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

Xiaokang Dai\textsuperscript{1,2} Wenyuan Wu\textsuperscript{2,2} and Yong Feng\textsuperscript{2}

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

Abstract. Multi-key Fully Homomorphic Encryption (MKFHE) based on Learning With Error (LWE) usually lifts ciphertexts of different users to new ciphertexts under a common public key to enable homomorphic evaluation. The main obstacle of current MKFHE schemes in applications is huge ciphertext expansion cost especially in data intensive scenario. For example, a boolean circuit with input length $N$, multiplication depth $L$, security parameter $\lambda$, the number of additional encryptions introduced to achieve ciphertext expansion is $O(N^6L^4)$.

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

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

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

1 Introduction

Fully Homomorphic Encryption (FHE). The concept of FHE was proposed by Rivest et al. [41], within a year of publishing of the RSA scheme [42]. The
first truly fully homomorphic scheme was proposed by Gentry in his doctoral dissertation [21] in 2009. Based on Gentry's ideas, a series of FHE schemes have been proposed [22] [44] [10] [20] [23] [15] [14], and their security and efficiency have been continuously improved. FHE is suitable to the problem of unilateral outsourcing computations. However in the case of multiple data providers, in order to support homomorphic evaluation, data must be encrypted by a common public key. Due to privacy of data, it is unreasonable to require participants to use other people’s public keys to encrypt their own data.

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

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

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

**Multi-key Fully Homomorphic Encryption (MKFHE).** To deal with privacy of multiple data providers, López-Alt et al [26] proposed the concept of MKFHE and constructed the first MKFHE scheme based on modified-NTRU [43]. Conceptually, it was an enhancement of the FHE on functionality that allowed data provider to encrypt data independent from other participants, its key generation and data encryption were done locally. To get the evaluated result, all participants were required to execute a round of threshold decryption protocol.

After López-Alt et al. proposed the concept of MKFHE, many schemes were proposed. In 2015, Clear and McGoldrick [16] constructed a LWE-based MKFHE. This scheme defined the common private key as the concatenation of all private keys, and constructed a masking scheme to converts the ciphertext under individual public key to common public key by introducing CRS and circular-LWE assumptions, which only supports single-hop computation. In 2016, Mukherjee and Wich [35], Perkert and shiehian [38], Brakerski and Perlman [12] constructed MKFHE scheme based on GSW respectively. Mukherjee and Wich [35] simplified the mask scheme of [16], and focused on constructing a two-round MPC protocol. Different methods in [38] and [12] were put forward delicately to constructing a multi-hop MKFHE. Brakerski and Perlman [12] introduced bootstrapping to realize ciphertext expansion, thereby realizing the multi-hop function. Perkert and shiehian [38] realized multi-hop function through ingenious construction. It is worth mentioning that all MKFHE schemes constructed based on the LWE requires a ciphertext expansion procedure.

### 1.1 Motivation

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

**Ciphertext expansion is expensive:** Although the MKFHE based on LWE can use LHL to remove CRS. In order to convert the ciphertext under different keys to the ciphertext under a same key (ciphertext expansion procedure), participants and the computing server need to do a lot of preparatory work. For ciphertext expansion, it is necessary to encrypt the random matrix $R \in \mathbb{Z}_q^{m \times m}$ of each ciphertext. For a boolean circuit with input length $N$, multiplication
depth $L$, security parameter $\lambda$, $m = n \log q + \omega(\log \lambda)$, the additional encryption operation introduced is $O(N\lambda^6L^4)$, in contrast to $O(N)$ for single-key FHE. For computing-sensitive participants, this is a lot of overhead.

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

Leftover Hash Lemma (LHL) over integer ring $\mathbb{Z}$ enjoys the leakage resilient property: It can transform an average quality random sources into higher quality [24] which can be used to get rid of CRS as [11] does. However, regularity lemma [29] over polynomial rings do not have corresponding properties, as [17] mentioned if the $j$-th Number theoretical Transfer (NTT) coordinate of each ring element in $x = (x_1, \ldots, x_l)$ is leaked, then the $j$-th NTT coordinate of $a_{l+1} = \sum a_i x_i$ is defined, so $a_{l+1}$ is very far from uniform, yet this is only a $1/n$ leakage rate. Therefore, it seems to be more difficult to remove CRS for RLWE-based MKFHE.

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Key structure</th>
<th>CRS</th>
<th>Noise flooding</th>
<th>Interaction(setup phase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>THFHE [5]</td>
<td>Sum</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MKFHE [13]</td>
<td>Concatenation</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>MKFHE [35]</td>
<td>Concatenation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scheme#1</td>
<td>Sum</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Scheme#2</td>
<td>Sum</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

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

1.3 Our Results

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

Since regularity lemma [30] on rings has no corresponding leakage resilient properties, we cannot apply the LWE construction routine trivially to RLWE-based MKFHE. As a compromise, we introduce a round of bit commitment protocol to guarantee the independence of each participants, and construct the corresponding KL-MKFHE based on ROM.

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

The concept of KL-MKFHE : Different from previous definition [35], we abandon ciphertext expansion procedure, instead, introducing a key lifting procedure which at a lower cost. In addition to the properties that required by
MKFHE, such as Correctness, Compactness, semantic security, Simulatability of decryption, KL-MKFHE should satisfy the following two additional properties:

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

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

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

We believe that this Rényi divergence-based proof method provides an alternative idea for those proofs that need to introduce strong assumptions and large noise to ensure security. For example, we give the optimal proof method for the leakage-resilient of the DGSW scheme in Appendix B(without introducing large noise). More details please refer to Section 4.4.

**Leakage resistance implies a smaller q**: We notice that, in the distributed decryption phase, introducing large noise to cover up the information of the private key is essentially to ensure the security of the plaintext. But adding noise is just one way to achieve it. In particular, we observe that if the encryption scheme is leakage resistant, so the same purpose can be achieved alternatively by just increasing the significant bits of private key appropriately.
Fig. 1. The procedures of MKFHE and KL-MKFHE
Assuming that the output length of the circuit to be evaluated is \( W \), without noise flooding, the information of private key leaked in the partial decryption results is \( W \log q \) bits. We only need to increase the length of \( s \) by an additional \( W \log q \) bits to ensure the semantic security of the ciphertext. Since there is no noise flooding in encryption and distributed decryption, we can set \( q = 2^{O(L)}B_x \) to be the same size as the single-key homomorphic scheme, where \( q = 2^{O(\lambda L)}B_x \) in [5] [35] with noise flooding technology. Refer to Section 4.5 for a detailed discussion.

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

The security of Scheme\#1 is based on the LWE assumption. The common private key is the sum of the private keys of all participants. We note that previous MKFHE or Th-FHE schemes [33] [5] adopt this key structure are all based on the CRS model. Without CRS, our solution is simpler and more efficient in construction. For a circuit with an input length \( N \), our scheme requires local users to perform \( O(N) \) encryption operations, while it is \( O(N^{6}L^{4}) \) for those schemes that require ciphertext expansion.

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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>PubKey + EvalKey</th>
<th>Ciphertext</th>
<th>Module q</th>
<th>Extra encryption</th>
<th>Interaction(setup phase)</th>
<th>CRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKFHE [38]</td>
<td>( \tilde{O}(\lambda^{6}L^{4}(k + N \lambda^{3}L^{2})) )</td>
<td>( \tilde{O}(Nk^{2}\lambda^{6}L^{4}) )</td>
<td>( 2^{O(\lambda L)}B_x )</td>
<td>( \tilde{O}(N^{3}L^{4}) )</td>
<td>( \times )</td>
<td>✔</td>
</tr>
<tr>
<td>MKFHE [11]</td>
<td>( \tilde{O}(k^{3}\lambda^{15}L^{11}) )</td>
<td>( \tilde{O}(Nk^{4}\lambda^{6}L^{6}) )</td>
<td>( 2^{O(\lambda L)}B_x )</td>
<td>( \tilde{O}(Nk^{3}\lambda^{15}L^{10}) )</td>
<td>2 rounds</td>
<td>( \times )</td>
</tr>
<tr>
<td>Th-FHE [5]</td>
<td>( \tilde{O}(\lambda^{6}L^{4}) )</td>
<td>( \tilde{O}(N\lambda^{6}L^{4}) )</td>
<td>( 2^{O(\lambda L)}B_x )</td>
<td>( \times )</td>
<td>1 rounds</td>
<td>✔</td>
</tr>
<tr>
<td>Scheme#1</td>
<td>PubKey + EvalKey : ( \tilde{O}(N(k\lambda L + W)^{2}L^{2}) )</td>
<td>( \tilde{O}(N(k\lambda L + W)^{2}L^{2}) )</td>
<td>( 2^{O(\lambda L)}B_x )</td>
<td>( \times )</td>
<td>2 rounds</td>
<td>( \times )</td>
</tr>
</tbody>
</table>

**Table 2.** 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 \( \mathbb{Z}_q \); \( \lambda \) denotes the security parameter, \( k \) participants number, \( B_x \) the initial LWE noise; \( N, L, W \) denotes the input length, depth, and output length of the circuit respectively.

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

**Scheme\#2: RLWE-based KL-MKFHE under ROM:**
Same as the scheme in [13], Scheme#2 is based on circular-RLWE. We introduce a bit commitment protocol to guarantee the randomness of each participant’s public key. Due to the sum key structure, the dimension of $t \otimes t$ is independent of the number of participant $k$, so the ciphertext relinearization algorithm pulls the ciphertext after tensor product back to initial dimension by one shot, in addition, the “one shot algorithm” introduces less noise. We note that, as we mention before, regularity lemma on polynomial ring: $\mathbb{Z}[x]/x^d + 1$ does not enjoy the leakage resilient property, we have to introduce smudging noise in partial decryption phase as other RLWE-based MKFHE.

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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Evalkey</th>
<th>Ciphertext</th>
<th>Time</th>
<th>Interaction(Setup phase)</th>
<th>CRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKFHE [13]</td>
<td>$\tilde{O}(kd)$</td>
<td>$\tilde{O}(kd)$</td>
<td>$\tilde{O}(k^2d)$</td>
<td>$\tilde{O}(kd)$</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheme#2</td>
<td>$\tilde{O}(kd)$</td>
<td>$O(d)$</td>
<td>$O(1)$</td>
<td>$\tilde{O}(d)$</td>
<td>2 rounds</td>
</tr>
</tbody>
</table>

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

2 Preliminaries

2.1 Notation:

We give the definitions of the relevant notations in Table 4. Let $\text{negl}(\lambda)$ a negligible function parameterized by $\lambda$. Vectors are represented by lowercase bold letters such as $\mathbf{v}$, unless otherwise specified. Vectors are row vectors by default.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>security parameter</th>
<th>$n$</th>
<th>dimension of LWE problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>number of participants</td>
<td>$d$</td>
<td>degree of RLWE problem</td>
</tr>
<tr>
<td>$N$</td>
<td>circuit input length</td>
<td>$q$</td>
<td>module base</td>
</tr>
<tr>
<td>$L$</td>
<td>circuit multiplicative depth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W$</td>
<td>circuit output length</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.
and matrices are represented by uppercase bold letters such as \( \mathbf{M} \). \( [k] \) denotes the set of integers \( \{1, \ldots, k\} \). If \( X \) is a distribution, then \( a \leftarrow X \) denotes that value \( a \) is chosen according to the distribution \( X \), or a finite set, then \( a \leftarrow U(X) \) denotes that the value of \( a \) is uniformly sampled from \( X \). For two distribution \( X, Y \), we use \( X \approx Y \) to represent \( X \) and \( Y \) are statistically indistinguishable, while \( X \approx^\text{stat} Y \) are computationally indistinguishable.

In order to decompose elements in \( \mathbb{Z}_q \) into binary, we review the Gadget matrix \([31][2] \) here, let \( \mathbf{G}^{-1}(\cdot) \) be the computable function that for any

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

Let \( g = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l \), \( \mathbf{G} = \mathbf{I}_m \otimes g \in \mathbb{Z}_q^{m \times ml} \), 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:

\[
\text{Pr}_{x \in \mathcal{D}_n}[|e| > B] = \text{negl}(n).
\]

**Lemma 1** (Smudging lemma \([5]\) ) Let \( B_1 = B_1(\lambda) \), and \( B_2 = B_2(\lambda) \) be positive integers and let \( e_1 \in [-B_1, B_1] \) be a fixed integer, let \( e_2 \in [-B_2, B_2] \) be chosen uniformly at random. Then the distribution of \( e_2 \) is statistically indistinguishable from that of \( e_2 + e_1 \) as long as \( B_1/B_2 = \text{negl}(\lambda) \).

**Theorem 1** (\([27, \text{Theorem 5.3.2}]\) ) 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 larger than \( m + t \), is at most \( e^{-t^2/(m+t)} \).

**The Rènyi divergence in \([6]\) :** 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 \( a \in (1, +\infty) \), we define the The Rènyi divergence of order \( a \) by:

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

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

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

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

The definitions are extended in the natural way to continuous distributions. The divergence \( R_1 \) is the (exponential of ) the Kullback-Leibler divergence.
Theorem 2 ([6, Theorem 4.2]) Let $\Phi, \Phi'$ denote two distribution with $\text{Supp}(\Phi) \subseteq \text{Supp}(\Phi')$, and $\text{Supp}(\Phi) \subseteq \text{Supp}(\Phi')$. 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 $A$ for problem $P$ with advantage $\epsilon$, we can construct a distinguisher $A'$ for problem $P'$ with run-time and distinguishing advantage, respectively, bounded from above and below by (for any $a \in (1, +\infty)$):

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

2.3 Gaussian distribution on Lattice

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

\[
D_{\Lambda+c,s}(x) = \frac{\rho_s(x)}{\rho_s(\Lambda + c)}
\]

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

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

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

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

In particular, for any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon = \epsilon(n)$ such that $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$. 
Lemma 3 (Simplified version of [37, Theorem 3.1]) Let $\epsilon > 0, r_1, r_2 > 0$ be two Gaussian parameters, and $\Lambda \subset \mathbb{Z}^m$ be a lattice. If $\frac{r_1 r_2}{\sqrt{r_1^2 + r_2^2}} \geq \eta_\epsilon(\Lambda)$, then

$$\Delta(y_1 + y_2, y') \leq 8\epsilon$$

where $y_1 \leftarrow \mathcal{D}_{\Lambda, r_1}$, $y_2 \leftarrow \mathcal{D}_{\Lambda, r_2}$, and $y' \leftarrow \mathcal{D}_{\Lambda, \sqrt{r_1^2 + r_2^2}}$.

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

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

2.4 The Learning With Error (LWE) Problem

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

Definition 4 (Decision-LWE) Let $\lambda$ 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 LWE$_{n,q,\chi}$ problem is to distinguish the following distribution:

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

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

2.5 The Ring Learning With Error (RLWE) Problem

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

Definition 5 (Decision-RLWE) Let $\lambda$ be a security parameter. For parameters $d = d(\lambda)$, where $d$ is a power of 2, $q = q(\lambda) > 2$, and a distribution $\chi = \chi(\lambda)$ over $R = \mathbb{Z}[x]/x^d + 1$, let $R_q = R/qR$, the Decision-RLWE$_{d,q,\chi}$ problem is to distinguish the following distribution:

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

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

The DGSW scheme [11] and GSW scheme is similar to Dual-Regev scheme and Regev scheme resp. which is defined as follows:

- \( pp \leftarrow \text{Gen}(1^\lambda, 1^L) \): For a given security parameter \( \lambda \), 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 \( B_\chi \), module \( q = \text{poly}(n) \cdot B_\chi \), Output \( pp = (n, m, q, \chi, B_\chi) \) as the initial parameters.
- \((pk, sk) \leftarrow \text{KeyGen}(pp)\): Let \( sk = t = (-s, 1) \), \( pk = (A, b) \), where \( s \leftarrow U(\{0, 1\}^{m-1}) \), \( A \leftarrow U(\mathbb{Z}_{m-1}^{1 \times n}) \), \( b = sA \mod q \).
- \( C \leftarrow \text{Enc}(pk, u)\): Input public key \( pk \) and plaintext \( u \in \{0, 1\} \), choose a random matrix \( R \leftarrow U(\mathbb{Z}_q^{m \times w}) \), \( w = ml \), \( l = \lceil \log q \rceil \) and an error matrix \( E \leftarrow \chi^{m \times w} \), Output the ciphertext:
  \[
  C = \left( \begin{array}{c} A \\ b \end{array} \right) R + E + uG, \quad \text{where } G \text{ is a gadget Matrix.}
  \]
- \( u \leftarrow \text{Dec}(sk, C)\): Input private key \( sk \), ciphertext \( C \), let \( w = (0, \ldots, \lfloor q/2 \rfloor) \in \mathbb{Z}_q^{m}, v = (tC, G^{-1}(w^T)) \), output \( u' = \lceil \frac{v}{q/2} \rceil \).

**Homomorphic addition and multiplication**: For ciphertext \( C_1, C_2 \in \mathbb{Z}_q^{m \times w} \), let \( C_{\text{add}} = C_1 + C_2 \), \( C_{\text{mult}} = C_1 G^{-1}(C_2) \). It is easy to verify that \( C_{\text{add}} \) and \( C_{\text{mult}} \) are ciphertext of \( u_1 + u_2 \) and \( u_1 u_2 \), respectively.

For the security and correctness of the DGSW scheme, please refer to [11]. Compared with the GSW scheme, DGSW scheme has bigger ciphertext, which is \( O(n^2 \log^3 q) \), while \( O(n^2 \log q) \) for GSW scheme. As [11] mentioned, DGSW scheme makes it more convenient to use the leakage resilient property of LHL to remove CRS.

2.7 Multi-Key Fully Homomorphic Encryption

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

**Definition 6** Let \( \lambda \) be the security parameter, \( L \) be the circuit depth, and \( k \) be the number of participants. A leveled multi-key fully homomorphic encryption scheme consists of a tuple of efficient probabilistic polynomial time algorithms MKFHE = (\text{Init}, \text{Gen}, \text{Enc}, \text{Expand}, \text{Eval}, \text{Dec}) \) which defines as follows.

- \( pp \leftarrow \text{Init}(1^\lambda, 1^L) \) : Input security parameter \( \lambda \), circuit depth \( L \), output system parameter \( pp \). We assume that all algorithm take \( pp \) as input.
- \((pk_i, sk_i) \leftarrow \text{Gen}(pp, \text{crs})\) : Input \( pp \), common reference string \( \text{crs} \) (generated by a third party or random oracle), output a key pair for participant \( i \).
- \( c_i \leftarrow \text{Enc}(pk_i, u_i)\) : Input \( pk_i \) and plaintext \( u_i \), output ciphertext \( c_i \).
– $v_i \leftarrow \text{Enc}(pk_i, r_i)$: Input $pk_i$ and the random $r_i$ used in ciphertext $c_i$, output auxiliary ciphertext $v_i$.
– $\tilde{c}_i \leftarrow \text{Expand}\{|pk_i|_{i \in [k]}, v_i, c_i\}$: Input the ciphertext $c_i$ of participant $i$, the public key set $\{|pk_i|_{i \in [k]}\}$ of all participants, auxiliary ciphertext $v_i$, output expanded ciphertext $\tilde{c}_i$ which is under $f(sk_i, \ldots, sk_k)$ whose structure is undefined.
– $\bar{c}_{\text{eval}} \leftarrow \text{Eval}(S, C)$: Input circuit $C$, the set of all ciphertext $S = \{\tilde{c}_i\}_{i \in [N]}$ while $N$ is the input length of circuit $C$, output evaluated ciphertext $\bar{c}_{\text{eval}}$.
– $u \leftarrow \text{Dec}(\bar{c}_{\text{eval}}, f(sk_1, \ldots, sk_k))$: Input evaluated ciphertext $\bar{c}_{\text{eval}}$, private key function $f(sk_1, \ldots, sk_k)$, output $u$ (This is usually a distributed process).

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

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

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

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

Definition 7 A KL-MKFHE scheme is a tuple of probabilistic polynomial time algorithm $(\text{Init, Gen, KeyLifting, Enc, Eval, Dec})$, which can be divided into two phases, online phase: KeyLifting and Dec, where interaction is allowed between participants, but the rounds should be constant, local phase: Init, Gen, Enc, and Eval, whose operations do not involve interaction. These five algorithms are described as follows:

– $pp \leftarrow \text{Init}(1^\lambda, 1^L)$: Input security parameter $\lambda$, circuit depth $L$, output public parameters $pp$.
– $(pk_i, sk_i) \leftarrow \text{Gen}(pp)$: Input public parameter $pp$, output the key pair of participant $i$. 
– \{hk_i\}_{i \in [k]} \leftarrow \text{KeyLifting}(\{pk_i, sk_i\}_{i \in [k]}): \text{Input key pair } \{pk_i, sk_i\}_{i \in [k]} \text{ of all participants, output the hybrid key } \{hk_i\}_{i \in [k]} \text{ of all } i. \text{(online phase: two round interaction)}

– c_i \leftarrow \text{Enc}(hk_i, u_i): \text{Input plaintext } u_i \text{ and } hk_i, \text{ output ciphertext } c_i

– \hat{c} \leftarrow \text{Eval}(C, S): \text{Input circuit } C, \text{ ciphertext set } S = \{c_i\}_{i \in [N]}, \text{ output ciphertext } \hat{c}

– u \leftarrow \text{Dec}(\hat{c}, f(sk_1 \ldots sk_k)): \text{Input evaluated ciphertext } \hat{c}, f(sk_1 \ldots sk_k), \text{ output } C(u_i)_{i \in [N]}. \text{(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 participants under their own hybrid key \(hk_i\) which are different among participants, however, the resulting ciphertext \(c_i\) supports homomorphic evaluation without extra modification.

we require KL-MKFHE to satisfy the following properties:

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

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

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

\[
\Pr [A(pp, pk, \text{Enc}(pk, 1)) - A(pp, pk, \text{Enc}(pk, 0)) \neq 0] = \text{negl}(\lambda).
\]

**Correctness and Compactness:** A KL-MKFHE scheme is correct if for a given security parameter \(\lambda\), circuit depth \(L\), participants \(k\), we have the following

\[
\Pr \left[ \text{Dec}(f(sk_1 \ldots sk_k), \hat{c}) \neq C(u_1 \ldots u_N) \right] = \text{negl}(\lambda).
\]

probability is negligible, where \(C\) 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 **Scheme#1:** A KL-MKFHE scheme based on DGSW in plain model without noise flooding

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

Following the definition of KL-MKFHE, the hybrid keys \( \{hk_i\}_{i \in [k]} \) which are obtained by KeyLifting(·) algorithm are different from each other. Each participant encrypts his own plaintext \( u_i \) by \( hk_i \) and get \( C_i \). The ciphertexts \( \{C_i \in [N]\} \) can be used to evaluation without extra computation by Claim 1. We achieve this property by allowing two round interaction between participants.

Key Lifting

\[ \{hk_i\}_{i \in [k]} \leftarrow \text{KeyLifting}(\{pk_i, sk_i\}_{i \in [k]}): \text{Input the DGSW key pair} \{pk_i, sk_i\}_{i \in [k]} \]

of all participants, where \( pk_i = (A_i, b_{i,i}) \), \( A_i \leftarrow U(Z_q^{(m-1) \times n}), s_i \leftarrow U \{0, 1\}^{m-1}, b_{i,i} = s_i A_i \mod q \). Assuming there is a broadcast channel, all participants are engaged in the following two interaction:

- First round: \( i \) broadcasts \( A_i \) and receives all \( \{A_j\}_{j \in [k] \setminus i} \).
- Second round: \( i \) generates and broadcasts \( \{b_{i,j}\}_{j \in [k]} \), where \( b_{i,j} = s_i A_j \mod q \)

After above two round interaction, \( i \) receives \( \{b_{j,i}\}_{j \in [k] \setminus i} \)

\[
\text{let } b_i = \sum_{j=1}^{k} b_{j,i}, \text{ } i \text{ obtains hybrid key } hk_i = (A_i, b_i)
\]

Claim 1 Let \( \bar{t} = (-s, 1) \), \( s = \sum_{i=1}^{k} s_i \), for ciphertext \( C_i, C_j \) encrypted by hybrid key \( hk_i, hk_j \) respectively:

\[
C_i = \begin{pmatrix} A_i \\ b_i \end{pmatrix} R_i + E_i + u_i G, \quad C_j = \begin{pmatrix} A_j \\ b_j \end{pmatrix} R_j + E_j + u_j G,
\]

we have (omit small error):

\[
\bar{t}C_i \approx u_i \bar{t}G, \quad \bar{t}C_j \approx u_j \bar{t}G
\]

\[
\bar{t}(C_i + C_j) \approx (u_i + u_j)\bar{t}G, \quad \bar{t}C_i G^{-1}(C_j) \approx (u_i u_j)\bar{t}G
\]

Proof. According to the construction of KeyLifting(·) we have:

\[
\bar{t}C_i = \sum_{i=1}^{k} (-s_i, 1) \left[ \begin{pmatrix} A_i \\ \sum_{j=1}^{k} b_{j,i} \end{pmatrix} + E_i + u_i G \right] = \bar{t}E_i + u_i \bar{t}G \approx u_i \bar{t}G.
\]

Similarly, \( \bar{t}C_j \approx u_j \bar{t}G \), and \( \bar{t}(C_i + C_j) \approx (u_i + u_j)\bar{t}G \)

\[
\bar{t}C_i G^{-1}(C_j) \approx u_i \bar{t}GG^{-1}(C_j) \approx u_i \bar{t}C_j \approx (u_i u_j)\bar{t}G
\]

\[ \blacksquare \]
Therefore, although $C_i$ and $C_j$ are encrypted by different hybrid keys, they correspond to the same decryption key $i$ and support homomorphic evaluation without extra modification.

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

We note that as long as our encryption scheme is leakage-resistant, properly lengthening private key $s$ can guarantee the semantic security of the scheme even if part of $s$ is leaked. For the second problem, the general solution is to assume that $b_{j,i}$ generated by adversary $j$ satisfies the linear relationship $b_{j,i} = s_j A_i$, $s_j \in \{0,1\}^{m-1}$, and introduce a large noise in the encryption process to ensure security. Large encryption noise leads to large modulus $q$, which further leads to high computational and communication overhead. In order to alleviate this problem, we proposed an analysis method based on Rényi divergence that neither introduces above assumptions nor a large noise in the encryption process. For more details, please refer to Section 4.4.

4.2 The entire scheme

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

\[- \text{pp} \leftarrow \text{Init}(1^{\lambda}, 1^L, 1^W) : \text{Let } \lambda \text{ be security parameter, } L \text{ circuit depth, } W \text{ circuit output length, lattice dimension } n = n(\lambda, L), \text{ noise distribution } \chi \text{ over } \mathbb{Z}, e \leftarrow \chi, \text{ where } |e| \text{ is bounded by } B_{\chi} \text{ with overwhelming probability, modulus } q = 2^{O(L)} B_{\chi}, k = \text{poly}(\lambda), m = (kn + W) \log q + \lambda, \text{ suitable choosing above parameters to make } \text{LWE}_{n,m,q,B_{\chi}} \text{ is infeasible. Output } \text{pp} = (k, n, m, q, \chi, B_{\chi}) \]

\[- \text{pk}_i, \text{sk}_i \leftarrow \text{Gen(pp)} : \text{Input } \text{pp}, \text{ output the DGSW key pair } (\text{pk}_i, \text{sk}_i) \text{ of participants } i, \text{ where } \text{pk}_i = (A_i, b_{i,i}), A_i \leftarrow U(\mathbb{Z}_q^{(m-1) \times n}), s_i \leftarrow U(\{0,1\}^{m-1}), \]  

\[b_{i,i} = s_i A_i \mod q.\]

\[- \text{hk}_i \leftarrow \text{KeyLifting}(\{\text{pk}_i, \text{sk}_i\}_{i \in [k]}) : \text{All participants are engaged in the Key lifting procedure 4.1, output the hybrid key } \text{hk}_i.\]

\[- \text{C}_i \leftarrow \text{Enc}(\text{hk}_i, u_i) : \text{Input hybrid key } \text{hk}_i, \text{ plaintext } u_i \in \{0,1\}, \text{ output ciphertext } \text{C}_i = \begin{pmatrix} A_i \\ b_{i,i} \end{pmatrix} \text{R} + \text{E} + u_i \text{G}, \text{ where } \text{R} \leftarrow U(\mathbb{Z}_q^{n \times ml}), l = \lceil \log q \rceil, \]  

\[\text{E} \leftarrow \chi^{m \times ml}, \text{ G} = I_m \otimes g \text{ is a gadget matrix.}\]

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

**Remark :** In the security proof in Section 4.5, we require that $\chi$ be a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}, \sigma}$ over $\mathbb{Z}$ with $\sigma > \sqrt{2} \eta_\epsilon(\mathbb{Z})$. and $ml > 4L$. 

Homomorphic addition and multiplication: Let $C_i$, $C_j$ be ciphertext under hybrid key $hk_i$ and $hk_j$ respectively, by claim 1, we have the following results.

- $C_{\text{add}} \leftarrow \text{Add}(C_i, C_j)$: Input ciphertext $C_i$, $C_j$, output $C_{\text{add}} = C_i + C_j$, which $iC_{\text{add}} \approx (u_i + u_j)\tilde{t}G$
- $C_{\text{mult}} \leftarrow \text{Mult}(C_i, C_j)$: Input ciphertext $C_i$, $C_j$, output $C_{\text{mult}} = C_iG^{-1}(C_j)$, which $iC_{\text{mult}} \approx u_i u_j \tilde{t}G$

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

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

4.3 Correctness analysis

To illustrate the correctness of Scheme#1, we first study the accumulation of noise. For fresh ciphertext $C = \begin{pmatrix} A_i \\ b_i \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix} + uG$ under $\tilde{t}$, it holds that $\tilde{t}C = e_1 - sE_0 + u\tilde{t}G$. Let $e_{\text{init}} = e_1 - sE_0$, after $L$ depth circuit evaluation:

$$\tilde{t}C^{(L)} = e_L + u_L \tilde{t}G$$ (1)

According to the noise analysis of GSW in [23], the noise $e_L$ in $C^{(L)}$ is bounded by $(ml)^2e_{\text{init}}$. By the distributed decryption of Scheme#1 we have:

$$\gamma = \sum_{i=1}^k \gamma_i + \langle c_{\text{low}}, G^{-1}(w^T) \rangle = \sum_{i=1}^k -s_i, C_{\text{up}}G^{-1}(w^T) + \langle c_{\text{low}}, G^{-1}(w^T) \rangle$$

$$= iC^{(L)}G^{-1}(w^T) = \langle e_L, G^{-1}(w^T) \rangle + u_L \lceil \frac{q}{2} \rceil$$

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

$$\langle e_L, G^{-1}(w^T) \rangle \leq \lambda \cdot \|e_L\|_{\infty} \leq \lambda \cdot (ml)^L \cdot \|e_{\text{init}}\|_{\infty} \leq \lambda \cdot (ml)^L \cdot (km + 1)B_{\chi}$$

Thus, $\log(|e_L, G^{-1}(w^T)|) = O(L)$. For those $q = 2^{O(L)}B_{\chi} \geq 4\langle e_L, G^{-1}(w^T) \rangle$, requirements are fulfilled.
4.4 Semantic Security of Encryption against Semi-Malicious Adversary

The concept of a semi-malicious adversary was proposed by Asharov et al. in [5], 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. For a more formal definition, please refer to [5]. First, we note that DGSW is leakage-resilient, so Scheme#1 is leakage-resilient as well, and second, we prove Scheme#1's semantic security.

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

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

**Remark:** Note that in the proof of [11], the condition for the establishment of Lemma 5 is \( |e/e'| = \text{negl}(\lambda) \). We point out that this condition is not required with our analytical method. Because it is similar to our analysis method below, we prove it in the Appendix B.

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

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

**A common approach:** We complete the simulation by constructing a reduction from Scheme#1 to the DGSW scheme. Consider the following Game:
1. Challenger generates $\text{pk}_{\text{DGSW}} = (A, b_1)$ where $A \leftarrow U(Z_q^{(m-1)\times n})$, $b_1 = s_1A$, $s_i \leftarrow U\{0,1\}^{m-1}$ and send $\text{pk}_{\text{DGSW}}$ to adversary $A$.

2. $A$ adaptively chooses $\{b_i\}_{i \in [k]/1}$ where $b_i = s_iA$ and $s_i \in \{0,1\}^{m-1}$ after seeing $\text{pk}_{\text{DGSW}}$, chooses a bit $u \in \{0,1\}$ and sets $\text{hk}_{\text{Scheme#1}} = (A, b)$, where $b = \sum_{i=1}^k b_i$, then send $\text{hk}_{\text{Scheme#1}}$ and $u$ to Challenger.

3. Challenger chooses a bit $\alpha \in \{0,1\}$, if $\alpha = 0$, set $C_{\text{Scheme#1}} \leftarrow \text{Enc}(\text{hk}_{\text{Scheme#1}}, u)$, otherwise $C_{\text{Scheme#1}} \leftarrow U(Z_q^{m\times ml})$, and send $C_{\text{Scheme#1}}$ to $A$.

4. After receiving $C_{\text{Scheme#1}}$, $A$ output bit $\bar{\alpha}$, if $\bar{\alpha} = \alpha$, then $A$ wins.

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

**Proof.** We construct $C_{\text{Scheme#1}}$ by DGSW.$\text{Enc}(\text{pk}_{\text{DGSW}}, u)$:

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

$$\text{DGSW.} \text{Enc}(\text{pk}_{\text{DGSW}}, 0) = \begin{pmatrix} A \\ b_1 \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix} = \begin{pmatrix} C_0 \\ e_1 \end{pmatrix}$$

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

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

$$s'C_0 = s'(AR + E_0) = \sum_{i=2}^k b_iR + s'E_0$$

$$C' = \text{DGSW.} \text{Enc}(\text{pk}_{\text{DGSW}}, 0) + \begin{pmatrix} 0 \\ s'C_0 \end{pmatrix}$$

$$= \begin{pmatrix} A \\ b_1 \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix} + \begin{pmatrix} 0 \\ s'C_0 \end{pmatrix}$$

$$= \begin{pmatrix} A \\ b \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 + s'E_0 \end{pmatrix}$$

If $||e_1||_\infty$ is bounded by $2^\lambda B$, and $||s'E_0||_\infty < kmB$, thus $s'E_0/e_1 = \text{negl}(\lambda)$. By Lemma 1, we have $C' \overset{\text{stat}}{=} C_{\text{Scheme#1}}$ if $A$ can distinguish between $C_{\text{Scheme#1}}$ and uniform random distribution by advantage $\text{Adv}$, then he can distinguish between $\text{DGSW.} \text{Enc}(\text{pk}_{\text{DGSW}}, u)$ and the uniform random distribution with the same advantage.

**Remark:** When $||e_1||_\infty$ is bounded by $2^\lambda B$, according to the correctness analysis in Section 4.3, the initial noise $e_{init} = e_1 - sE_0$ is bounded by $(2^\lambda + km)B$. After $l$-level evaluation, $(e_L, G^{-1}(w^T))$ is bounded by $l\cdot(ml)^L\cdot(2^\lambda + km)B$, $\log((e_L, G^{-1}(w^T))) = O(\lambda + L)$. Thus result in a $q = 2^{O(\lambda + L)}B$.
Rényi divergence-based optimization: The work of Shi et al [6] 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 this results, they obtained better parameters of 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 the $\text{Scheme#1}'s$ 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.

Let $0^{1\times ml}$ be a zero vector of length $ml$, $\Phi$ be the distribution of hybrid key of $\text{Scheme#1}$ followed by $0^{1\times ml}$:

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

which determined by KeyLifting$(\cdot)$ procedure. Let $D_0(A, b, 0^{1\times ml})$ be the joint distribution of $(A, b, 0^{1\times ml})$ and the ciphertext of $\text{Scheme#1} (\begin{pmatrix} A \\ b \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix})$ over the randomness $R, E_0, e_1$:

$$(A, b, 0^{1\times ml}, (\begin{pmatrix} A \\ b \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix}) \leftarrow D_0(A, b, 0^{1\times ml})$$

Let $D_1(A, b, 0^{1\times ml})$ be the joint distribution of $(A, b, 0^{1\times ml})$ and $U \leftarrow U(\mathbb{Z}_q^{m\times ml})$:

$$(A, b, 0^{1\times ml}, U) \leftarrow D_1(A, b, 0^{1\times ml})$$

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

$$X_0 = \{ x : (A, b, 0^{1\times ml}) \leftarrow \Phi, \quad x = (A, b, 0^{1\times ml}, (\begin{pmatrix} A \\ b \end{pmatrix} R + \begin{pmatrix} E_0 \\ e_1 \end{pmatrix}) \leftarrow D_0(A, b, 0^{1\times ml}) \},$$

$$X_1 = \{ x : (A, b, 0^{1\times ml}, U) \leftarrow D_1(A, b, 0^{1\times ml}) \}.$$
from some distribution. For random $R \in \mathbb{Z}_q^{n \times m_l}$, without loss of generality, assuming $\frac{m_l}{n} = \rho$, we can divide $R$ into $\rho$ square matrices:

$$R = (R_1, R_2, \ldots, R_\rho)$$

where $R_i \in \mathbb{Z}_q^{n \times n}$. Similarly, for $E_0 \in \mathbb{Z}_q^{(m-1) \times m_l}$, $e_1 \in \mathbb{Z}_q^{m_l}$:

$$E_0 = (E_{0,1}, E_{0,2}, \cdots, E_{0,\rho})$$

$$e_1 = (e_{1,1}, e_{1,2}, \cdots, e_{1,\rho})$$

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

$$C' = \left( bR_1 + s'E_{0,1} + e_{1,1}, bR_2 + s'E_{0,2} + e_{1,2}, \cdots, bR_\rho + s'E_{0,\rho} + e_{1,\rho} \right)$$

Let $\{v_i \in \mathbb{Z}_q^n\}_{i \in \rho}$ be the solution of equation:

$$\{v_i, R_i = s'E_{0,i}\}_{i \in \rho}$$

Obviously, if $R_i$ is random over $\mathbb{Z}_q^{n \times n}$, then $v_i$ has a unique solution with an overwhelming probability. Define set $V$:

$$V = \{0^{1 \times m_l}, (v_1, v_2, \cdots, v_\rho)\}$$

Define the distribution $d \leftarrow D(V)$ over set $V$:

$$\{d \leftarrow D(V) : \Pr(d = 0^{1 \times m_l}) = p \land (v_1, v_2, \cdots, v_\rho)) = 1 - p\}$$

Let $\Phi'$ be the distribution of hybrid key of Scheme#1 followed by $d$ which is sampled from $D(V)$:

$$(A', b', d) \leftarrow \Phi'$$

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

$$X_0' = \{x : (A', b', d) \leftarrow \Phi',$$

$$x = (A', b', d, \left( (b' + d_1)R_1' + e_1' (b' + d_2)R_2' + e_2' , \cdots, (b' + d_\rho)R_\rho' + e_\rho' \right) \} \leftarrow D_0(A', b', d)\}$$

$$X_1' = \{x : (A', b', d) \leftarrow \Phi',$$

$$x = (A', b', d, U) \leftarrow D_1(A', b', d)\}.$$}

where $R' = (R_1', R_2', \cdots, R_\rho')$, $d = (d_1, d_2, \cdots, d_\rho)$

Thus for any $C'$:

$$C' = \left( bR_1 + s'E_{0,1} + e_{1,1}, bR_2 + s'E_{0,2} + e_{1,2}, \cdots, bR_\rho + s'E_{0,\rho} + e_{1,\rho} \right)$$
it is a sample:

\[ x = (A', b', d, (b' + a_1)R'_1 + e'_1, (b' + a_2)R'_2 + e'_2, \cdots, (b' + a_y)R'_y + e'_y) \]

of \( X'_0 \) with \( A' = A, b' = b, R' = R, E'_0 = E_0, d = (v_1, v_2, \cdots, v_y), e'_i = e_{1,i} \).

We note that \( C' \) only forms part of sample of \( X'_0 \). The completed sample also contains \( \{v_i\}_{i \in [g]} \) which are determined by \( \{v_iR_i = s'E_{0,i}\}_{i \in [g]} \) where \( R_i, E_{0,i} \) is generated by challenger, \( s' \) is generated by adversary. Consider the following process:

- Challenger generates DGSW ciphertext \( C = (A \ b \ E_0) \) and sends it to adversary.
- After receiving DGSW ciphertext, adversary adaptively generates \( s', C' \), and sends \( s' \) to the challenger.
- Challenger computes \( \{v_i\}_{i \in [g]} \) by \( \{v_iR_i = s'E_{0,i}\}_{i \in [g]} \) and sends it to adversary.
- Adversary constructs a complete \( X'_0 \) sample from \( C' \) and \( \{v_i\}_{i \in [g]} \).

Note that exposing \( \{v_i\}_{i \in [g]} \) to adversary will reveal the linear relationship between \( R_i \) and \( E_{0,i} \). We need to ensure that after the adversary gets \( \{v_i\}_{i \in [g]} \), the DGSW ciphertext is still indistinguishable. Let’s take a look at what is the distribution of \( \{v_i\}_{i \in [g]} \). For \( v_iR_i = s'E_{0,i} \), thus \( v_i = s'E_{0,i}R_i^{-1} \). Because \( R_i \) is uniform over \( \mathbb{Z}_q^{n \times n} \) and \( E_{0,i} \) is discrete Gaussian over \( \mathbb{Z}_q^{(m-1) \times n} \), so \( E_{0,i}R_i^{-1} \) is uniform over \( \mathbb{Z}_q^{(m-1) \times n} \) and unknown to adversary. Therefore, when \( s' \neq 0 \), \( v_i \) is uniform random over \( \mathbb{Z}_q^m \), that is, \( v_i \) and \( s' \) are independent except at zero.

Consider the first \( m-1 \) elements of the first column of the DGSW ciphertext and the first element of equation \( v_1R_1 = s'E_{0,1} \):

\[
\begin{pmatrix}
  a_1, a_2, \cdots, a_n
\end{pmatrix}
\begin{pmatrix}
  r_1 \\
  r_2 \\
  \cdots \\
  r_n
\end{pmatrix}
+ 
\begin{pmatrix}
  e_1 \\
  e_2 \\
  \cdots \\
  e_{m-1}
\end{pmatrix}
= 
\begin{pmatrix}
  c_1 \\
  c_2 \\
  \cdots \\
  c_{m-1}
\end{pmatrix}
\]

Let \( e = (e_1, e_2, \cdots, e_{m-1}), c = (c_1, c_2, \cdots, c_{m-1}) \).

**Claim 3** If there is an adversary can distinguish \( (A, c, v_1, s') \) with \( (A, z, v_1, s') \) where \( z \leftarrow \mathbb{Z}_q^{(m-1)} \), then he can also distinguish \( n-1 \) dimensional LWE samples. (we note that \( (A, c) \) are \( n \) dimensional LWE samples.)

**Proof.** We show that for \( m-1 \) LWE sample of \( n-1 \) dimensions, and a vector \( u = (u_1, u_2, \cdots, u_{m-1}) \) generated by adversary. We can always construct a sample of
(A, c, v₁, s'). For m − 1 LWE sample of n − 1 dimensional and u:

\[
\begin{pmatrix}
(a'_1, a'_2, \ldots, a'_{n-1}) & (r_1, r_2, \ldots, r_{n-1}) + (c'_1, c'_2, \ldots, c'_{m-1})
\end{pmatrix} = \begin{pmatrix}
(r_1, r_2, \ldots, r_{n-1}) + (c'_1, c'_2, \ldots, c'_{m-1})
\end{pmatrix}
\]

\[
u = (u_1, u_2, \ldots, u_{m-1})
\]

Let \(w_0 = t(w_1 e'_1 + w_2 e'_2 + \cdots + u_{m-1} e'_{m-1})\), where \(t \leftarrow U(\mathbb{Z}_q)\), \(\{w_i\}_{i \in [n-1]} \leftarrow U(\mathbb{Z}_q)\), \(r_n = w_0 - \sum_{i=1}^{n-1} w_i r_i\), \(a_n \leftarrow U(\mathbb{Z}_q^{m-1})\), \(\{a_i = a'_i + w_i a_n\}_{i \in [n-1]}\)

\[
\begin{pmatrix}
e_1 \\
e_2 \\
\vdots \\
e_{m-1}
\end{pmatrix} = \begin{pmatrix}
e'_1 \\
e'_2 \\
\vdots \\
e'_{m-1}
\end{pmatrix} + w_0 a_n
\]

Thus:

\[
\begin{pmatrix}
a_1, a_2, \ldots, a_n
\end{pmatrix} \begin{pmatrix}
r_1 \\
r_2 \\
\vdots \\
r_{n-1}
\end{pmatrix} + \begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{m-1}
\end{pmatrix} = \begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{m-1}
\end{pmatrix}
\]

\(t^{-1} w_1 r_1 + t^{-1} w_2 r_2 + \cdots + t^{-1} w_{n-1} r_{n-1} + t^{-1} r_n = u_1 e_1 + u_2 e_2 + \cdots + u_{m-1} e_{m-1}\)

We note that \(r_n\) is independent of \(\{r_i\}_{i \in [n-1]}\) for the reason that \(w_0\) is uniform over \(\mathbb{Z}_q\), and similarly \(\{a_i\}_{i \in [n]}\) are independent. Thus:

\[
\{\{a_i\}_{i \in [n]}, c_1, t_{i \in [m-1]}, (t^{-1} w_1, t^{-1} w_2, \cdots, t^{-1} w_{n-1}, t^{-1}), u\}
\]

is a sample of \((A, c, v_1, u)\). If there is an adversary can distinguish \((A, c, v_1, u)\), then he can also distinguish \(n - 1\) dimensional LWE samples.

So far, we have completed the construction of \(X'_0\) samples: that is, for each given DGSW ciphertext, after getting \(\{v_i\}_{i \in [\rho]}\) from challenger, the adversary can convert it into a sample of \(X'_0\). Our distributions \(D_0(\cdot)\) and \(D_1(\cdot)\) also satisfy the publicly sampleable property which required by Theorem 2: there exists a sampling algorithm \(S\) with run-time \(T_S\) such that for all \((\alpha, \beta)\), given any sample \(x\) from \(D_\alpha(\beta)\):

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

Then, by Theorem 2, if given a \(T\)-time distinguisher \(A\) for problem \(P\) with advantage \(\epsilon\), we can construct a distinguisher \(A'\) for problem \(P'\) with run-time and distinguishing advantage, respectively, bounded from above and below by (for any \(a \in (1, +\infty)\)):

\[
\frac{64}{\epsilon^2} \log \left(\frac{8 R'_n(\Phi||\Phi')}{\epsilon^{n/(n-1)}+1}\right) \cdot (T_S + T) \quad \text{and} \quad \frac{\epsilon}{4 \cdot R_n(\Phi||\Phi')} \cdot \left(\frac{\epsilon}{2}\right)^{\frac{1}{a+1}}.
\]
For convenience, we take $R_{\infty}(\Phi||\Phi')$ analysis, let:

$$R_{\infty}(\Phi||\Phi') = \max_{Y \in \text{Supp}(\Phi)} \frac{\Phi(Y)}{\Phi'(Y)} = \frac{\Phi(A_0, b_0, 0^{1 \times ml})}{\Phi'(A_0, b_0, 0^{1 \times ml})}.$$  

$$= \frac{\Pr(A = A_0, b = b_0)}{\Pr(A = A_0, b = b_0, d = 0^{1 \times ml})} = \frac{1}{p}.$$

Because $(A, b)$ and $D(V)$ are independent, thus:

$$(3) = \frac{\Pr(A = A_0, b = b_0)}{\Pr(A = A_0, b = b_0, d = 0^{1 \times ml})} = \frac{1}{p}$$

Then, given a $T$-time distinguisher $\mathcal{A}$ for problem $P$ with advantage $\epsilon$, 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, $\{s_{j \in [k]/i}\}$ is chosen uniformly random over $\{0, 1\}^{m-1}$, so $\Phi$ is a uniform random distribution over $\mathbb{Z}_m^{q \times n}$, and the security is obvious. Under the semi-malicious adversary model, the common approach assumes $b_{j,i} = s_{j}A_{i}$ and $\{s_{j \in [k]/i}\} \in \{0, 1\}^{m-1}$ is chosen adaptively, and introduces large noise in the encryption process to ensure security. In Rényi divergence-based optimization, we need to introduce neither above assumptions nor large encryption noise.

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

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

We note that the introduction of noise flooding in the partial decryption phase is essentially to guarantee the semantic security of fresh ciphertext, and noise flooding achieves this by masking the private key information in the partial decryption noise. For partial decryption to be simulatable, the magnitude of the noise introduced needs to be exponentially larger than the noise after the homomorphic evaluation. At the same time, as mentioned in [35], masking techniques based on noise flooding can only guarantee weak simulatable properties: input all private keys $\{sk_{j}\}_{j \in [k]/i}$ except $sk_i$, evaluated result $u_{L}$, ciphertext $C^{(k)}$, it can simulate the local decryption result $\gamma_i$, while for stronger security requirements: input any private key set $\{sk_{j}\}_{j \in S}$ for any subset $S$ of $[k]$, evaluated result $u_{eval}$ and ciphertext $C^{(L)}$, to simulate $\{\gamma_i\}_{i \in U, U = [k] - S}$, it don’t know how to achieve it.
With noise flooding: To illustrate how our approach works, let’s first review the noise flooding technique. Let $C^{(L)} = \left( \begin{array}{c} C_{up} \\ c_{low} \end{array} \right)$ 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:

$$\gamma_i = \langle -s_i, C_{up} G^{-1}(w^T) \rangle + e''_i$$

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

$$\gamma_i = u_L \left[ \frac{q}{2} + \langle e_L, G^{-1}(w^T) \rangle + e''_i - \langle c_{low}, G^{-1}(w^T) \rangle \right] + \sum_{j \neq i}^{k} s_j, C_{up} G^{-1}(w^T)$$

For a simulator $S$, input $\{sk_j\}_{j \in [k]/i}$, evaluated result $u_L$, ciphertext $C^{(L)}$, output simulated $\gamma'_i$:

$$\gamma'_i = u_L \left[ \frac{q}{2} + e''_i - \langle c_{low}, G^{-1}(w^T) \rangle \right] + \sum_{j \neq i}^{k} s_j, C_{up} G^{-1}(w^T)$$

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

$$\langle e_L, G^{-1}(w^T) \rangle + e''_{i \text{ stat}} \approx e''_i$$

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

$$|\langle e_L, G^{-1}(w^T) \rangle / e''_{i \text{ stat}}| = \text{negl}(\lambda)$$

thus $\gamma_i \approx \gamma'_i$.

In short, the noise $e''_i$ is introduced to cover up some information(private key $s_i$ and the noise $E_i$ in initial ciphertext) of participant $i$ contained in $e_L$ (Noise after decrypting the ciphertext of level $L$, $IC^{(L)} = e_L + u_L i G$). Thus the partial decryption result of participant $i$ can be simulated, providing other participants information.

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

$$C_1 = \left( \begin{array}{c} A_1 \\ b_1 \end{array} \right) R_1 + E_1 + u_1 G, \quad C_2 = \left( \begin{array}{c} A_2 \\ b_2 \end{array} \right) R_2 + E_2 + u_2 G,$$
After performing a homomorphic multiplication operation, we obtain:

\[
C_1 G^{-1}(C_2) = \left[ \begin{pmatrix} A_1 \\ b_1 \end{pmatrix} R_1 + E_1 + u_1 G \right] G^{-1}(C_2)
= \left( \begin{pmatrix} A_1 \\ b_1 \end{pmatrix} R_1 G^{-1}(C_2) + E_1 G^{-1}(C_2) + u_1 \begin{pmatrix} A_2 \\ b_2 \end{pmatrix} R_2 + u_1 E_2 + u_1 u_2 G \right)
= \Pi_1 + \delta_1 + u_1 u_2 G
\]

where:

\[
\Pi_1 = \left( \begin{pmatrix} A_1 \\ b_1 \end{pmatrix} R_1 G^{-1}(C_2) + u_1 \begin{pmatrix} A_2 \\ b_2 \end{pmatrix} R_2 \right)
= E_1 G^{-1}(C_2) + u_1 E_2
\]

and \( \Pi_1 = 0, \delta_1 \) is the noise after the first homomorphic evaluation. For the ciphertexts \( C_3, C_4 \) of the same level, we have \( C_3 G^{-1}(C_4) = \Pi_1' + \delta_1' + u_3 u_4 G \), where \( \Pi_1', \delta_1' \) and \( \Pi_1, \delta_1 \) have the same structure.

Let \( C^{(2)}, C^{(2)'} \) be the ciphertext at level 2:

\[
C^{(2)} = C_1 G^{-1}(C_2), \quad C^{(2)'} = C_3 G^{-1}(C_4)
\]

\[
\delta_2 = \delta_1 G^{-1}(C^{(2)'}) + u_1 u_2 \delta_1'
\]

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

\[
C^{(L)} = C^{(L-1)} G^{-1}(C^{(L-1)'}) = \Pi_{L-1} + \delta_{L-1} + u_{L-1} u_{L-1} G
\]

\[
\delta_{L-1} = \delta_{L-2} G^{-1}(C^{(L-1)'}) + u_{L-1} \delta_{L-2}'
\]

To find out what information \( \delta_{L-1} \) contains, first, we observe \( \delta_1 = E_1 G^{-1}(C_2) + u_1 E_2 \).

**Lemma 6** For the DGSW ciphertext \( C_1, C_2 \), let \( C^{(2)} = C_1 G^{-1}(C_2) \), the noise \( \delta_1 \) obtained by decrypting \( C^{(2)} \) is dominated by the noise \( E_1 \) in \( C_1 \):

\[
\delta_1 \approx E_1 G^{-1}(C_2)
\]

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

**Lemma 7** Let \( \epsilon = 2^{-\lambda}, \sigma > \sqrt{2} \sigma \epsilon(Z), m = (kn+W)l, l = \lfloor \log q \rfloor, \{y_i\}_{i \in [ml]} \leftarrow D_{\sigma, \sigma}, y' \leftarrow D_{\sqrt{m} \sigma}, \) we have:

\[
\Delta(\sum_{i=1}^{ml} y_i, y') \leq 8ml \epsilon.
\]
Proof. Let \( \{ y_i^{(1)} \}_{i \in [ml/2]} \) \( \sim \mathcal{D}_{Z, \sqrt{2}\delta} \), by lemma 3:
\[
\Delta(y_1 + y_2, y_1^{(1)}) < 8\epsilon \\
\Delta(y_3 + y_4, y_2^{(1)}) < 8\epsilon \\
\vdots
\]
\[
\Delta(y_{ml-1} + y_{ml}, y_{ml}^{(1)}) < 8\epsilon
\]
By the subadditivity of statistical distances (we proved it in Appendix A) we have:
\[
\Delta(\sum_{i=1}^{ml} y_i, \sum_{i=1}^{ml} y_i^{(1)}) < \frac{ml}{2} \cdot 8\epsilon.
\]
Let \( \{ y_i^{(2)} \}_{i \in [ml/4]} \) \( \sim \mathcal{D}_{Z, 2\delta} \), again by lemma 3:
\[
\Delta(y_1^{(1)} + y_2^{(1)}, y_1^{(2)}) < 8\epsilon
\]
thus:
\[
\Delta(\sum_{i=1}^{ml} y_i^{(1)}, \sum_{i=1}^{ml} y_i^{(2)}) < \frac{ml}{4} \cdot 8\epsilon.
\]
Iterating the above process, we have:
\[
\Delta(\sum_{i=1}^{ml} y_i, y') \leq \frac{ml}{2} \cdot 8\epsilon + \frac{ml}{4} \cdot 8\epsilon + \cdots + 8m\epsilon = 8ml\epsilon.
\]
we complete the proof.

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

Here, we prove Lemma 6:

Proof. First, according to the LWE assumption, replace \( \mathbf{G}^{-1}(\mathbf{C}_2) \) with \( \mathbf{M} \sim U\{0,1\}^{ml \times ml} \). 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 \( \delta_1 \), \( \mathbf{E}_1 \mathbf{M} \) respectively. We have:
\[
\delta_1(1,1) = z_1 e_1 + z_2 e_2 + \cdots + z_{ml} e_{ml} + e_{ml+1} \\
\mathbf{E}_1\mathbf{M}(1,1) = z_1 e_1 + z_2 e_2 + \cdots + z_{ml} e_{ml}
\]
where \( \{ z_i \}_{i \in [ml]} \) is the first column of \( \mathbf{M} \), \( \{ e_i \}_{i \in [ml]} \) \( \sim \mathcal{D}_{Z, \sigma} \) is the first row of \( \mathbf{E}_1 \), \( \mathbf{E}_2(1,1) = e_{ml+1} \) \( \sim \mathcal{D}_{Z, \sigma} \). Suppose, the number of 1s in \( \{ z_i \}_{i \in [ml]} \) is \( r \). By lemma 7 we have:
\[
\Delta(\delta_1(1,1), \mathcal{D}_{Z, \sqrt{r+1}\sigma}) \leq 8(r + 1)\epsilon.
\]
\[
\Delta(\mathbf{E}_1\mathbf{M}(1,1), \mathcal{D}_{Z, \sqrt{r}\sigma}) \leq 8r\epsilon
\]
For our parameter setting, \( 8r \epsilon \leq 8m \ell = \text{poly}(\lambda) \cdot 2^{-\lambda} = \text{negl}(\lambda) \). Thus :

\[
\delta_1(1, 1) \sim D_{\mathbb{Z}, \sqrt{\tau_1}}
\]

\[
E_1M(1, 1) \sim D_{\mathbb{Z}, \sqrt{\tau_2}}
\]

The statistical distance of \( \delta_1(1, 1) \) and \( E_1M(1, 1) \) is :

\[
\Delta(\delta_1(1, 1), E_1M(1, 1)) = \frac{1}{2} \sum_{-\infty}^{+\infty} \left| \frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} - \frac{\rho_{\sqrt{\tau_2}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} \right| = \sum_{-\infty}^{+\infty} \left( \frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} - \frac{\rho_{\sqrt{\tau_2}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} \right)
\]

\[
= 2 \sum_{-\infty}^{+\infty} \left( \frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} - \frac{\rho_{\sqrt{\tau_2}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} \right) < 2 \sum_{-\infty}^{+\infty} \frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})},
\]

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

\[
\frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} = \frac{\rho_{\sqrt{\tau_2}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})}
\]

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

\[
2 \sum_{-\infty}^{+\infty} \frac{\rho_{\sqrt{\tau_1}}(x)}{\rho_{\sqrt{\tau_2}}(\mathbb{Z})} < \frac{2}{C \sqrt{2\pi}} \text{exp} \{ -\frac{C^2}{2} \}
\]

\[
= \frac{2}{C \sqrt{2\pi}} \text{exp} \{ -\frac{1}{2} r(r + 1) \ln \frac{r + 1}{r} \}
\]

\[
= \frac{2}{C \sqrt{2\pi}} \text{exp} \{ -\frac{r + 1}{2} \}
\]

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

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

for \( ml > 4\lambda \). Thus, the statistical distance of \( \delta_1(1, 1) \) and \( E_1M(1, 1) \) :

\[
\Delta(\delta_1(1, 1), E_1M(1, 1)) < \frac{2}{C \sqrt{2\pi}} \text{exp} \{ -\frac{\lambda + 1}{2} \} = \text{negl}(\lambda).
\]

We completed the proof, for other item of \( \delta_1(i, j) \) and \( E_1M(i, j) \) the statement also holds.

\[\blacksquare\]

According to the results we proved above, the noise \( E_2 \) of the right ciphertext \( C_2 \) in the ciphertext \( C_1G^{-1}(C_2) \) is masked by the noise \( E_1 \) in the left ciphertext \( C_1 \). Similarly, the noise \( E_4 \) of \( C_4 \) in \( C_3G^{-1}(C_4) \) is masked by the noise \( E_3 \) of \( C_3 \) on the leftside. For the noise \( \delta_2 = \delta_1G^{-1}(C^{(2')}) + u_1u_2\delta_1' \) of the third level, \( \delta_1' \) is
Fig. 2. Circuit

masked by $\delta_1$, and similarly the noise $\delta_{L-1} = \delta_{L-2}G^{-1}(C^{(L-2)}) + u_{L-2}\delta_{L-2}$ of the $L$-th level, $\delta_{L-2}$ is masked by $\delta_{L-2}$. 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 $\delta_{L-1}$ of the ciphertext $C^{(L)}$ of the $L$-th level only contains the information of noise $E_t (t \in [N])$ in a certain initial ciphertext. At this point, we only need to left-multiply $C^{(L)}$ by a ciphertext $Enc(1)$ whose plaintext is 1, and let $C_{\text{clear}} = Enc(1)G^{-1}(C^{(L)})$. Thus, the noise $\delta_{\text{clear}}$ in $C_{\text{clear}}$ does not contain any information about the noise $\{E_i\}_{i \in [N]}$ in the initial ciphertext $\{C_i\}_{i \in [N]}$. Decrypting $C_{\text{clear}}$, we have:

$$tC_{\text{clear}}G^{-1}(w^T) = t\delta_{\text{clear}}, G^{-1}(w^T) + u_L \left\lfloor \frac{q}{2} \right\rfloor.$$ 

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

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

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

4.6 Bootstrapping

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

However, for those MKFHE schemes that requires ciphertext expansion, additional ciphertext expansion is required, for the reason that $C^{(L)}$ is the ciphertext under $t$, but $\{evk_i\}_{i \in [k]}$ are the ciphertext under $\{t_i\}_{i \in [k]}$. In order to expand $\{evk_i\}_{i \in [k]} \rightarrow \{evk_i\}_{i \in [k]}$, participant $i$ needs to encrypt the random matrix of the ciphertext corresponding to $evk_i$. The extra encryption of $i$ need to done locally is $O(\lambda^6L^6)$.

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

It is regrettable that general polynomial ring $R : \mathbb{Z}[x]/f(x)$ cannot enjoy the leak resilient property of the LHL on the integer ring $\mathbb{Z}$. This means that we cannot transplant the above construction process trivially to RLWE-based FHE. Indeed, [17] pointed out that for $x = (x_1, \ldots, x_l) \in \mathbb{R}$, if the $j$-th NTT coordinate of each $x_i, i \in [l]$ is leaked, then the $j$-th NTT coordinate of $a_{i+1} = \sum_{i=1}^{l} a_i x_i$ is defined, thus $a_{i+1}$ is far from random, although the leakage ratio is only $1/n$. We also notice a trivial solution : for $a, s \in \mathbb{R}_q$, $b = (a, s) \in \mathbb{R}_q$, $b$ leaks information about $s$ at most $n \log q$ bits, therefore, as long as we set $l$ long enough, for example, $l = l + n \log q$, then obviously $b$ is close to uniformly random, but this will result in a extremely large key, thus it is not practical.

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

**Claim 4** For a given $\delta \in \{0,1\}^\lambda$, $k$ probabilistic polynomial time(ppt) adversary $\mathcal{A}$, Each $\mathcal{A}$ makes $q_{\text{ro}} = \text{poly}(\lambda)$ queries to $H$, let $\overline{\text{Adv}}$ denotes the probability of finding a collision, then: $\overline{\text{Adv}} = \text{negl}(\lambda)$

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

**Key generation with bit commitment.**

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

1. $\text{pp} \leftarrow \text{Setup}(1^\lambda, 1^L)$: Input security parameter $\lambda$, circuit depth $L$, output $\text{pp} = (d, q, \chi, B_\chi)$, which $\chi$ is an noise distribution over $R : \mathbb{Z}[x]/x^d + 1$, satisfying $e \leftarrow \chi$, $|e|_\infty$ is bounded by $B_\chi$, and RLWE$_{d,q,\chi,B_\chi}$ is infeasible.

2. $i$ generates $\Phi_i = \{a_i, d_i, e_i\}$ where $a_i \leftarrow U(R_q)$ is used for public key, $d_i, f_i \leftarrow U(R_q^2)$ for evaluation key, and commitment $\Psi_i = \{\delta_i, \epsilon_i, \zeta_i\}, \delta_i = H(a_i), \epsilon_i = H(d_i), \zeta_i = H(f_i)$, broadcast $\Psi_i$.

3. After all $\{\Psi_i\}_{i \in [k]}$ are public, $i$ discloses $\Phi_i$.

4. After receiving $\{\Phi_j\}_{j \in [k]/i}$, $i$ broadcast $\{b_i, h_i\}$, where $b_i = as_i + e_1, h_i = ds_i + e_2, a = \sum_{i=1}^k a_i, d = \sum_{i=1}^k d_i, (s_i, e_1, e_2) \leftarrow \chi^{l+2}$.

5. After receiving $\{b_j, h_j\}_{j \in [k]/i}$, $i$ output $\text{pk}_i = (a, b)$ and evaluation key $\text{evk}_i = (h_i, \eta_i, \theta_i)$

$$b = \sum_{i=1}^k b_i \quad \quad \quad \quad \eta_i = dr_i + e_3 + s_ig$$

$$\theta_i = fs_i + e_4 + r_ig \quad \quad \quad (r_i, e_3, e_4) \leftarrow \chi^{2l+1}$$

**Re-linearization ciphertext**

Multiplying two ciphertext $c_1, c_2 \in R_q^2$, which under the same private key $t = (1, s), \ s = \sum_{i=1}^k s_i$, resulting $c_{\text{mult}} = c_1 \otimes c_2 \in R_q^2$, where its corresponding private key is $t \otimes t = (1, s, s^2)$. In order to re-linearize $c_{\text{mult}}$, we need to construct the ciphertext of $s^2$ under $t$. Let total evaluation key $\Pi = (\eta, \theta, h)$.

$$\eta = \sum_{i=1}^k \eta_i \quad \theta = \sum_{i=1}^k \theta_i \quad h = \sum_{i=1}^k h_i$$
Let \( k = (k_0, k_1) \), \( k_0 = -\theta g^{-1}(h) \in R_q \), \( k_1 = (\eta + f g^{-1}(h)) \in R_q \), obviously \( k_0 + k_1 s \approx s^2 g \) (omit small error). Let \( c_{\text{mult}} = (c_0, c_1, c_2, c_3) \).

\[
(c_{\text{mult}}, t \otimes t) = c_0 + (c_1 + c_2)s + s^2 c_3 \\
= c_0 + (c_1 + c_2)s + s^2 g g^{-1}(c_3) \\
= c_0 + k_0 g^{-1}(c_3) + (c_1 + c_2 + k_1 g^{-1}(c_3))s.
\]

Let \( c_{\text{linear}} = (c'_0, c'_1) \), \( c'_0 = c_0 + k_0 g^{-1}(c_3) \), \( c'_1 = c_1 + c_2 + k_1 g^{-1}(c_3) \), output \( c_{\text{linear}} \) as re-linearized ciphertext. The algorithm defines as follows:

- \( c_{\text{linear}} \leftarrow \text{Relinear}(c_{\text{mult}}, \{\text{evk}_i\}_{i \in [k]}) \): Input \( c_{\text{mult}} \in R_q^d \), evaluation key \( \{\text{evk}_i\}_{i \in [k]} \), perform the following algorithm, output \( c_{\text{linear}} = (c'_0, c'_1) \).

---

Ciphertext Relinearization

**Input:** \( c_{\text{mult}} = (c_0, c_1, c_2, c_3) \in R_q^d \), \( \{\text{evk}_i\}_{i \in [k]} = \{h_i, \eta_i, \theta_i\}_{i \in [k]} \)

**Output:** \( c_{\text{linear}} = (c'_0, c'_1) \in R_q^2 \)

1: \( \eta \leftarrow \sum_{i=1}^{k} \eta_i \), \( \theta \leftarrow \sum_{i=1}^{k} \theta_i \), \( h \leftarrow \sum_{i=1}^{k} h_i \)
2: \( k_0 \leftarrow -\theta g^{-1}(h) \), \( k_1 \leftarrow \eta + f g^{-1}(h) \)
3: \( c'_0 \leftarrow c_0 + k_0 g^{-1}(c_3) \), \( c'_1 \leftarrow c_1 + c_2 + k_1 g^{-1}(c_3) \)
4: **Output:** \( (c'_0, c'_1) \)
5: **End.**

---

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

### 6 Conclusions

For the LWE-based MKFHE in order to alleviate the overhead of the local participants, we proposed the concept of KL-MKFHE which introduced a Key lifting procedure, getting rid of expensive ciphertext expansion operation and construct a DGWSW style KL-MKFHE under plain model. Our Scheme #1 is more friendly to local participants than previous scheme, for which the local encryption \( O(N \lambda^6 L^4) \) is reduced to \( O(N) \), and by abandoning noise flooding, it compress \( q \) from \( 2^{O(L)} B_X \) to \( 2^{O(L)} B_X \), reducing the computational scale of the entire scheme. However, the key length depends on the number of participants and the amount of leakage, which limits the application of the scheme to some extent. Further work will focus on compressing the key length.

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

References


Appendix

A the additivity of statistical distances

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

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

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

\[
\leq \frac{1}{2} \sum_{k \in E} \left( \left| \Pr(X = k) - \Pr(Y = k) \right| + \left| \Pr(Y = k) - \Pr(Z = k) \right| \right)
\]

\[
\leq \Delta(X, Y) + \Delta(Y, Z).
\]

\[\square\]

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

\[
\Delta(X + Y, Y + Z) \leq \Delta(X, Z)
\]

Proof.

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

\[
= \frac{1}{2} \sum_{k \in E} \left| \Pr(X = k - Y) - \Pr(Z = k - Y) \right|
\]

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

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

\[
\leq \frac{1}{2} \sum_{k \in E} \sum_{b \in E} \left| \Pr(Y = b) (\Pr(X = k - b) - \Pr(Z = k - b)) \right|
\]

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

\[
\leq \sum_{b \in E} \Pr(Y = b) \cdot \Delta(X, Z)
\]

\[
= \Delta(X, Z)
\]

\[\square\]

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

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

Proof. by Claim 5, We have :

\[
\Delta(X + Y, Z + W) \leq \Delta(X + Y, Z + Y) + \Delta(Z + Y, Z + W)
\]
then, by Claim 6, We have:

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

\[ \square \]

B The proof of DGSW leakage-resilient in [11], and our improved method.

For a given DGSW ciphertext:

\[ C = \left( \begin{array}{c} A \\ b \end{array} \right) R + \left( \begin{array}{c} E_0 \\ e_1 \end{array} \right) \]

Let \( C_0 = A R + E_0, \ e_1 = b R + e_1. \) Because \( b = s A, \) thus \( C \) can be rewritten as:

\[ C = \left( \begin{array}{c} C_0 \\ s C_0 + e_1 - s E_0 \end{array} \right) \quad (5) \]

The proof in [11] required \( s E_0/e_1 = \text{negl}(\lambda), \) thus \( C \approx \text{stat} \left( \begin{array}{c} C_0 \\ s C_0 + e_1 \end{array} \right) \). Using the leftover hash lemma with \( C_0 \) as seed and \( s \) as source, they had that \((C_0, s C_0)\) were jointly statistically indistinguishable from uniform, which Lemma 5 followed.

**Our method:** Below we show that \( s E_0/e_1 = \text{negl}(\lambda) \) 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 (5). For random \( R \in \mathbb{Z}_q^{n \times ml} \), without loss of generality, assuming \( \frac{ml}{n} = g \), we can divide \( R \) into \( g \) square matrices:

\[ R = (R_1, R_2, \cdots, R_g) \]

where \( R_i \in \mathbb{Z}_q^{n \times n} \). Similarly, for \( E_0 \in \mathbb{Z}_q^{(m-1) \times ml}, \ e_1 \in \mathbb{Z}_q^{ml} : \)

\[ E_0 = (E_{0,1}, E_{0,2}, \cdots, E_{0,g}) \]
\[ e_1 = (e_{1,1}, e_{1,2}, \cdots, e_{1,g}) \]

where \( E_{0,i} \in \mathbb{Z}_q^{(m-1) \times n}, \ e_{1,i} \in \mathbb{Z}_n \). Let \( \{v_i \in \mathbb{Z}_q^n \}_{i \in [g]} \) be the solution of equation:

\[ \{ v_i, R_i = s E_{0,i} \}_{i \in [g]} \]

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

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

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

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

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

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

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

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

Define set $V$:

$$V = \{ 0^{1 \times ml}, (v_1, v_2, \cdots, v_g) \}$$

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

$$\{ \mathbf{d} \leftarrow \mathcal{D}(V) : \Pr(\mathbf{d} = 0^{1 \times ml}) = p \Pr(\mathbf{d} = (v_1, v_2, \cdots, v_g)) = 1 - p \}$$

Let $\Phi'$ 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

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

$$X'_1 = \{ x : (\mathbf{A}', \mathbf{b}', \mathbf{d}) \leftarrow \Phi', \ x = (\mathbf{A}', \mathbf{b}', \mathbf{d}, \mathbf{U}) \leftarrow \mathcal{D}_1(\mathbf{A}', \mathbf{b}', \mathbf{d}) \}. $$

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

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

it is a sample of $X'_0$:

$$\begin{pmatrix} \mathbf{A}' \mathbf{R}' + \mathbf{E}'_0 \\ (\mathbf{b}' + \mathbf{d}_1) \mathbf{R}'_1 + \mathbf{e}'_1, \cdots, (\mathbf{b}' + \mathbf{d}_g) \mathbf{R}'_g + \mathbf{e}'_g \end{pmatrix}$$
with \( A' = A, b' = b, R' = R, E'_0 = E_0, (e'_1, e'_2, \ldots, e'_g) = (e_{1,1}, e_{1,2}, \ldots, e_{1,g}), \)
\[ d = (v_1, v_2, \ldots, v_g). \]

The process that follows is the same as we showed in Section 4.4. By Theorem 2, if there is an adversary who can distinguish the DGSW ciphertext with uniform distribution (Problem \( P \)) that leaks part of private key, then he can distinguish \((C_0, sC_0 + e_1)\) with uniform distribution which is jointly statistically indistinguishable by leftover hash lemma.