: KeyLifting and Dec, where interaction is allowed between parties; local phase: Init, Gen, Enc, and Eval, whose operations do not involve interaction. These five algorithms are described as follows :

- $\text{pp} \leftarrow \text{Init}(1^\lambda, 1^L)$: Input security parameter λ , circuit depth L , output public parameters pp .
- $(\text{pk}_i, \text{sk}_i) \leftarrow \text{Gen}(\text{pp})$: Input public parameter pp , output the key pair of party i
- $\{\text{hk}_i\}_{i \in [k]} \leftarrow \text{KeyLifting}(\{\text{pk}_i, \text{sk}_i\}_{i \in [k]})$: Input key pair $\{\text{pk}_i, \text{sk}_i\}_{i \in [k]}$ of all parties, output the hybrid key $\{\text{hk}_i\}_{i \in [k]}$ of all parties. (online phase: two-round interaction)
- $c_i \leftarrow \text{Enc}(\text{hk}_i, u_i)$: Input plaintext u_i and hk_i , output ciphertext c_i
- $\hat{c} \leftarrow \text{Eval}(\mathcal{C}, S)$: Input circuit \mathcal{C} , ciphertext set $S = \{c_i\}_{i \in [N]}$, output ciphertext \hat{c}
- $u \leftarrow \text{Dec}(\hat{c}, f(\text{sk}_1 \dots \text{sk}_k))$: Input evaluated ciphertext \hat{c} , $f(\text{sk}_1 \dots \text{sk}_k)$, 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 $\text{Eval}(\cdot)$ is encrypted by parties under their hybrid key hk_i which are different among parties, however, the resulting ciphertext c_i supports homomorphic evaluation without extra modification.

we require KL-MKFHE to satisfy the following properties :

Plain model : No trusted setup or Common Reference String

Locally Computationally Compactness : For a computational task corresponds to a Boolean circuit with an input length of N , a KL-MKFHE scheme is locally computationally compact if the parties do $O(N)$ encryptions as the single-key FHE scheme.

Two round interaction : Only two round interaction is allow in $\text{KeyLifting}(\cdot)$ procedure.

The indistinguishable of initial ciphertext : Let N and W be the input and out length of a circuit, respectively. Let $\{c_i\}_{i \in [N]}$, $\{\gamma_i\}_{i \in [W]}$ be the initial ciphertext and partial decryption result respectively. The following two distributions are computationally indistinguishable for any probabilistic polynomial time adversary \mathcal{A} .

$$(\text{pp}, \{\text{pk}_i\}_{i \in [k]}, \{\text{hk}_i\}_{i \in [k]}, \{c_i\}_{i \in [N]}, \{\gamma_i\}_{i \in [W]}) \stackrel{\text{comp}}{\approx} (\text{pp}, \{\text{pk}_i\}_{i \in [k]}, \{\text{hk}_i\}_{i \in [k]}, \mathbf{U}, \{\gamma_i\}_{i \in [W]})$$

where \mathbf{U} is uniform

Correctness and Compactness : A KL-MKFHE scheme is correct if for a given security parameter λ , circuit depth L , parties k , we have the following

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

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 $\text{poly}(\lambda, L, k)$, but independent of circuit size.

4 The "interactive LWE " problem

This section introduces a new hardness problem called the "interactive LWE " problem. This problem can be seen as a variant of the standard LWE problem, where the interactive process reveals the linear relationship between the secret and the noise in the standard LWE. After some attempts, we, unfortunately, cannot prove or disprove it, but we hope it is hard. We would be grateful and admired if readers could prove or disprove it.

This new problem is introduced because we will use it in the optimization proof method based on Rényi divergence. If it is hard, we believe that it will be useful elsewhere.

Definition 7 (Interactive LWE) Let λ be security parameter, $n = n(\lambda)$, $w = w(\lambda)$, $q = q(\lambda) > 2$, $m = O(n \log q)$ be integers satisfying $n|w$. Let $\chi = \chi(\lambda)$ be a distribution defined over \mathbb{Z} , the matrix version of the $\text{LWE}_{n,q,\chi}$ samples are $(\mathbf{A}, \mathbf{B} = \mathbf{A} \cdot \mathbf{R} + \mathbf{E})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{R} \leftarrow U(\mathbb{Z}_q^{n \times w})$, $\mathbf{E} \leftarrow \chi^{m \times w}$. Consider the following Game.

1. Challenger generates the matrix version $\text{LWE}_{n,q,\chi}$ samples (\mathbf{A}, \mathbf{B}) and an uniform samples (\mathbf{A}, \mathbf{U}) , send it to an adversary \mathcal{A} .

2. After receiving (\mathbf{A}, \mathbf{B}) and (\mathbf{A}, \mathbf{U}) , \mathcal{A} adaptively chooses $\mathbf{s} \in \mathbb{Z}_q^m$, $\|\mathbf{s}\|_\infty < \sqrt{n \log q}$ and send it to Challenger.
3. After receiving \mathbf{s} , challenger computes $\{\mathbf{v}_i\}_{i \in [g]}$ by $\{\mathbf{v}_i \mathbf{R}_i = \mathbf{s} \mathbf{E}_i\}_{i \in [g]}$, where $\mathbf{R}_i \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{E}_i \in \mathbb{Z}_q^{m \times n}$ are i -th block of $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_g)$ and $\mathbf{E} = (\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_g)$ respectively, send $\{\mathbf{v}_i\}_{i \in [g]}$ to \mathcal{A} .
4. After receiving $\{\mathbf{v}_i\}_{i \in [g]}$, \mathcal{A} try to distinguish $(\mathbf{A}, \mathbf{B}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s})$ and $(\mathbf{A}, \mathbf{U}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s})$

Let $\text{Adv} = \Pr(\mathcal{A}(\mathbf{A}, \mathbf{B}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s}) = 1) - \Pr(\mathcal{A}(\mathbf{A}, \mathbf{U}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s}) = 1)$. If $\text{Adv} > \text{negl}(\lambda)$, \mathcal{A} wins, otherwise Challenger wins.

5 A KL-MKFHE scheme based on DGSW in plain model without noise flooding

Our scheme is based on DGSW. In this section, we first introduce the *key lifting* process, describe the entire scheme, and finally give parameter analysis and security proof.

5.1 Key lifting procedure

Following the definition of KL-MKFHE, the hybrid keys $\{\text{hk}_i\}_{i \in [k]}$ which are obtained by $\text{KeyLifting}(\cdot)$ algorithm are different from each other. Each party encrypts his plaintext u_i by hk_i and gets \mathbf{C}_i . The ciphertexts $\{\mathbf{C}_i\}_{i \in [N]}$ can be used to evaluation without extra computation by Claim 1. We achieve this property by allowing two-round interaction between parties.

$\{\text{hk}_i\}_{i \in [k]} \leftarrow \text{KeyLifting}(\{\text{pk}_i, \text{sk}_i\}_{i \in [k]})$: Input the DGSW key pair $\{\text{pk}_i, \text{sk}_i\}_{i \in [k]}$ of all parties, where $\text{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1) \times n})$, $\mathbf{s}_i \leftarrow U\{0, 1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i \mathbf{A}_i \pmod q$. Assuming there is a broadcast channel, all parties are engaged in the following two interactions:

- First round : i broadcasts pk_i and receives $\{\text{pk}_j\}_{j \in [k] \setminus i}$ from the channel.
- Second round : i generates and broadcasts $\{\mathbf{b}_{i,j} = \mathbf{s}_i \mathbf{A}_j\}_{j \in [k] \setminus i}$, and receives $\{\mathbf{b}_{j,i} = \mathbf{s}_j \mathbf{A}_i\}_{j \in [k] \setminus i}$ from the channel.

After above two round interaction, i receives $\{\mathbf{b}_{j,i} = \mathbf{s}_j \mathbf{A}_i\}_{j \in [k] \setminus i}$. Let $\mathbf{b}_i = \sum_{j=1}^k \mathbf{b}_{j,i}$, i obtains hybrid key $\text{hk}_i = (\mathbf{A}_i, \mathbf{b}_i)$

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 :

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

it holds that(omit small error) :

$$\begin{aligned} \bar{\mathbf{t}} \mathbf{C}_i &\approx u_i \bar{\mathbf{t}} \mathbf{G}, & \bar{\mathbf{t}} \mathbf{C}_j &\approx u_j \bar{\mathbf{t}} \mathbf{G} \\ \bar{\mathbf{t}} (\mathbf{C}_i + \mathbf{C}_j) &\approx (u_i + u_j) \bar{\mathbf{t}} \mathbf{G}, & \bar{\mathbf{t}} \mathbf{C}_i \mathbf{G}^{-1} (\mathbf{C}_j) &\approx (u_i u_j) \bar{\mathbf{t}} \mathbf{G} \end{aligned}$$

Proof. According to the construction of $\text{KeyLifting}(\cdot)$, it holds that :

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

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

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

■

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.

Two hidden dangers for semi-malicious adversaries : There are two main security concerns about $\text{KeyLifting}(\cdot)$. First, a semi-malicious adversary may generate matrix \mathbf{A} with trapdoor, then \mathbf{s}_i is leaked. More specifically, our scheme leaks the key \mathbf{s}_i in two phases: in the $\text{KeyLifting}(\cdot)$ phase, $\{\mathbf{b}_{i,j} = \mathbf{s}_i \mathbf{A}_j\}_{j \in [k]}$ will lose \mathbf{s}_i at most $kn \log q$ bits, in the distributed decryption phase, since we do not introduce noise flooding, for a circuit with output length W , distributed decryption lose \mathbf{s}_i at most $W \log q$ bits, so the total leaked amount of \mathbf{s}_i is $(kn+W) \log q$ bits. According to the proof of Lemma 5, the length of \mathbf{s}_i must be at least $(kn+W) \log q + 2\lambda$ to ensure the indistinguishable of the ciphertext, which is why we set $m = (kn+W) \log q + 2\lambda$ in the scheme. 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 party i may not distributed as requirement. 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 phase to ensure security. Large encryption noise leads to large modulus q and high computational and communication overhead. In order to alleviate this problem, we proposed an analysis method based on Rényi divergence that neither introduces the above assumptions nor a large noise in the encryption process. For more details, please refer to Section 5.4.

5.2 The entire scheme

Our scheme is based on the DGSW scheme, containing the following five algorithms (Init, Gen, KeyLifting, Enc, Eval, Dec)

- $\text{pp} \leftarrow \text{Init}(1^\lambda, 1^L, 1^W)$: 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 $q = 2^{O(L)} B_\chi$, $k = \text{poly}(\lambda)$, $m = (kn+W) \log q + \lambda$, suitable choosing above parameters to make $\text{LWE}_{n,m,q,B_\chi}$ is infeasible. Output $\text{pp} = (k, n, m, q, \chi, B_\chi)$
- $(\text{pk}_i, \text{sk}_i) \leftarrow \text{Gen}(\text{pp})$: Input pp , output the DGSW key pair $(\text{pk}_i, \text{sk}_i)$ of parties i , where $\text{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i})$, $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{(m-1) \times n})$, $\mathbf{s}_i \leftarrow U\{0, 1\}^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i \mathbf{A}_i \pmod q$.
- $\text{hk}_i \leftarrow \text{KeyLifting}(\{\text{pk}_i, \text{sk}_i\}_{i \in [k]})$: All parties are engaged in the *Key lifting* procedure 5.1, output the hybrid key hk_i .
- $\mathbf{C}_i \leftarrow \text{Enc}(\text{hk}_i, u_i)$: 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.
- $\mathbf{C}^{(L)} \leftarrow \text{Eval}(S, \mathcal{C})$: Input the ciphertext set $S = \{\mathbf{C}_i\}_{i \in [N]}$ which are encrypted by hybrid key $\{\text{hk}_i\}_{i \in [k]}$, circuit \mathcal{C} with input length N , depth L , output $\mathbf{C}^{(L)}$.

Remark : In the security proof in Section 5.5, we require that χ be a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}, \sigma}$ over \mathbb{Z} with $\sigma > \sqrt{2} \eta_\epsilon(\mathbb{Z})$. and $ml > 4\lambda$.

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.

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

Distributed decryption Similar to [28], the decryption procedure is a distributed procedure :

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

5.3 Correctness analysis

To illustrate the correctness of our scheme, 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 :

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

According to the noise analysis of GSW in [19], the noise \mathbf{e}_L in $\mathbf{C}^{(L)}$ is bounded by $(ml)^L\mathbf{e}_{init}$. By the distributed decryption of our scheme, it holds that :

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

In order to decrypt correctly, it requires $\langle \mathbf{e}_L, \mathbf{G}^{-1}(\mathbf{w}^T) \rangle < \frac{q}{4}$. For our parameter settings :

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

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

5.4 Semantic Security of Encryption against Semi-Malicious Adversary

The concept of a semi-malicious adversary was proposed by Asharov et al. in [6], which is formalized as a polynomial capability Turing machine with an additional witness tape. It must explain the "legality" of the record on the output tape. Please refer to [6] for a more formal definition.

The semantic security of our scheme: For an 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 generate it adaptively. Under the semi-malicious adversary model, a common method to prove security is as follows: Assume that $\mathbf{b}_{i,j}$ satisfies the linear relationship $\mathbf{b}_{i,j} = \mathbf{s}_i \mathbf{A}_j$, and $\mathbf{s}_i \in \{0, 1\}^{m-1}$, and introduce large noise during encryption. In the following, we introduce this general method and then give an optimization proof method based on Rényi divergence.

A common approach : We complete the simulation by constructing a reduction from our scheme to the DGSW scheme. Consider the following Game:

1. Challenger generates $\mathbf{pk}_1 = (\mathbf{A}_1, \mathbf{b}_{1,1} = \mathbf{s}_1 \mathbf{A}_1)$ where $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{(m-1) \times n})$, $\mathbf{s}_1 \leftarrow U\{0, 1\}^{m-1}$ and send \mathbf{pk}_1 to adversary \mathcal{A}
2. After receiving \mathbf{pk}_1 , the adversary \mathcal{A} generates $\{\mathbf{pk}_i\}_{i \in [k]/1}$, where $\mathbf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i} = \mathbf{s}_i \mathbf{A}_i)$, and send it to Challenger.
3. After receiving $\{\mathbf{pk}_i\}_{i \in [k]/1}$, Challenger sets $\{\mathbf{b}_{1,i} = \mathbf{s}_1 \mathbf{A}_i\}_{i \in [k]/1}$ (the leakage of \mathbf{s}_1), and send it to \mathcal{A}
4. After receiving $\{\mathbf{b}_{1,i}\}_{i \in [k]/1}$, \mathcal{A} adaptively chooses $\{\mathbf{s}'_i\}_{i \in [k]/1}$, where $\mathbf{s}'_i \in \{0, 1\}^{m-1}$, set $\{\mathbf{b}_{i,1} = \mathbf{s}'_i \mathbf{A}_1\}_{i \in [k]/1}$, and send it to Challenger.
5. After receiving $\{\mathbf{b}_{i,1}\}_{i \in [k]/1}$, Challenger sets $\mathbf{hk}_1 = (\mathbf{A}_1, \sum_{i=1}^k \mathbf{b}_{i,1})$.
6. \mathcal{A} chooses a bit $u \leftarrow \{0, 1\}$, send it to Challenger.
7. Challenger chooses a bit $\alpha \leftarrow \{0, 1\}$, if $\alpha = 0$ sets $\mathbf{C} \leftarrow \text{Enc}(\mathbf{hk}_1, u)$, otherwise $\mathbf{C} \leftarrow U(\mathbb{Z}_q^{m \times ml})$, send \mathbf{C} to \mathcal{A} .
8. After receiving \mathbf{C} , \mathcal{A} output bit $\bar{\alpha}$, if $\bar{\alpha} = \alpha$, then \mathcal{A} wins.

Claim 2 Let $\text{Adv} = |Pr[\bar{\alpha} = \alpha] - \frac{1}{2}|$ 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 of DGSW and the uniform distribution with the same (up to negligible) advantage.

Proof. After the third step of the above game, \mathcal{A} obtained pk_1 and $\{\mathbf{b}_{1,i}\}_{i \in [k]/1}$ (the leakage of \mathbf{s}_1). Next, we use the ciphertext of DGSW to construct \mathbf{C} . Consider the following sequence :

1. Challenger chooses a bit $\alpha \leftarrow \{0, 1\}$, if $\alpha = 0$ sets $\mathbf{C}_{\text{DGSW}} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}$, otherwise $\mathbf{C}_{\text{DGSW}} \leftarrow U(\mathbb{Z}_q^{m \times ml})$, send it to \mathcal{A} .
2. After receiving \mathbf{C}_{DGSW} , \mathcal{A} adaptively chooses $\{\mathbf{s}'_i\}_{i \in [k]/1}$, a bit $u \leftarrow \{0, 1\}$, send it to Challenger.
3. After receiving $\{\mathbf{s}'_i\}_{i \in [k]/1}$ and u , let $\mathbf{C}_{\text{DGSW}} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}$, $\mathbf{s}' = \sum_{i=2}^k \mathbf{s}'_i$,

$$\mathbf{C}' = \mathbf{C}_{\text{DGSW}} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix} + u\mathbf{G}$$

Obviously, if $\alpha = 1$, \mathbf{C}' is uniform, otherwise it holds that :

$$\begin{aligned} \mathbf{s}'\mathbf{C}_0 &= \mathbf{s}'(\mathbf{A}_1\mathbf{R} + \mathbf{E}_0) = \sum_{i=2}^k \mathbf{b}_{i,1}\mathbf{R} + \mathbf{s}'\mathbf{E}_0 \\ \mathbf{C}' &= \mathbf{C}_{\text{DGSW}} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix} + u\mathbf{G} \\ &= \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix} + u\mathbf{G} \\ &= \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0 \end{pmatrix} + u\mathbf{G} \end{aligned}$$

If $\|\mathbf{e}_1\|_\infty$ is bounded by $2^\lambda B_\chi$, and $\|\mathbf{s}'\mathbf{E}_0\|_\infty < kmB_\chi$, thus $\mathbf{s}'\mathbf{E}_0/\mathbf{e}_1 = \text{negl}(\lambda)$. By Lemma 1, it holds that $\mathbf{C}' \stackrel{\text{stat}}{\approx} \mathbf{C}$, if \mathcal{A} can distinguish between \mathbf{C} and uniform distribution by advantage Adv , then he can distinguish between \mathbf{C}_{DGSW} and the uniform distribution with the same advantage. We note that the above sequence handles the leakage of \mathbf{s}_1 , for \mathbf{C}_{DGSW} is a ciphertext generated by pk_1 , which security is guaranteed by Lemma 5. \blacksquare

Remark: When $\|\mathbf{e}_1\|_\infty$ is bounded by $2^\lambda B_\chi$, according to the correctness analysis in Section 5.3, the initial noise $\mathbf{e}_{\text{init}} = \mathbf{e}_1 - \mathbf{s}\mathbf{E}_0$ 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 : The work of Shi et al. [7] pointed out that Rényi divergence can also be applied in distinguish problems, and in some cases, it can lead to better parameters than statistical distance. Based on these results, they obtained better parameters of the Regev encryption scheme. Theorem 2 states: if there is an algorithm that can distinguish the P problem, then there is an algorithm that can distinguish the P' problem. Our proof method is as follows :

- Define the P problem as distinguishing our ciphertext from a uniform distribution
- Prove that for a given DGSW ciphertext, there exists a distribution X'_0 , and a sample x of X'_0 can be constructed from this DGSW ciphertext,
- Define the P' problem as distinguishing X'_0 from a uniform distribution

Thus, if there is an adversary who can distinguish the P problem, then he can distinguish the P' problem and can also distinguish the DGSW ciphertext from the uniform distribution.

Claim 3 If there is an adversary who can distinguish the ciphertext of our scheme from the uniform distribution, then the adversary can distinguish the DGSW ciphertext from the uniform distribution.

Proof. Let $\mathbf{0}^{1 \times ml}$ be a zero vector of length ml , Φ be the distribution of hybrid key of Challenger followed by $\mathbf{0}^{1 \times ml}$:

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

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

$$(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml}, \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}) \leftarrow \mathcal{D}_0(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml})$$

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

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

Let P be the decision problems defined as follows :

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

$$X_0 = \{x : (\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml}, \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}) \leftarrow \mathcal{D}_0(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml})\}.$$

$$X_1 = \{x : (\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml}, \mathbf{U}) \leftarrow \mathcal{D}_1(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml})\}.$$

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

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

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

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

where $\mathbf{E}_{0,i} \in \mathbb{Z}_q^{(m-1) \times n}$, $\mathbf{e}_{1,i} \in \mathbb{Z}_q^n$. Then, \mathbf{C}' can be expressed as :

$$\mathbf{C}' = \begin{pmatrix} \mathbf{A}_1 \mathbf{R} + \mathbf{E}_0 \\ \mathbf{b}_1 \mathbf{R}_1 + \mathbf{s}' \mathbf{E}_{0,1} + \mathbf{e}_{1,1}, \mathbf{b}_1 \mathbf{R}_2 + \mathbf{s}' \mathbf{E}_{0,2} + \mathbf{e}_{1,2}, \dots, \mathbf{b}_1 \mathbf{R}_g + \mathbf{s}' \mathbf{E}_{0,g} + \mathbf{e}_{1,g} \end{pmatrix}$$

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 with an overwhelming probability (See Appendix A). Define set V :

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

Define the distribution $\mathcal{D}(V)$ over set V :

$$\mathbf{d} \leftarrow \mathcal{D}(V) : \begin{cases} \Pr(\mathbf{d} = \mathbf{0}^{1 \times ml}) = p \\ \Pr(\mathbf{d} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_g)) = 1 - p \end{cases}$$

Let Φ' be the joint distribution of the hybrid key of our scheme and $\mathcal{D}(V)$:

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

Let P' be the decision problems defined as follows :

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

$$\begin{aligned} X'_0 &= \{x : (\mathbf{A}', \mathbf{b}', \mathbf{d}) \leftarrow \Phi', \\ x &= \left(\mathbf{A}', \mathbf{b}', \mathbf{d}, \left((\mathbf{b}' + \mathbf{d}_1)\mathbf{R}'_1 + \mathbf{e}'_1, (\mathbf{b}' + \mathbf{d}_2)\mathbf{R}'_2 + \mathbf{e}'_2, \dots, (\mathbf{b}' + \mathbf{d}_g)\mathbf{R}'_g + \mathbf{e}'_g \right) \right) \leftarrow \mathcal{D}_0(\mathbf{A}', \mathbf{b}', \mathbf{d})\} \\ X'_1 &= \{x : (\mathbf{A}', \mathbf{b}', \mathbf{d}) \leftarrow \Phi', \quad x = (\mathbf{A}', \mathbf{b}', \mathbf{d}, \mathbf{U}) \leftarrow \mathcal{D}_1(\mathbf{A}', \mathbf{b}', \mathbf{d})\}. \end{aligned}$$

where $\mathbf{R}' = (\mathbf{R}'_1, \mathbf{R}'_2, \dots, \mathbf{R}'_g)$, $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_g)$

Thus for any \mathbf{C}' :

$$\mathbf{C}' = \left(\begin{array}{c} \mathbf{A}_1 \mathbf{R} + \mathbf{E}_0 \\ \mathbf{b}_1 \mathbf{R}_1 + \mathbf{s}' \mathbf{E}_{0,1} + \mathbf{e}_{1,1}, \mathbf{b}_1 \mathbf{R}_2 + \mathbf{s}' \mathbf{E}_{0,2} + \mathbf{e}_{1,2}, \dots, \mathbf{b}_1 \mathbf{R}_g + \mathbf{s}' \mathbf{E}_{0,g} + \mathbf{e}_{1,g} \end{array} \right)$$

it is a sample :

$$x = (\mathbf{A}', \mathbf{b}', \mathbf{d}, \left((\mathbf{b}' + \mathbf{d}_1)\mathbf{R}'_1 + \mathbf{e}'_1, (\mathbf{b}' + \mathbf{d}_2)\mathbf{R}'_2 + \mathbf{e}'_2, \dots, (\mathbf{b}' + \mathbf{d}_g)\mathbf{R}'_g + \mathbf{e}'_g \right))$$

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

We note that \mathbf{C}' only forms part of the sample of X'_0 . The completed sample also contains $\{\mathbf{v}_i\}_{i \in [g]}$ which 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 \mathcal{A} . Consider the following sequence :

1. Challenger generates DGSW ciphertext $\mathbf{C}_{\text{DGSW}} = \left(\begin{array}{c} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{array} \right) \mathbf{R} + \left(\begin{array}{c} \mathbf{E}_0 \\ \mathbf{e}_1 \end{array} \right)$ and send it to adversary
2. After receiving \mathbf{C}_{DGSW} , \mathcal{A} adaptively generates \mathbf{s}' , $\|\mathbf{s}'\|_\infty \leq \sqrt{n \log q}$, and send it to Challenger.
3. Challenger computes $\{\mathbf{v}_i\}_{i \in [g]}$ by $\{\mathbf{v}_i \mathbf{R}_i = \mathbf{s}' \mathbf{E}_{0,i}\}_{i \in [g]}$, and then 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 \mathcal{A} gets $\{\mathbf{v}_i\}_{i \in [g]}$, \mathbf{C}_{DGSW} is still indistinguishable. By the leftover hash lemma, we can replace $\mathbf{b}_{1,1}$ by $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$. Thus, distinguish $\left(\left(\begin{array}{c} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{array} \right), \mathbf{C}_{\text{DGSW}}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s}' \right)$ with uniform corresponding to distinguish the "Interactive LWE" problem defined in Section 4:

$$(\mathbf{A}, \mathbf{A}\mathbf{R} + \mathbf{E}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s}') \quad \text{and} \quad (\mathbf{A}, \mathbf{U}, \{\mathbf{v}_i\}_{i \in [g]}, \mathbf{s}')$$

So far, we have completed the construction of X'_0 samples: that is, for each given DGSW ciphertext \mathbf{C}_{DGSW} , after getting \mathbf{s}' from \mathcal{A} , Challenger can convert it into a sample of X'_0 . Since the outputs of our distributions of $\mathcal{D}_0(\cdot)$ and $\mathcal{D}_1(\cdot)$ contain the samples $(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml})$ or $(\mathbf{A}', \mathbf{b}', \mathbf{d})$ of the prior distributions Φ and Φ' , thus $\mathcal{D}_0(\cdot)$ and $\mathcal{D}_1(\cdot)$ satisfy the publicly sampleable property (see Theorem 2) required by Theorem 2. The sampling algorithm S is just the encryption operation of our scheme with hybrid key $(\mathbf{A}_1, \mathbf{b}_1, \mathbf{0}^{1 \times ml})$ or $(\mathbf{A}', \mathbf{b}', \mathbf{d})$. Then, by Theorem 2, 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 $a \in (1, +\infty]$) :

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

■

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

$$\begin{aligned} R_\infty(\Phi|\Phi') &= \max_{Y \in \text{Supp}(\Phi)} \frac{\Phi(Y)}{\Phi'(Y)} = \frac{\Phi(\mathbf{A}_0, \mathbf{b}_0, \mathbf{0}^{1 \times ml})}{\Phi'(\mathbf{A}_0, \mathbf{b}_0, \mathbf{0}^{1 \times ml})} \\ &= \frac{\Pr(\mathbf{A} = \mathbf{A}_0, \mathbf{b} = \mathbf{b}_0)}{\Pr(\mathbf{A} = \mathbf{A}_0, \mathbf{b} = \mathbf{b}_0, \mathbf{d} = \mathbf{0}^{1 \times ml})} \end{aligned} \quad (2)$$

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

$$(2) = \frac{\Pr(\mathbf{A} = \mathbf{A}_0, \mathbf{b} = \mathbf{b}_0)}{\Pr(\mathbf{A} = \mathbf{A}_0, \mathbf{b} = \mathbf{b}_0) \Pr(\mathbf{d} = \mathbf{0}^{1 \times ml})} = \frac{1}{p}$$

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 :

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

Remark : Under the semi-honest adversary model, $\{\mathbf{A}_i\}_{i \in [k]}$ and $\{\mathbf{s}_i\}_{i \in [k]}$ are sampled as specified by the protocol, 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. However, in our proof method based on the Rényi divergence, it is necessary to assume that "Interactive LWE " is hard.

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

5.5 Noise flooding technology VS Leakage resilient property in partial decryption

We note that introducing noise flooding in the partial decryption phase is essential 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 [28], masking techniques based on noise flooding can only guarantee weak simulatable properties: input all private keys $\{\mathbf{sk}_j\}_{j \in [k]/i}$ except \mathbf{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 $\{\mathbf{sk}_j\}_{j \in S}$ 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 do not know how to achieve it.

With noise flooding : To illustrate how our approach works, let us first review the noise flooding technique. 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 $\text{LocalDec}(\cdot)$, we have :

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

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

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

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

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

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

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

For the parameter settings in [28] : $B_{smdg} = 2^{L\lambda \log \lambda} B_\chi$, $q = 2^{\omega(L\lambda \log \lambda)} B_\chi$, obviously :

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

$$\text{thus } \gamma_i \stackrel{\text{stat}}{\approx} \gamma_i'.$$

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 party i contained in \mathbf{e}_L (Noise obtained by 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 party i can be simulated, providing other parties with 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 us look at how \mathbf{e}_L is generated. For the initial ciphertext :

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

After performing a homomorphic multiplication operation, we obtain:

$$\begin{aligned} \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2) &= \left[\begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1 + \mathbf{E}_1 + u_1 \mathbf{G} \right] \mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_1 \end{pmatrix} \mathbf{R}_1 \mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{E}_1 \mathbf{G}^{-1}(\mathbf{C}_2) + u_1 \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{b}_2 \end{pmatrix} \mathbf{R}_2 + u_1 \mathbf{E}_2 + u_1 u_2 \mathbf{G} \\ &= \Pi_1 + \delta_1 + u_1 u_2 \mathbf{G} \end{aligned}$$

where :

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

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 $\mathbf{C}_3 \mathbf{G}^{-1}(\mathbf{C}_4) = \Pi'_1 + \delta'_1 + u_3 u_4 \mathbf{G}$, 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 :

$$\begin{aligned} \mathbf{C}^{(2)} &= \mathbf{C}_1 \mathbf{G}^{-1}(\mathbf{C}_2), \quad \mathbf{C}^{(2)'} = \mathbf{C}_3 \mathbf{G}^{-1}(\mathbf{C}_4) \\ \delta_2 &= \delta_1 \mathbf{G}^{-1}(\mathbf{C}^{(2)'}) + u_1 u_2 \delta'_1 \end{aligned}$$

we have $\mathbf{C}^{(2)} \mathbf{G}^{-1}(\mathbf{C}^{(2)'}) = \Pi_2 + \delta_2 + u_1 u_2 u_3 u_4 \mathbf{G}$. For the ciphertext at level L , we have :

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

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 :

$$\delta_1 \stackrel{\text{stat}}{\approx} \mathbf{E}_1 \mathbf{G}^{-1}(\mathbf{C}_2) \quad (3)$$

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 [30] has already proved the case of two discrete Gaussian summations, while we 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 :

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

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

$$\begin{aligned} \Delta(y_1 + y_2, y_1^{(1)}) &< 8\epsilon \\ \Delta(y_3 + y_4, y_2^{(1)}) &< 8\epsilon \\ &\dots \\ \Delta(y_{ml-1} + y_{ml}, y_{\frac{ml}{2}}^{(1)}) &< 8\epsilon \end{aligned}$$

By the subadditivity of statistical distances (we proved it in Appendix B), we have :

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

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:

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

Iterating the above process, we have :

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

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, which already satisfies our needs. For the case of summing multiple discrete Gaussian, if one follows the path of [30], 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 :

$$\begin{aligned} \delta_1(1, 1) &= z_1 e_1 + z_2 e_2 + \dots + z_{ml} e_{ml} + e_{ml+1} \\ \mathbf{E}_1 \mathbf{M}(1, 1) &= z_1 e_1 + z_2 e_2 + \dots + z_{ml} e_{ml} \end{aligned}$$

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 :

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

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

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

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

$$\begin{aligned} \Delta(\delta_1(1, 1), \mathbf{E}_1 \mathbf{M}(1, 1)) &= \frac{1}{2} \sum_{-\infty}^{+\infty} \left| \frac{\rho_{\sqrt{r}\sigma}(x)}{\rho_{\sqrt{r}\sigma}(\mathbb{Z})} - \frac{\rho_{\sqrt{r+1}\sigma}(x)}{\rho_{\sqrt{r+1}\sigma}(\mathbb{Z})} \right| = \sum_{-x}^x \left(\frac{\rho_{\sqrt{r}\sigma}(x)}{\rho_{\sqrt{r}\sigma}(\mathbb{Z})} - \frac{\rho_{\sqrt{r+1}\sigma}(x)}{\rho_{\sqrt{r+1}\sigma}(\mathbb{Z})} \right) \\ &= 2 \sum_{-\infty}^{-x} \left(\frac{\rho_{\sqrt{r+1}\sigma}(x)}{\rho_{\sqrt{r+1}\sigma}(\mathbb{Z})} - \frac{\rho_{\sqrt{r}\sigma}(x)}{\rho_{\sqrt{r}\sigma}(\mathbb{Z})} \right) < 2 \sum_{-\infty}^{-x} \frac{\rho_{\sqrt{r+1}\sigma}(x)}{\rho_{\sqrt{r+1}\sigma}(\mathbb{Z})}. \end{aligned}$$

where $x = \sqrt{r(r+1)} \ln \frac{r+1}{r} \sigma$ is the root of equation :

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

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

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

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 1,

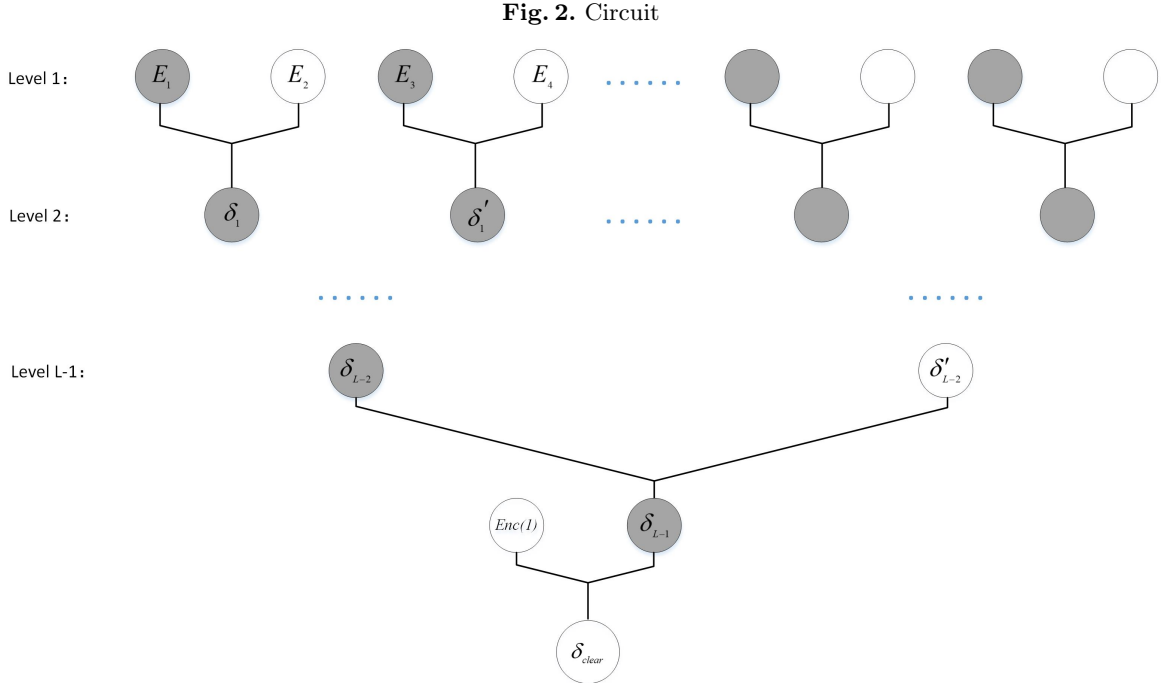
$$\Pr(r < \lambda) \leq e^{-\frac{(\frac{1}{2}ml - \lambda)^2}{ml - \lambda}} = \text{negl}(\lambda)$$

for $ml > 4\lambda$. Thus, the statistical distance of $\delta_1(1, 1)$ and $\mathbf{E}_1\mathbf{M}(1, 1)$:

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

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

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 :

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

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

Here, the reader might think that doing so would result in a longer key than noise flooding. We point out that as long as the output length W of the 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.

5.6 Bootstrapping

In order to eliminate the dependence on the circuit depth to achieve full homomorphism, we need to use Gentry's bootstrapping technology. It is worth noting that the bootstrapping procedure of our scheme is the same as the single-key homomorphic scheme: After *Key lifting* procedure, party 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 $\bar{\mathbf{t}} = (-\sum_{i=1}^k \mathbf{s}_i, 1)$, homomorphic evaluation of the decryption circuit could be executed directly as $\mathbf{C}^{(L)}$ are need to be refresh. Therefore, 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 require ciphertext expansion, additional ciphertext expansion is required, for the reason that $\mathbf{C}^{(L)}$ is the ciphertext under $\bar{\mathbf{t}}$, but $\{\text{evk}_i\}_{i \in [k]}$ are the ciphertext under $\{\mathbf{t}_i\}_{i \in [k]}$. In order to expand $\{\text{evk}_i\}_{i \in [k]} \rightarrow \{\widehat{\text{evk}_i}\}_{i \in [k]}$, party i needs to encrypt the random matrix of the ciphertext corresponding to evk_i . The extra encryption of i needs to be done locally are $O(\lambda^9 L^6)$.

6 Conclusions

For the LWE-based MKFHE, in order to alleviate the overhead of the local parties, we proposed the concept of KL-MKFHE, which introduced a *Key lifting* procedure, getting rid of expensive ciphertext expansion operation and constructing a DGSW style KL-MKFHE under the plain model. Our scheme is more friendly to local parties than the previous scheme, for which the local encryption $O(N\lambda^6 L^4)$ are reduced to $O(N)$. By abandoning noise flooding, it compresses 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 parties and the amount of leakage, which limits the scheme's application to some extent. Further work will focus on compressing the key length.

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., Asharov, G., Dahari, H., Goyal, V.: Towards accountability in CRS generation. pp. 278–308. LNCS, Springer, Heidelberg (2021)
4. 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)
5. Ananth, P., Jain, A., Jin, Z., Malavolta, G.: Unbounded multi-party computation from learning with errors. pp. 754–781. LNCS, Springer, Heidelberg (2021)
6. 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)

7. 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)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) *ITCS 2012*. pp. 309–325. ACM (Jan 2012)
9. 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)
10. 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)
11. 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)
12. 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)
13. 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)
14. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Genaro, R., Robshaw, M.J.B. (eds.) *CRYPTO 2015, Part II. LNCS*, vol. 9216, pp. 630–656. Springer, Heidelberg (Aug 2015)
15. 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)
16. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive, Report 2012/144* (2012), <https://eprint.iacr.org/2012/144>
17. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) *41st ACM STOC*. pp. 169–178. ACM Press (May / Jun 2009)
19. 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)
20. 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)
21. 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)
22. Lovász, L., Pelikán, J., Vesztergombi, K.: *Discrete mathematics: elementary and beyond*. Springer Science & Business Media (2003)
23. 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)
24. 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)
25. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing* 37(1), 267–302 (2007)
26. 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>
27. 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)
28. 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)
29. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher, M. (ed.) *41st ACM STOC*. pp. 333–342. ACM Press (May / Jun 2009)
30. 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)
31. Peikert, C., Shiehian, S.: Multi-key fhe from lwe, revisited. In: *Theory of Cryptography Conference*. pp. 217–238. Springer (2016)
32. Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. *Cryptology ePrint Archive, Report 2016/196* (2016), <https://eprint.iacr.org/2016/196>
33. 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)

Proof.

$$\begin{aligned}
\Delta(X + Y, Y + Z) &= \frac{1}{2} \sum_{k \in E} |\Pr(X + Y = k) - \Pr(Z + Y = k)| \\
&= \frac{1}{2} \sum_{k \in E} |\Pr(X = k - Y) - \Pr(Z = k - Y)| \\
&= \frac{1}{2} \sum_{k \in E} \left| \sum_{b \in E} (\Pr(Y = b) \Pr(X = k - b) - \Pr(Y = b) \Pr(Z = k - b)) \right| \\
&= \frac{1}{2} \sum_{k \in E} \left| \sum_{b \in E} \Pr(Y = b) (\Pr(X = k - b) - \Pr(Z = k - b)) \right| \\
&\leq \frac{1}{2} \sum_{k \in E} \sum_{b \in E} |\Pr(Y = b) (\Pr(X = k - b) - \Pr(Z = k - b))| \\
&= \frac{1}{2} \sum_{b \in E} \Pr(Y = b) \sum_{k \in E} |\Pr(X = k - b) - \Pr(Z = k - b)| \\
&\leq \sum_{b \in E} \Pr(Y = b) \cdot \Delta(X, Z) \\
&= \Delta(X, Z)
\end{aligned}$$

■

Claim 6 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 4, We have :

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

then, by Claim 5, We have :

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

■

C The proof of DGSW leakage-resilient in [9], and our improved method.

For a given DGSW ciphertext :

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}$$

Let $\mathbf{C}_0 = \mathbf{A}\mathbf{R} + \mathbf{E}_0$, $\mathbf{c}_1 = \mathbf{b}\mathbf{R} + \mathbf{e}_1$. Because $\mathbf{b} = \mathbf{s}\mathbf{A}$, thus \mathbf{C} can be rewritten as :

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{s}\mathbf{C}_0 + \mathbf{e}_1 - \mathbf{s}\mathbf{E}_0 \end{pmatrix} \quad (4)$$

The proof in [9] required $\mathbf{s}\mathbf{E}_0/\mathbf{e}_1 = \text{negl}(\lambda)$, thus $\mathbf{C} \stackrel{\text{stat}}{\approx} \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{s}\mathbf{C}_0 + \mathbf{e}_1 \end{pmatrix}$. Using the leftover hash lemma with \mathbf{C}_0 as a seed and \mathbf{s} as a source, they had that $(\mathbf{C}_0, \mathbf{s}\mathbf{C}_0)$ were jointly statistically indistinguishable from uniform, which Lemma 5 followed.

Our method : Below we show that $\mathbf{s}\mathbf{E}_0/\mathbf{e}_1 = \text{negl}(\lambda)$ is not necessary to prove that DGSW is leakage-resilient. Through the above analysis, we know that for any DGSW ciphertext, we can always write it in the form of (4). 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 :

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

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

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

where $\mathbf{E}_{0,i} \in \mathbb{Z}_q^{(m-1) \times n}$, $\mathbf{e}_{1,i} \in \mathbb{Z}_q^n$. Let $\{\mathbf{v}_i \in \mathbb{Z}_q^n\}_{i \in [g]}$ be the solution of equation :

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

Obviously, if \mathbf{R}_i is random over $\mathbb{Z}_q^{n \times n}$, then \mathbf{v}_i has a unique solution with an overwhelming probability. Let $\mathbf{0}^{1 \times ml}$ be a zero vector of length ml , Φ be the distribution of public key of DGSW followed by $\mathbf{0}^{1 \times ml}$:

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

Let $\mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml})$ be the joint distribution of public key and ciphertext of DGSW, over the randomness \mathbf{R} , \mathbf{E}_0 , \mathbf{e}_1 :

$$(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml}, \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}) \leftarrow \mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml})$$

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 $\mathbf{U} \leftarrow U(\mathbb{Z}_q^{m \times ml})$:

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

Let P be the decision problems defined as follows :

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

$$\begin{aligned}X_0 &= \{x : (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml}, \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}) \leftarrow \mathcal{D}_0(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml})\}. \\ X_1 &= \{x : (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml}) \leftarrow \Phi, \quad x = (\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml}, \mathbf{U}) \leftarrow \mathcal{D}_1(\mathbf{A}, \mathbf{b}, \mathbf{0}^{1 \times ml})\}.\end{aligned}$$

Define set V :

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

Define the distribution $\mathbf{d} \leftarrow \mathcal{D}(V)$ over set V :

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

Let Φ' be the joint distribution of DGSW public key and $\mathcal{D}(V)$:

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

Let P' be the decision problems defined as follows :

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

$$\begin{aligned}X'_0 &= \{x : (\mathbf{A}', \mathbf{b}', \mathbf{d}) \leftarrow \Phi', \\ x &= (\mathbf{A}', \mathbf{b}', \mathbf{d}, \begin{pmatrix} \mathbf{A}'\mathbf{R}' + \mathbf{E}'_0 \\ (\mathbf{b}' + \mathbf{d}_1)\mathbf{R}'_1 + \mathbf{e}'_1, \dots, (\mathbf{b}' + \mathbf{d}_g)\mathbf{R}'_g + \mathbf{e}'_g \end{pmatrix}) \leftarrow \mathcal{D}_0(\mathbf{A}', \mathbf{b}', \mathbf{d})\}. \\ X'_1 &= \{x : (\mathbf{A}', \mathbf{b}', \mathbf{d}) \leftarrow \Phi', \quad x = (\mathbf{A}', \mathbf{b}', \mathbf{d}, \mathbf{U}) \leftarrow \mathcal{D}_1(\mathbf{A}', \mathbf{b}', \mathbf{d})\}.\end{aligned}$$

where $\mathbf{R}' = (\mathbf{R}'_1, \dots, \mathbf{R}'_g) \leftarrow U(\mathbb{Z}_q^{n \times ml})$, $\mathbf{e}'_i \leftarrow \chi^n$, $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_g)$. Thus, for \mathbf{C}' :

$$\mathbf{C}' = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{sC}_0 + \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{AR} + \mathbf{E}_0 \\ \mathbf{bR} + \mathbf{e}_1 + \mathbf{sE}_0 \end{pmatrix} = \begin{pmatrix} \mathbf{AR} + \mathbf{E}_0 \\ \mathbf{bR}_1 + \mathbf{e}_{1,1} + \mathbf{sE}_{0,1}, \dots, \mathbf{bR}_g + \mathbf{e}_{1,g} + \mathbf{sE}_{0,g} \end{pmatrix}$$

it is a sample of X'_0 :

$$\begin{pmatrix} \mathbf{A}'\mathbf{R}' + \mathbf{E}'_0 \\ (\mathbf{b}' + \mathbf{d}_1)\mathbf{R}'_1 + \mathbf{e}'_1, \dots, (\mathbf{b}' + \mathbf{d}_g)\mathbf{R}'_g + \mathbf{e}'_g \end{pmatrix}$$

with $\mathbf{A}' = \mathbf{A}$, $\mathbf{b}' = \mathbf{b}$, $\mathbf{R}' = \mathbf{R}$, $\mathbf{E}'_0 = \mathbf{E}_0$, $(\mathbf{e}'_1, \mathbf{e}'_2, \dots, \mathbf{e}'_g) = (\mathbf{e}_{1,1}, \mathbf{e}_{1,2}, \dots, \mathbf{e}_{1,g})$, $\mathbf{d} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_g)$.

The following process is the same as we showed in Section 5.4. By Theorem 2, if there is an adversary who can distinguish the DGSW ciphertext with uniform distribution (Problem P) that leaks part of the private key, then he can distinguish $(\mathbf{C}_0, \mathbf{sC}_0 + \mathbf{e}_1)$ with uniform distribution which is jointly statistically indistinguishable by leftover hash lemma.