Multi-Key Fully Homomorphic Encryption: removing noise flooding in distributed decryption via the smudging lemma on discrete Gaussian distribution

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

[1] Chongqing Key Laboratory of Automated Reasoning and Cognition, Chongqing Institute of Green and Intelligent Technology of Chinese Academy of Sciences, Chongqing, China
[2] Chongqing School, University of Chinese Academy of Sciences, Chongqing, China
daixiaokang@cigit.ac.cn wuwenyuan@cigit.ac.cn yongfeng@cigit.ac.cn
Corresponding author: Wenyuan Wu

Abstract. The current Multi-key Fully Homomorphic Encryption (MKFHE) needs to add exponential noise in the distributed decryption phase to ensure the simulatability of partial decryption. Such a large noise causes the ciphertext modulus of the scheme to increase exponentially compared to the Single-key Fully Homomorphic Encryption (FHE), further reducing the efficiency of the scheme and making the hardness problem on the lattice on which the scheme relies have a sub-exponential approximation factor $\widetilde{O}(n \cdot 2^{\sqrt{nL}})$ (which means that the security of the scheme is reduced). To address this problem, this paper analyzes in detail the noise in partial decryption of the MKFHE based on the LWE problem. It points out that this part of the noise is composed of private key and the noise in initial ciphertext. Therefore, as long as the encryption scheme is leak-resistant and the noise in partial decryption is independent of the noise in the initial ciphertext, the semantic security of the ciphertext can be guaranteed. In order to make the noise in the initial ciphertext independent of the noise in the partial decryption, this paper proves the smudging lemma on discrete Gaussian distribution and achieves this goal by multiplying the initial ciphertext by a "dummy" ciphertext with a plaintext of 1. Based on the above method, this paper removes the exponential noise in the distributed decryption phase for the first time and reduces the ciphertext modulus of MKFHE from $2^{\omega(\lambda L \log \lambda)}$ to $2^{O(\lambda+L)}$ as the same level as the FHE.

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

1 Introduction

To address the privacy concerns of multiple data providers, López-Alt et al. [16] introduced the concept of MKFHE and developed the first MKFHE scheme based on the modified-NTRU problem. Conceptually, it enhances the functionality of FHE by allowing data providers to do encryption independently from other parties and the key generation and data encryption are done locally. To obtain the evaluated result, all parties are required to execute a round of threshold decryption protocol.

1.1 Motivation

A series of works [3,8,17] have shown that MKFHE is an excellent base tool for building round optimal MPC. Although MKFHE is conceptually appealing, its ciphertext modulus $q = 2^{\omega(\lambda L \log \lambda)}$ (Note: λ is the security parameter, L is the circuit depth) as in the schemes [3, 9, 10, 17, 20] is exponentially larger than the ciphertext modulus $q = 2^{O(L)}$ of FHE. Such a large ciphertext modulus leads to inefficiencies in the scheme and makes the hardness problems on the lattice that the scheme relies on easier to solve (which means the security of the scheme is reduced). The details are as follows.

Noise flooding technology results in a large modulus q. Unlike the FHE, the decryption of MKFHE is a distributed process: after the homomorphic evaluation is completed, each participant needs to use their own private key to "partially decrypt" the ciphertext, and then make the "partially decrypted" result public. After all participants make the results public, these results are summed up to finally get the evaluation result of plaintext. As far as we know, whether it is MKFHE or Threshold Fully Homomorphic Encryption (Th-FHE), such as [3, 8–10, 17, 20], in order to ensure the simulatability of the "partially decrypted" result, it is necessary to add exponential noise to it. This technology is generally called noise flooding. For example, let the noise accumulated after homomorphic evaluation is $\mathbf{e}$, the private key of participant i is $\mathbf{s}_i$, to simulate the partial decryption result p_i of participant i, the noise e added to p_i must satisfy $\langle \mathbf{e}, \mathbf{s}_i \rangle / e = \mathsf{negl}(\lambda)$ (Note: $\mathsf{negl}(\lambda)$ is a negligible function with respect to λ). To ensure the correctness of the decryption result, the modulus q needs to satisfy $q > 4e$. Thus the flooding noise e results in a q that is exponentially larger than the q in single-key FHE. Typically in [17], the flooding noise $e = 2^{O(\lambda L \log \lambda)} B_\chi$ (defined as $e^{sm} = 2^{O(d\lambda \log \lambda)} B_\chi$ in [17, page.755], d denote the multiplicative depth, which we denote as L in our paper), and the ciphertext modulus $q = 2^{\omega(\lambda L \log \lambda)} B_\chi$ (in [17, page.755]). Such a large ciphertext modulus q severely affects the efficiency of the scheme, effectively ruling out parameter choices with dimension $n < 32768$. For example, under the parameter set $(n = 32768, q = 2^{880})$ recommended by the homomorphic encryption standardization document [1] targeting 128-bit LWE security, even a single homomorphic multiplication is not feasible for $q = 2^{\omega(\lambda L \log \lambda)} B_\chi$, since $\lambda L \log \lambda = 896 > 880$. In a larger parameter setting, such as $(n = 65536, q = 2^{1792})$, two levels of homomorphic multiplication are barely achievable with the 124-bit LWE securtiy (estimated by Lattice Estimator). In contrast, for the same parameter set $(n = 65536, q = 2^{1792})$, the original GSW scheme can support up to approximately 59 levels of homomorphic multiplication (see [14, page.87]. The decryption requires $q/B > 8(N+1)^L$ where $N = (n+1)\log q$, in our test, we set $B = 6\sigma$ with $\sigma = 3.2$). Notably, works such as [8–10, 17, 20] and recent LWE-based MKFHE scheme [3] adopt the *ciphertext expansion* and *noise flooding* techniques introduced in [17]. As a result, they require ciphertext moduli of the same scale as in [17], rendering these MKFHE schemes largely impractical in terms of efficiency. Furthermore, a large ciphertext modulus q implies that the associated GapSVP_γ problem underlying the hardness of LWE must be solved with a large approximation factor $\gamma = \widetilde{O}(n \cdot 2^{\lambda L \log \lambda})$ thereby weakening the underlying lattice assumption.

1.2 Our method

In order to remove the flooding noise introduced in the distributed decryption phase, we studied the result of partial decryption of participant i. This result can be simply expressed as

$$p_i = \langle \mathbf{s}_i, \mathbf{e} \rangle + m \mod q,$$

where $\mathbf{s}_i$ is the private key of participant i, $\mathbf{e}$ is the noise after homomorphic evaluation, m is the evaluation result, and q is the ciphertext modulus. The actual equation of p_i is more complicated and contains some other items, in order to briefly explain the general idea, we ignore it here. (For the specific equation of p_i, please refer to Section 4.4). Assuming there are k participants, in order to obtain the homomorphic evaluation result m, the partial decryption $\{p_i\}_{i\in[k]}$ of all participants needs to be made public. However p_i contains the inner product $\langle \mathbf{s}_i, \mathbf{e} \rangle$ of the private key $\mathbf{s}_i$ and noise $\mathbf{e}$. If p_i is made public directly, $\langle \mathbf{s}_i, \mathbf{e} \rangle$ will be leaked, and the security of the ciphertext cannot be proved at this time. Therefore, the current approach as used in [8–10, 17] is to add an flooding noise e to p_i to mask $\langle \mathbf{s}_i, \mathbf{e} \rangle$.

$$p_i = e + \langle \mathbf{s}_i, \mathbf{e} \rangle + m \mod q.$$

As we mentioned in the Motivation, this will result in a large modulus q.

Our Observations. We note that if the encryption is leak-resistant, then even if $\langle \mathbf{s}_i, \mathbf{e} \rangle$ leaks part of $\mathbf{s}_i$, as long as the entropy of $\mathbf{s}_i$ is large enough, the security of the ciphertext can be proven. This can be guaranteed by leakage resilient cryptography, such as the Lemma 1 states in [8]. Now there is only $\mathbf{e}$ left to be handle. Note that $\mathbf{e}$ is the noise accumulated after the homomorphic evaluation, which is determined by the noise in the initial ciphertext. Therefore, leaking $\mathbf{e}$ may leak the noise in the initial ciphertext, and the security cannot be guaranteed at this time. We observed that the ciphertext of the encryption scheme based on the LWE problem has an asymmetric property: that is, when two ciphertexts are multiplied, the noise of the ciphertext on the left will mask the noise of the ciphertext on the right. That is to say, the ciphertext obtained by multiplying two ciphertexts contains almost no noise from the ciphertext on the right, but only the noise from the ciphertext on the left. For example, let $\mathbf{C}_1$ and $\mathbf{C}_2$ be the two ciphertexts of the dual GSW scheme [8] and $\mathbf{E}_1$ and $\mathbf{E}_2$ be the noise of $\mathbf{C}_1$ and $\mathbf{C}_2$ respectively. Let $\mathbf{C}_{\mathsf{mult}} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$ be the ciphertext obtained by multiplying $\mathbf{C}_1$ and $\mathbf{C}_2$, where $\mathbf{G}^{-1}(\mathbf{C}_2)$ represents the bit decomposition of $\mathbf{C}_2$. The noise in $\mathbf{C}_{\mathsf{mult}}$ is $\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{E}_2$, actually we have

$$\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{E}_2 \approx_s \mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2), \tag{1}$$

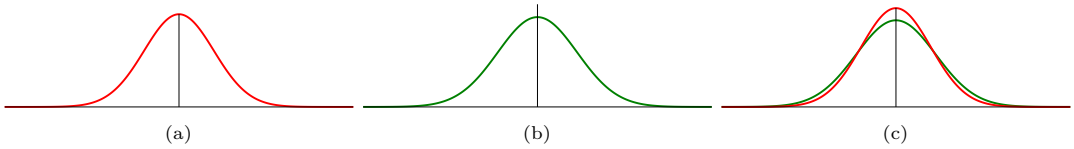
where $\approx_s$ denotes that the left and right sides of the equation are statistically indistinguishable. That is to say, $\mathbf{E}_2$ is "drown out" by $\mathbf{E}_1\mathbf{G}^{-1}(\mathbf{C}_2)$. Therefore, the ciphertext $\mathbf{C}_{\mathsf{mult}}$ is independent of the noise $\mathbf{E}_2$ in the ciphertext $\mathbf{C}_2$. Based on the above observations, we can preprocess the initial ciphertext before homomorphic evaluation: we can left-multiply the initial ciphertext by a "dummy" ciphertext (plaintext is 1), so that the ciphertext after multiplication is independent of the noise in the initial ciphertext. Then using the ciphertext as input and performing homomorphic evaluation, the noise $\mathbf{e}$ in the result of the homomorphic evaluation is also independent of the initial ciphertext. Combined with the anti-leakage property of the encryption scheme, the simulatability of distributed decryption can be guaranteed even without adding flooding noise. For a detailed discussion, please refer to Section 4.4. To rigorously prove this asymmetric property of ciphertext multiplication, that is, to prove Equation (1), we need to prove the smudging lemma over discrete Gaussian distribution.

Smudging lemma over discrete Gaussian. Before going into a detailed technical description, we first give a general idea so that we can have an intuitive understanding. The discrete Gaussian version of the smudging lemma is obtained from the observation of the continuous Gaussian distribution: when n is large enough, the sum of n independent and identically distributed (iid) Gaussian distributions is almost the same as the sum of $n+1$ idd Gaussian distributions. Let X, Y be Gaussian distributions with variance $n\sigma^2$ and $(n+1)\sigma^2$ in $\mathbb{R}$ respectively, with probability density function $f(x)$ and $g(x)$ as shown in Figure 1.

$$f(x) = \frac{1}{\sqrt{n}\sigma} e^{-\frac{\pi x^2}{n\sigma^2}}, \qquad g(x) = \frac{1}{\sqrt{n+1}\sigma} e^{-\frac{\pi x^2}{(n+1)\sigma^2}}.$$

Let the intersection point of $f(x)$ and $g(x)$ be t. It's easy to see that t is greater than $\sqrt{\frac{n}{2\pi}}\sigma$. The statistical distance between X and Y is

$$\Delta(X,Y) = \int_{x>t} g(x) - f(x)\,\mathrm{d}x < \int_{x>t} g(x)\,\mathrm{d}x < \int_{x>\sqrt{\frac{n}{2\pi}}\sigma} g(x)\,\mathrm{d}x = \mathsf{negl}(n).$$



(a) (b) (c)

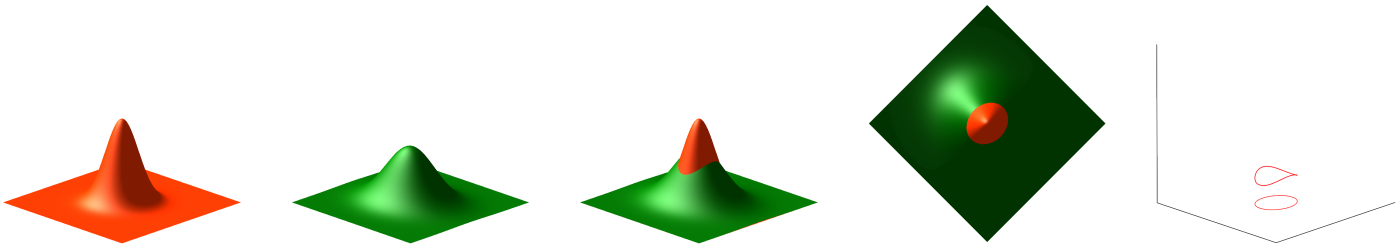
Fig. 1: Probability density function of one-dimensional Gaussian distribution. (a) shows $f(x)$, (b) shows $g(x)$ and (c) is the intersection of $f(x)$ and $g(x)$.

That is to say, if the noise e is Gaussian with variance σ^2, we only need to sample e' from a Gaussian distribution with variance $n\sigma^2$. Then $e + e' \approx_s e'$, and $||e/e'|| = O(n^{-1})$, while for the general smudging lemma 1 must satisfy $||e/e'|| = \mathsf{negl}(n)$.

The one-dimensional case is relatively simple. Now consider the two-dimensional case. Let $\Sigma_1, \Sigma_2 = \Sigma_1 + \sigma^2\mathbf{I}$ be two symmetric positive definite matrices on $\mathbb{R}^{2\times2}$. The probability density functions $f(\mathbf{x})$ and $g(\mathbf{x})$ of two-dimensional random variables X and Y on $\mathbb{R}^{2\times2}$ respectively are

$$f(\mathbf{x}) = \frac{1}{\sqrt{\det(\Sigma_1)}} e^{-\pi \mathbf{x}\Sigma_1^{-1}\mathbf{x}^T}, \qquad g(\mathbf{x}) = \frac{1}{\sqrt{\det(\Sigma_2)}} e^{-\pi \mathbf{x}\Sigma_2^{-1}\mathbf{x}^T},$$

as shown in Figure 2. At this time, the intersection of $f(\mathbf{x})$ and $g(\mathbf{x})$ is a space curve, as shown in the



(a) (b) (c) (d) (e)

Fig. 2: Probability density function of two-dimensional Gaussian distribution. (a) is the probability density function of $f(\mathbf{x})$, (b) is the probability density function of $g(\mathbf{x})$, (c) is the intersection of $f(\mathbf{x})$ and $g(\mathbf{x})$, (d) is the top view of intersection, (e) is the intersection and the projection to xy plane.

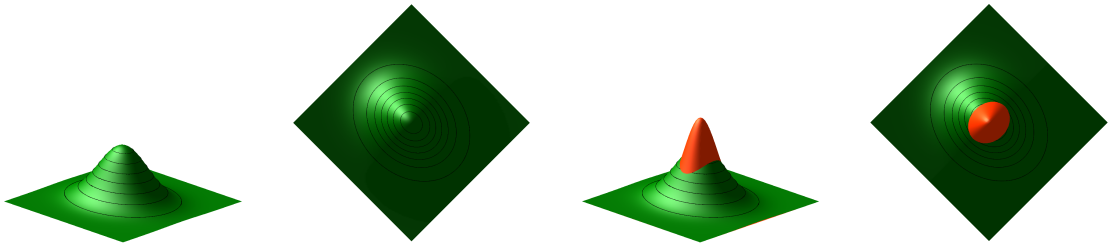
subfigure (c) of Figure 2. Projecting the space curve onto the xy plane, it is an ellipse, as shown in the subfigure (e) of Figure 2 which is $\mathcal{E}_{ints}$

$$\mathcal{E}_{ints} : \frac{1}{\pi} \ln\left(\frac{\det(\Sigma_1)}{\det(\Sigma_2)}\right) = \mathbf{x}(\Sigma_2^{-1} - \Sigma_1^{-1})\mathbf{x}^T.$$

Then the statistical distance between X and Y is

$$\Delta(X,Y) = \int_{\mathbb{R}^2\setminus\mathcal{E}_{ints}} g(\mathbf{x}) - f(\mathbf{x})\,\mathrm{d}\mathbf{x} \le \int_{\mathbb{R}^2\setminus\mathcal{E}_{ints}} g(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{2}$$

The upper bound on the right side of Equation (2) is not easy to find, because the integral region and the integral function are inconsistent. The integral region is determined by the ellipse $\mathcal{E}_{ints}$ whose "shape" is $\Sigma_2^{-1} - \Sigma_1^{-1}$, while the integral function $g(\mathbf{x})$ has the "shape" Σ_2. The isoprobability lines of $g(\mathbf{x})$ is shown in the subfigure (a) in Figure 3 which has the "shape" Σ_2.



(a) (b) (c) (d)

Fig. 3: The isoprobability lines and intersection plots. (a) is the isoprobability lines of $g(\mathbf{x})$. (b) is the top view of isoprobability lines of $g(\mathbf{x})$, it is shaped like some ellipses of Σ_2. (c) is the isoprobability lines of $g(\mathbf{x})$ and the intersection in one picture. (d) is the top view of (c).

For the integral of the area enclosed by the isoprobability line, there is a closed analytical expression that can be applied, which is generally called the tail probability of the Gaussian distribution [5]

$$\Pr[\mathbf{x}\Sigma_2^{-1}\mathbf{x} \geq \chi_2^2(\alpha)] = \int_{\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T \geq \chi_2^2(\alpha)} g(\mathbf{x})\,\mathrm{d}\mathbf{x} < 1-\alpha, \tag{3}$$

where $\chi_2^2(\alpha)$ is the quantile function of the chi-square distribution with 2 degrees of freedom and α as the probability [22]. Equation (3) means that for the two-dimensional Gaussian random variable Y (whose probability density function is $g(\mathbf{x})$), the probability that it falls outside the Σ_2 ellipse with radius $\chi_2^2(\alpha)$ is less than $1-\alpha$. Note that the upper bound of the statistical distance between X and Y requires integrating $g(\mathbf{x})$ outside the ellipse $\mathcal{E}_{ints}$. The tail probability of the Gaussian distribution as shown in Equation (3) support integrating $g(\mathbf{x})$ outside a region of the ellipse Σ_2. Put the isoprobability lines and intersection curve in one picture, as shown in the subfigure (c) of Figure 3. Projecting the subfigure (c) to the xy plane, we get Figure 4. The red ellipse is $\mathcal{E}_{ints}$ which is the

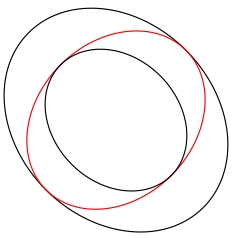


Fig. 4: Projection of isoprobability lines and intersection curve on the xy plane. The red ellipse is the projection of the intersection curve, and the black ellipses are the projection of the isoprobability lines.

projection of the intersection curve to the xy plane. The two black ellipses are the projections of the isoprobability lines on the xy plane. The larger one is just tangent to $\mathcal{E}_{ints}$, and the smaller one is just inscribed to $\mathcal{E}_{ints}$. Since we only need to find the upper bound of the statistical distance $\Delta(X,Y)$, we can find an ellipse $\mathcal{E}_{insc}$ with the "shape" of Σ_2 inscribed in the ellipse $\mathcal{E}_{ints}$, as the smaller black ellipse shown in Figure 4. At this time, we have the statistical distance

$$\Delta(X,Y) = \int_{\mathbb{R}^2\setminus\mathcal{E}_{ints}} g(\mathbf{x}) - f(\mathbf{x})\,\mathrm{d}\mathbf{x} \leq \int_{\mathbb{R}^2\setminus\mathcal{E}_{ints}} g(\mathbf{x})\,\mathrm{d}\mathbf{x} \leq \int_{\mathbb{R}^2\setminus\mathcal{E}_{insc}} g(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Let the ellipse $\mathcal{E}_{insc}$ be

$$\mathcal{E}_{insc} : \mathbf{x}\Sigma_2^{-1}\mathbf{x}^T = k,$$

where $k \in \mathbb{R}$ is the radius to be determined. Then $\mathcal{E}_{insc}$ is exactly the smaller black ellipse in Figure 4. At this time, the radius k to be determined satisfies $k\lambda_1 = \lambda_2$, where λ_1 is the maximum eigenvalue of Σ_2, and λ_2 is the minimum eigenvalue of $\Sigma_2 - \Sigma_1$. Further, according to the result of Equation (3), the upper bound of the statistical distance can be determined.

Extending the above result to a multi-dimensional discrete Gaussian random variable requires extending Banaszczyk's spherical theorem to the ellipsoid. Different from the integral of Gaussian function, the discrete Gaussian summation on $\mathbb{Z}^n$ is not easy. As a compromise, we use continuous Gaussian integrals instead. The main idea is still the same, first find the intersection equation, which forms an ellipsoid in this case, and then the statistical distance. Informally, we have the following result for discrete Gaussian random variables. See Section 3. for more formal result.

Lemma (Informal)**.** *Let n be an integer, $\mathcal{D}_{\mathbb{Z}^n,\sigma}$ be the discrete Gaussian distribution with variance σ^2 on $\mathbb{Z}^n$, U be the uniform distribution over $\{0,1\}^{n\times n}$. Let $\mathbf{e}_1, \mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}$, $\mathbf{M} \leftarrow U$, the distribution of $\mathbf{e}_1\mathbf{M}$ be X and the distribution of $\mathbf{e}_1\mathbf{M} + \mathbf{e}_2$ be Y. It holds that*

$$\Delta(X, Y) < 2^{-n}.$$

1.3 Our Contributions

As we elaborated in the Motivation section, the use of noise flooding techniques in multi-key homomorphic encryption schemes leads to a ciphertext modulus that is exponentially larger than that of traditional single-key schemes, significantly reducing efficiency. To address this, we prove a smudging lemma over the discrete Gaussian distribution and reveal an important asymmetry in the multiplication of ciphertexts under the DGSW encryption scheme–specifically, when two ciphertexts are multiplied, the noise from the right ciphertext is masked by the left one. This key observation enables us to eliminate the need for noise flooding in DGSW-based multi-key homomorphic encryption, thereby reducing the ciphertext modulus to the same order of magnitude as that in conventional single-key schemes.

We removed the flooding noise introduced in the distributed decryption phase and for the first time reduced the ciphertext modulus of the MKFHE from $q = 2^{\omega(\lambda L \log \lambda)}$ to $q = 2^{O(\lambda+L)}$. Accordingly, the approximation factor γ of the GapSVP_γ problem is reduced from $\widetilde{O}(n \cdot 2^{\lambda L \log \lambda})$ to $\widetilde{O}(n \cdot 2^{\lambda+L})$. In addition, we constructed a key conversion method in the initialization phase of the scheme. The ciphertext generated by the converted key directly supports homomorphic evaluations, thereby removing the complex *ciphertext expansion* operations in above schemes. Based on the above results, we constructed a MKFHE based on the LWE problem under the plain model. We give a comparison with schemes [3, 8, 18, 20] in Table 1.

Table 1: Scheme complexity comparison

Scheme	Module q	Interaction	Model
MKFHE [20]	$2^{O(\lambda L)}B_\chi$	–	CRS
MKFHE [8]	$2^{O(\lambda L)}B_\chi$	2 rounds	–
MKFHE [3]	$2^{O(\lambda L)}B_\chi$	2 rounds	CRS
Th-FHE [18]	$\mathsf{poly}(\lambda)B_\chi$	1 rounds	Trusted Third Party
Our scheme	$2^{O(\lambda+L)}B_\chi$	2 rounds	–

The "Module q" column denotes the module base; λ denotes the security parameter, B_χ denotes the initial LWE noise, L denotes the depth of the circuit. The "Interaction" column denotes the round of interaction introduced in the initialization phase. The "Model" column indicates the computational model adopted. CRS stands for "Common Random String", which is generated by a trusted third party and distributed to all participants for use during key initialization..

1.4 Related Work

Several lines of research have sought to improve the efficiency and scalability of MKFHE. Notably, the following works are closely related to the goals of this paper. In the work [23], the authors address the performance bottleneck in existing NTRU-based MKFHE schemes, where a relinearisation step is performed immediately after every homomorphic multiplication–leading to significant computational overhead and severely reduced efficiency. To mitigate this, they propose a strategy that defers relinearisation by combining two multiplication operations before applying a single relinearisation. This approach effectively reduces the number of relinearisation operations by half and significantly improves the overall efficiency of homomorphic evaluation. Additionally, by constructing the NTRU-FHE system over prime cyclotomic rings, which have fewer subfields, the scheme gains stronger resistance against subfield attacks. However, the proposed method is tailored specifically to NTRU-type structures and does not directly generalize to GSW or BGV-type MKFHE schemes. In [13], the authors address a key limitation of traditional homomorphic authenticators (HAs), which only support homomorphic computations over data authenticated by a single user and cannot scale to multi-source settings. This work is the first to formally introduce and define the concept of *multi-key homomorphic authenticators (Multi-key HAs)*. Based on lattice assumptions, the authors construct a Multi-key HA scheme that supports homomorphic evaluation of circuits with polynomial-depth evaluation. The proposed scheme enables computations over data authenticated under different keys, while preserving *succinctness*–the size of the resulting authenticator depends only on the number of users involved, and grows logarithmically with the size of the input data.

1.5 Related work of MKFHE based on LWE

As suggested, we have re-investigated recent multi-key fully homomorphic encryption (MKFHE) schemes from the past three years, restricting our survey to those based on the Learning with Errors (LWE) assumption to ensure a fair comparison. The work most closely related to ours is *Maliciously Circuit-Private Multi-Key FHE and MPC Based on LWE [3]*, as both approaches are built upon [8] framework. However, the key difference lies in the focus: their work aims to construct an MKFHE scheme secure against malicious adversaries, whereas our emphasis is different. Methodologically, we exploit the asymmetry and leakage resilience properties of DGSW ciphertext multiplication, which allows us to avoid adding exponential-sized noise during distributed decryption. In contrast, their scheme still relies on the noise flooding technique from [17], resulting in ciphertext module of size $2^{\omega(\lambda L \log \lambda)}$.

The work *Low Communication Threshold Fully Homomorphic Encryption [18]* shares the same goal as ours—minimizing the noise introduced during distributed decryption. However, the approach it adopts is fundamentally different from ours. Their method involves introducing a trusted third party to apply a standard noise flooding technique to the ciphertext after homomorphic evaluation, followed by ciphertext compression. As a result, the modulus of the compressed ciphertext remains comparable to that of standard FHE schemes, namely $q = \mathsf{poly}(\lambda)$. Since the noise is compressed along with the ciphertext, only a $\mathsf{poly}(\lambda)$-sized noise needs to be added during partial decryption. A critical limitation of this approach is that the noise flooding must be performed by a trusted third party; otherwise, the noise distribution after compression may deviate from the desired discrete Gaussian. We have included a comparison of these two schemes in Table 1. In addition, we have also reviewed literature with methodologies similar to ours; please refer to the following subsection for details.

1.6 Related work on masking noise in ciphertext

Regarding the topic of "the use of the multiplicative properties of GSW ciphertexts to mask original ciphertext noise" after conducting an in-depth literature review, we have not found any relevant

literature. However, in terms of the objective of masking noise in ciphertexts, we have identified several papers with similar approaches, such as *Fhe circuit privacy almost for free [6]*, *Sanitization of fhe ciphertexts. [12]* and *Circuit privacy for FHEW/TFHE-style fully homomorphic encryption in practice [15]*. The work [6] is on achieving circuit privacy in fully homomorphic encryption (FHE), since the noise in homomorphically evaluated ciphertexts may leak information about the computation circuit, the work introduces a method to randomize intermediate computation results. The approach can be summarized as follows: first, it applies the efficiently computable function $\mathbf{G}_{rand}^{-1}(\cdot)$, to convert the homomorphically evaluated ciphertext into a discrete Gaussian sample. Then it performs an inner product (via ciphertext multiplication) with fresh encryption noise. This inner product term itself follows a discrete Gaussian distribution. To ensure that this term remains independent of the original noise in the evaluated ciphertext, an additional large noise term y is introduced. The key innovation of this work lies in its use of $\mathbf{G}_{rand}^{-1}(\cdot)$ to obfuscate the noise in homomorphically computed ciphertexts by transforming it into a discrete Gaussian sample. Once in this form, the noise becomes easier to manage—requiring only a moderately large noise injection (unlike traditional noise flooding techniques). Unlike the problem addressed in their work, our scheme requires analyzing whether the new noise term (a linear combination of the left ciphertext's noise plus the right ciphertext's noise) remains statistically independent of the right ciphertext's original noise. Crucially, since the new noise term is derived from a linear combination of the original noise vectors, the resulting multidimensional discrete Gaussian variables are not independent (their covariance is non-zero). Consequently, the noise distribution deviates from a spherical Gaussian, leading to fundamentally different proof techniques. For a detailed comparison, refer to their proof and our Lemma 3.

Similarly, with the goal of eliminating circuit information from ciphertext noise, the work [12] proposes an iterative bootstrapping-based sanitization process. The core idea involves repeatedly re-encrypting and homomorphically decrypting the ciphertext (i.e., bootstrapping) to progressively reduce the statistical discrepancy between the evaluated ciphertext and freshly encrypted ciphertexts. The process further enhances security by injecting small noise ("micro-flooding") after each bootstrapping operation, progressively obscuring the ciphertext's distribution. Through multiple iterations (e.g., λ times), the statistical distance between ciphertexts decreases exponentially (becoming $< 2^{\lambda}$), ultimately achieving a state where the processed ciphertext becomes statistically indistinguishable from a fresh ciphertext.

The work [15] builds upon and improves [6]. While the latter proposed ciphertext randomization for LWE-based FHE schemes, the former extends this result to the RLWE setting and further optimizes the ciphertext sanitization process from [12], achieving the desired privacy with just a single bootstrapping operation.

To summarize the key differences: in addressing the problem of eliminating information leakage from the noise (where we focus on initial ciphertext noise, as opposed to circuit privacy's concern with circuit information), these works employ fundamentally different approaches from ours. Their techniques include: discrete Gaussian sampling and bootstrapping-based methods.

Roadmap. In Section 2, we define some symbols and list some commonly used definitions and our extended results on lattice. In Section 3, we proved the discrete Gaussian version of smudging lemma. In Section 4, we constructed the MKFHE scheme based on LWE in the plain model. In Section 5 we prove the security of our scheme. In Section 6, we present the performance and complexity analysis of our scheme, and in Section 7, we provide a summary and outlook for future work.

2 Preliminaries

2.1 Notation

Let λ, n, and q be the security parameter, LWE dimension, and modulus base respectively. Let $\mathsf{negl}(\lambda)$ be a negligible function parameterized by λ. Lowercase bold letters such as $\mathbf{v}$, unless otherwise specified, represent vectors. Vectors are typically represented as row vectors, while matrices are denoted 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 the value a is chosen according to the distribution X. If X is a finite set, then $a \leftarrow X$ denotes that the value of a is uniformly sampled from X. For two distributions X and Y, let $\Delta(X, Y)$ denote the statistical distance between X and Y, $X \approx_s Y$ to represent that X and Y are statistically indistinguishable, while $X \approx_c Y$ represents that they are computationally indistinguishable.

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

2.2 Some background in probability theory

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

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

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

Average Conditional Min-Entropy(in [7]**).** Let X be a random-variable supported on a finite set $\mathcal{X}$, and let Z be a random variable supported on a finite set $\mathcal{Z}$. The average-conditional min-entropy $\widetilde{H}_\infty(X|Z)$ of X given Z is defined as

$$\widetilde{H}_\infty(X|Z) = -\log(E_z\left[\max_{x \in \mathcal{X}} \Pr[X = x|Z = z]\right]).$$

2.3 Gaussian distribution on Lattice

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

$$\mathcal{D}_{\Lambda+\mathbf{c},\sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda + \mathbf{c})}.$$

We note that $\rho_\sigma(\mathbf{x})$ is just a special case of $\rho_\Sigma(\mathbf{x})$, where $\Sigma = \sigma^2\mathbf{I}$. Therefore, some results on $\sigma^2\mathbf{I}$ should be naturally extended to Σ (symmetric positive definite).

Definition 3. *Let $\rho_\Sigma(\mathbf{x}) = e^{-\pi\mathbf{x}\Sigma^{-1}\mathbf{x}^T}$ be a Gaussian function with covariance matrix Σ (symmetric positive definite). Let $\Lambda \subset \mathbb{R}^n$ be a lattice, and $\mathbf{c} \in \mathbb{R}^n$. The discrete Gaussian distribution $\mathcal{D}_{\Lambda+\mathbf{c},\Sigma}$ with support $\Lambda + \mathbf{c}$ is defined as*

$$\mathcal{D}_{\Lambda+\mathbf{c},\Sigma}(\mathbf{x}) = \frac{\rho_\Sigma(\mathbf{x})}{\rho_\Sigma(\Lambda + \mathbf{c})}.$$

Obviously, the above definition does satisfy the definition of a probability distribution. For a positive definite matrix Σ, when $||\mathbf{x}|| \to \infty$, $\rho_\Sigma(\mathbf{x})$ converges.

Theorem 1 (Banaszczyk's spherical theorem [4]). *Let $\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^m : ||\mathbf{x}|| \leq 1\}$ be the closed ball of radius 1 in $\mathbb{R}^n$, for any lattice $\Lambda \in \mathbb{R}^m$, parameter $\sigma > 0$ and $u \geq 1/\sqrt{2\pi}$ it holds that*

$$\rho_\sigma(\Lambda \backslash u\sigma\sqrt{m}\mathcal{B}) \leq 2^{-c_u \cdot m} \cdot \rho_\sigma(\Lambda),$$

where $c_u = -\log(\sqrt{2\pi e}u \cdot e^{-\pi u^2})$.

Next we introduce the ellipsoidal version of Banaszczyk's spherical theorem, which will be used in our proof of the smudging lemma on discrete Gaussian distribution. We give the proofs of the theorem and lemma in Appendix A.1 A.2.

Poisson's summation formula. We recall that the Fourier transform of $\rho_\Sigma(\mathbf{x})$ is $\hat{\rho}_\Sigma(\mathbf{k}) = \det(\Sigma)\rho_{\Sigma^{-1}}(\mathbf{k})$. The Poisson's summation formula of $\rho_\Sigma(\mathbf{x})$ on a full-rank lattice Λ is

$$\rho_\Sigma(\Lambda) = \det(\Sigma)\det(\Lambda^*)\rho_{\Sigma^{-1}}(\Lambda^*).$$

Lemma 2. *For positive definite matrix Σ_1 and Σ_2, if $\Sigma_1\Sigma_2 - \Sigma_2$ is positive definite, then it holds that*

$$\rho_{\Sigma_1\Sigma_2}(\Lambda) \leq \det(\Sigma_1)\rho_{\Sigma_2}(\Lambda).$$

Theorem 2. *For any lattice $\Lambda \in \mathbb{R}^m$, let $\Sigma \in \mathbb{R}^{m\times m}$ be a positive definite matrix, $\mathcal{E}(k) = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{x}\Sigma^{-1}\mathbf{x}^T \leq k\}$ be a ellipsoid in $\mathbb{R}^n$ with radius $k > 0$, then it holds that*

$$\rho_\Sigma(\Lambda \backslash \mathcal{E}(k)) \leq 2^{-2k+m} \cdot \rho_\Sigma(\Lambda).$$

Definition 4 (Decision-LWE in [21]). *Let λ be security parameter, for parameters $n = n(\lambda)$ be an integer dimension, $q = q(\lambda) > 2$, $m = O(n \log q)$ be an integer, and a distribution $\chi = \chi(\lambda)$ over $\mathbb{Z}$, the $\mathsf{LWE}_{n,m,q,\chi}$ problem is to distinguish the following distribution*

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

As shown in Regev [19, 21], the $\mathsf{LWE}_{n,m,q,\chi}$ problem with χ being discrete Gaussian distribution with parameter $\sigma = \alpha q \geq 2\sqrt{n}$ is at least as hard as approximating the shortest independent vector problem (GapSVP_γ) to within a factor of $\gamma = \widetilde{O}(n/\alpha)$ in *worst case* dimension n lattices. It leads to the assumption $\mathcal{D}_0 \approx_c \mathcal{D}_1$.

2.4 Dual-GSW (DGSW) Encryption scheme

The Dual-GSW encryption scheme defined in [8] contains the four algorithms $\mathsf{Init}(\cdot)$, $\mathsf{KeyGen}(\cdot)$, $\mathsf{Enc}(\cdot)$, $\mathsf{Dec}(\cdot)$. It is noted that in [8], the ciphertext modulus is set as $q = \mathrm{poly}(N) \cdot n^{\omega(1)}$, where N is the number of parties and n is the LWE dimension. This modulus is essentially on the same scale as $q = 2^{\omega(\lambda L \log \lambda)} B_\chi$, due to the adoption of *ciphertext expansion* and *noise flooding* techniques originally proposed in [17] (see [8, page 657, 658] for reference). In contrast, our scheme does not rely on either of these two techniques. As a result, although our MKFHE construction is also based on the Dual-GSW framework, the ciphertext modulus in our scheme can be set as $q = 2^{O(\lambda+L)}$.

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L)$: For a given security parameter λ, circuit depth L, choose an appropriate lattice dimension $n = n(\lambda, L)$, $m = n \log q + \omega(\lambda)$. Let χ be a discrete Gaussian distribution over $\mathbb{Z}$ bounded by B_χ. Let χ' be a uniform distribution over $[-2^\lambda B_\chi, 2^\lambda B_\chi]$. Let $q = 2^{O(\lambda+L)} B_\chi$ be the ciphertext modulus. Output $\mathsf{pp} = (n, m, q, \chi, B_\chi)$ as the initial parameters.

– $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$: Input the initial parameters pp. Let $\mathbf{A} \leftarrow \mathbb{Z}_q^{(m-1)\times n}$, $\mathbf{s} \leftarrow \{0,1\}^{m-1}$, $\mathbf{b} = \mathbf{sA} \mod q$. Output $(\mathbf{A}, \mathbf{b})$ as the public key pk, $\mathbf{t} = (-\mathbf{s}, 1)$ as the private key sk.

– $\mathbf{C} \leftarrow \mathsf{Enc}(\mathsf{pk}, u)$: Input public key pk and plaintext $u \in \{0,1\}$, choose a random matrix $\mathbf{R} \leftarrow \mathbb{Z}_q^{n\times w}$, where $w = ml$, $l = \lceil \log q \rceil$ and an error matrix $\begin{pmatrix} \mathbf{E} \\ \mathbf{e} \end{pmatrix}$ where $\mathbf{E} \leftarrow \chi^{(m-1)\times w}$, $\mathbf{e} \leftarrow \chi'^{w}$ Output the ciphertext

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

where $\mathbf{G}$ is a gadget Matrix.

– $u \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathbf{C})$: Input private key sk, ciphertext $\mathbf{C}$, let $\mathbf{w} = (0, \ldots, \lceil q/2 \rceil) \in \mathbb{Z}_q^m$, $v = \langle \mathbf{tC}, \mathbf{G}^{-1}(\mathbf{w}^T) \rangle$, output $u = \lceil \frac{v}{q/2} \rceil$.

Brakerski et al. proved in [8] that DGSW is leak-resistant. Informally, even if a part of the private key of the DGSW scheme is leaked, the DGSW ciphertext remains semantically secure.

3 Smudging lemma over discrete Gaussian

In this section we will prove two results regarding discrete Gaussian on the integer lattice $\mathbb{Z}^n$. Simply put, when n is large enough, the distribution of the sum of n iid discrete Gaussian is statistically indistinguishable from the distribution of the sum of $n+1$ iid discrete Gaussian. This is similar to the continuous Gaussian distribution.

Lemma 3. *Let $n > 0$ be an integer, $\mathcal{D}_{\mathbb{Z}^n,\sigma}$ be the discrete gaussian distribution over $\mathbb{Z}^n$ with variance σ^2, U be the uniform distribution over $\{0,1\}^{n\times n}$. Let $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}$, $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}$, $\mathbf{M} \leftarrow U$. Let Σ and Σ' are the covariance matrix of $\mathbf{e}_1\mathbf{M}$ and $\mathbf{e}_1\mathbf{M} + \mathbf{e}_2$ respectively. Let $\delta \in \mathbb{R}$ and*

$$\frac{\rho_{\Sigma'}(\mathbb{Z}^n)}{\rho_{\Sigma}(\mathbb{Z}^n)} = \delta\sqrt{\frac{\det(\Sigma')}{\det(\Sigma)}},$$

if $\delta > e^{-2+\frac{6\pi}{n+1}}$, we have

$$\Delta(\mathbf{e}_1\mathbf{M}, \mathbf{e}_1\mathbf{M} + \mathbf{e}_2) < 2^{-n}.$$

Proof. We can think of $\mathbf{e}_1\mathbf{M}$ as an n-dimensional random variable $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ over $\mathbb{Z}^n$, where $\{x_i = \sum_{j=1}^n e_j z_{j,i}\}_{i\in[n]}$, e_j is the j-th element of $\mathbf{e}_1$, $z_{j,i}$ is the element in row j and column i of $\mathbf{M}$. According to the properties of covariance, we have the covariance matrix Σ of $\mathbf{x}$

$$\Sigma = \begin{pmatrix} \frac{1}{2}n\sigma^2 & \frac{1}{4}n\sigma^2 & \cdots & \frac{1}{4}n\sigma^2 \\ \frac{1}{4}n\sigma^2 & \frac{1}{2}n\sigma^2 & \cdots & \frac{1}{4}n\sigma^2 \\ & & \cdots & \\ \frac{1}{4}n\sigma^2 & \frac{1}{4}n\sigma^2 & \cdots & \frac{1}{2}n\sigma^2 \end{pmatrix}, \qquad Cov(x_i, x_j) \begin{cases} \frac{1}{2}n\sigma^2, & if\ i = j \\ \frac{1}{4}n\sigma^2, & if\ i \neq j \end{cases}$$

In the same way, we can also regard $\mathbf{e}_1\mathbf{M} + \mathbf{e}_2$ as a n-dimensional random variable $\mathbf{x}' = (x_1 + e'_1, x_2 + e'_2, \cdots, x_n + e'_n)$, where e'_i is the i-th element of $\mathbf{e}_2$. Let Σ' be the covariance matrix of $\mathbf{x}'$, by the properties of covariance, we have $\Sigma' = \Sigma + \sigma^2\mathbf{I}$. Thus, we have $\mathbf{x} \sim \mathcal{D}_{\mathbb{Z}^n,\Sigma}(\mathbf{x})$, and $\mathbf{x}' \sim \mathcal{D}_{\mathbb{Z}^n,\Sigma'}(\mathbf{x})$. The probability density function of $\mathbf{x}$ and $\mathbf{x}'$ are $f(\mathbf{x})$ and $g(\mathbf{x})$ respectively

$$f(\mathbf{x}) = \frac{\rho_\Sigma(\mathbf{x})}{\rho_\Sigma(\mathbb{Z}^n)} = \frac{e^{-\pi\mathbf{x}\Sigma^{-1}\mathbf{x}^T}}{\rho_\Sigma(\mathbb{Z}^n)}, \qquad g(\mathbf{x}) = \frac{\rho_{\Sigma'}(\mathbf{x})}{\rho_{\Sigma'}(\mathbb{Z}^n)} = \frac{e^{-\pi\mathbf{x}\Sigma'^{-1}\mathbf{x}^T}}{\rho_{\Sigma'}(\mathbb{Z}^n)}.$$

At this time, the intersection equation of $f(\mathbf{x})$ and $g(\mathbf{x})$ is

$$e^{\pi\mathbf{x}(\Sigma^{-1}-\Sigma'^{-1})\mathbf{x}^T} = \frac{\rho_{\Sigma'}(\mathbb{Z}^n)}{\rho_{\Sigma}(\mathbb{Z}^n)}.$$

Because $\Sigma' = \Sigma + \sigma^2\mathbf{I}$, we have $\Sigma'^{-1} = \Sigma^{-1} - (\Sigma + \frac{1}{\sigma^2}\Sigma^2)^{-1}$ by the Woodbury matrix identity or the Hua's identity. Thus, we have

$$e^{\pi\mathbf{x}(\Sigma+\frac{1}{\sigma^2}\Sigma^2)^{-1}\mathbf{x}^T} = \frac{\rho_{\Sigma'}(\mathbb{Z}^n)}{\rho_{\Sigma}(\mathbb{Z}^n)},$$

take the logarithm, we have

$$\mathbf{x}(\Sigma + \frac{1}{\sigma^2}\Sigma^2)^{-1}\mathbf{x}^T = \frac{1}{\pi}\ln\frac{\rho_{\Sigma'}(\mathbb{Z}^n)}{\rho_{\Sigma}(\mathbb{Z}^n)}.$$

Let $\mathbf{B} = \Sigma + \frac{1}{\sigma^2}\Sigma^2$, $a = \frac{1}{\pi}\ln\frac{\rho_{\Sigma'}(\mathbb{Z}^n)}{\rho_{\Sigma}(\mathbb{Z}^n)}$, we have the ellipsoid equation $\mathcal{E}_{ints}$ of the intersection of $f(\mathbf{x})$ and $g(\mathbf{x})$ is

$$\mathcal{E}_{ints}: \quad \mathbf{x}(a\mathbf{B})^{-1}\mathbf{x}^T = 1.$$

When $\mathbf{x}$ is on the ellipsoid $\mathcal{E}_{ints}$, we have $\mathbf{x}(a\mathbf{B})^{-1}\mathbf{x}^T = 1$ and $f(\mathbf{x}) = g(\mathbf{x})$, when $\mathbf{x}$ is outside $\mathcal{E}_{ints}$, we have $\mathbf{x}(a\mathbf{B})^{-1}\mathbf{x}^T > 1$ and $f(\mathbf{x}) < g(\mathbf{x})$, when $\mathbf{x}$ is inside the $\mathcal{E}_{ints}$, we have $\mathbf{x}(a\mathbf{B})^{-1}\mathbf{x}^T < 1$ and $f(\mathbf{x}) > g(\mathbf{x})$. By the definition of Statistical distance and the above result, we have

$$\Delta(\mathbf{x},\mathbf{x}') = \frac{1}{2}\sum_{\mathbf{x}\in\mathbb{Z}^n}|g(\mathbf{x}) - f(\mathbf{x})| = \frac{1}{2}\left(\sum_{\mathbf{x}\in\mathcal{E}_{ints}}(f(\mathbf{x}) - g(\mathbf{x})) + \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}}(g(\mathbf{x}) - f(\mathbf{x}))\right), \tag{4}$$

also because

$$\sum_{\mathbf{x}\in\mathbb{Z}^n} f(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{E}_{ints}} f(\mathbf{x}) + \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} f(\mathbf{x}) = 1. \tag{5}$$

$$\sum_{\mathbf{x}\in\mathbb{Z}^n} g(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{E}_{ints}} g(\mathbf{x}) + \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}) = 1. \tag{6}$$

Let (5) - (6), we have

$$\sum_{\mathbf{x}\in\mathcal{E}_{ints}} f(\mathbf{x}) - g(\mathbf{x}) = \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}) - f(\mathbf{x}). \tag{7}$$

Substituting Equation (7) into Equation (4), we have

$$\Delta(\mathbf{x},\mathbf{x}') = \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}) - f(\mathbf{x}) < \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}).$$

Because the "shapes" of $\mathcal{E}_{ints}$ and $g(\mathbf{x})$ are inconsistent (The "shape" of $\mathcal{E}_{ints}$ is $\mathbf{B}$, and the "shape" of $g(\mathbf{x})$ is Σ'), we need to find an ellipsoid with the "shape" of Σ' inscribed in $\mathcal{E}_{ints}$. Let $k > 0$ and

$$k\mathbf{x}^T = \Sigma'(a\mathbf{B})^{-1}\mathbf{x}^T.$$

When k takes the minimum eigenvalue of $\Sigma'(a\mathbf{B})^{-1}$, we have $k\mathbf{x}\Sigma'^{-1}\mathbf{x}^T = 1$ is inscribed in $\mathcal{E}_{ints}$. The minimum eigenvalue of $\Sigma'\mathbf{B}^{-1}$ and the maximum eigenvalue of $\mathbf{B}\Sigma'^{-1}$ are exactly reciprocals of each other, which is $\frac{n(n+1)}{4}$. (note that $\mathbf{B}\Sigma'^{-1} = \frac{1}{\sigma^2}\Sigma$). Therefore, the ellipsoid $\mathcal{E}_{insc}$ that is inscribed in $\mathcal{E}_{ints}$ is

$$\mathcal{E}_{insc}: \quad \mathbf{x}\Sigma'^{-1}\mathbf{x}^T = \frac{an(n+1)}{4}.$$

Thus, we have

$$\Delta(\mathbf{x}, \mathbf{x}') = \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}) - f(\mathbf{x}) < \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{ints}} g(\mathbf{x}) < \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{insc}} g(\mathbf{x}).$$

By Theorem 2 and the assumption $\delta > e^{-2+\frac{6\pi}{n+1}}$, we have

$$\sum_{\mathbf{x}\in\mathbb{Z}^n\setminus\mathcal{E}_{insc}} g(\mathbf{x}) < 2^{-\frac{an(n+1)}{4}+n} < 2^{-n}.$$

□

Remark. Note that $\int_{\mathbb{R}^n} \rho_\Sigma(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sqrt{\det(\Sigma)}$. In other words, when the ratio of the discrete Gaussian sum and the ratio of the continuous Gaussian integral are not significantly different (up to δ), Lemma 3 applies. We cannot accurately obtain the value of the discrete Gaussian sum $\rho_\Sigma(\mathbb{Z}^n)$, so we can only use the integral of the Gaussian function $\int_{\mathbb{R}^n} \rho_\Sigma(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sqrt{\det(\Sigma)}$ instead. This is our motivation for introducing δ. Numerical experiments show that the difference between the two is not significant, and the ratio is close to 1. Therefore $\delta > e^{-2+\frac{6\pi}{n+1}}$ should be considered a conservative estimate. The above results can be easily extended to discrete Gaussian matrices.

Lemma 4. *Let $m > 0$, $n > 0$ be two integers, $\mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$ be the discrete gaussian distribution over $\mathbb{Z}^{m\times n}$ with variance σ^2, U be the uniform distribution over $\{0,1\}^{n\times n}$. Let $\mathbf{E}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$, $\mathbf{E}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$, $\mathbf{M} \leftarrow U$. Let Σ and Σ' are the covariance matrix of $\mathbf{E}_1\mathbf{M}$ and $\mathbf{E}_1\mathbf{M}+\mathbf{E}_2$ respectively. Let $\delta \in \mathbb{R}$ and*

$$\frac{\rho_{\Sigma'}(\mathbb{Z}^{mn})}{\rho_\Sigma(\mathbb{Z}^{mn})} = \delta\sqrt{\frac{\det(\Sigma')}{\det(\Sigma)}},$$

if $\delta > e^{-2+\frac{2\pi(m+1)}{n+1}+\frac{2}{mn}}$, we have

$$\Delta(\mathbf{E}_1\mathbf{M}, \mathbf{E}_1\mathbf{M}+\mathbf{E}_2) < 2^{-n}.$$

Proof. The proof of Lemma 4 is exactly the same as the proof of Lemma 3, except that the covariance matrices of $\mathbf{E}_1\mathbf{M}$ and $\mathbf{e}_1\mathbf{M}$ are different. Also, we can think of $\mathbf{E}_1\mathbf{M}$ as an mn-dimensional random variable $\mathbf{x} = (x_1, x_2, \cdots, x_{mn})$ over $\mathbb{Z}^{mn}$, where $\{x_i = \sum_{j=1}^{n} e_{c,j} z_{j,d}\}_{i\in[mn]}$, $c = \lceil \frac{i}{n} \rceil$, $d = i \mod n$, $e_{c,j}$ is the element in row c and column j of $\mathbf{E}_1$, $z_{j,d}$ is the element in row j and column d of $\mathbf{M}$. Let $\mathbf{T} \in \mathbb{R}^{n\times n}$ be the symmetric matrix

$$\mathbf{T} = \begin{pmatrix} \frac{1}{2}n\sigma^2 & \frac{1}{4}n\sigma^2 & \cdots & \frac{1}{4}n\sigma^2 \\ \frac{1}{4}n\sigma^2 & \frac{1}{2}n\sigma^2 & \cdots & \frac{1}{4}n\sigma^2 \\ & & \cdots & \\ \frac{1}{4}n\sigma^2 & \frac{1}{4}n\sigma^2 & \cdots & \frac{1}{2}n\sigma^2 \end{pmatrix}$$

The covariance matrix $\Sigma \in \mathbb{R}^{mn\times mn}$ of the random variable $\mathbf{x}$ is

$$\Sigma = \begin{pmatrix} \mathbf{T} & & & \\ & \mathbf{T} & & \\ & & \cdots & \\ & & & \mathbf{T} \end{pmatrix}, \qquad Cov(x_i, x_j) \begin{cases} \frac{1}{2}n\sigma^2, & if\ i = j \\ \frac{1}{4}n\sigma^2, & if\ |i-j| < n,\ i \neq j \\ 0, & if\ |i-j| \geq n,\ i \neq j \end{cases}$$

The following proof is the same as Lemma 3, we omit it here. □

4 A MKFHE scheme based on DGSW in the plain model without noise flooding

Our scheme is based on the DGSW scheme. In this section, we first introduce a new key transform algorithm called $\mathsf{KeyLifting}(\cdot)$, then describe the entire scheme.

4.1 The key lifting Algorithm

The $\mathsf{KeyLifting}(\cdot)$ algorithm is a two-round interactive process defined in Algorithm 1. Without loss of generality, assuming that there are k participants in total, the input of the algorithm is the DGSW key pair $\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]}$ of all participants. After two rounds of interaction, the outputs is $\mathsf{hk}_i = (\mathbf{A}_i,\ \mathbf{b}_i)$ called the hybrid key.

Algorithm 1: $\mathsf{KeyLifting}(\cdot)$ converts DGSW key to hybrid key

Input: DGSW key pair $\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]}$.
Output: hybrid key $\mathsf{hk}_i = (\mathbf{A}_i, \mathbf{b}_i)$.
1: First round: participant i broadcasts pk_i and receives $\{\mathsf{pk}_j\}_{j\in[k]\setminus i}$ from the channel.
2: Second round: i generates and broadcasts $\{\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j\}_{j\in[k]\setminus i}$ and receives $\{\mathbf{b}_{j,i} = \mathbf{s}_j\mathbf{A}_i\}_{j\in[k]\setminus i}$ from the channel. Let $\mathbf{b}_i = \sum_{j=1}^{k} \mathbf{b}_{j,i}$
Return $\mathsf{hk}_i = (\mathbf{A}_i, \mathbf{b}_i)$.

The semi-malicious adversary may generate matrix $\{\mathbf{A}_j\}_{j\neq i}$ with trapdoor, then $\mathbf{s}_i$ is leaked. More specifically, in the $\mathsf{KeyLifting}(\cdot)$ phase, $\{\mathbf{b}_{i,j} = \mathbf{s}_i\mathbf{A}_j\}_{j\in[k],j\neq i}$ will lose $\mathbf{s}_i$ at most $(k-1)n\log q$ bits. Therefore, as long as the min-entropy of $\mathbf{s}_1$ after leakage remains greater than $\log q + 2\lambda$, that is, if the key length m satisfies $m - (k-1)n\log q > \log q + 2\lambda$, then the ciphertext remains computationally indistinguishable from uniform. This is precisely the property of leakage-resilient encryption, as described in the *Security* subsection on page 655 of [8], or refer to the detailed security proof of our scheme in Section 5 of this paper. The hybrid key of each participant is different, but the ciphertext generated by hybrid key directly supports homomorphic evaluation. As the Claim 1 states

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

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

it holds that(omit small error)

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

Proof. According to the $\mathsf{KeyLifting}(\cdot)$ algorithm, it holds that

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

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

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

4.2 Our scheme

Our scheme is similar to those proposed in [8] and [3], as all are based on the DGSW framework. However, unlike theirs, we eliminate the need for complex ciphertext expansion operations by introducing the keylifting algorithm to enable homomorphic evaluation. Moreover, our distributed decryption avoids the use of flooding noise. The scheme includes the following six algorithms $\mathsf{Init}(\cdot)$, $\mathsf{Gen}(\cdot)$, $\mathsf{KeyLifting}(\cdot)$, $\mathsf{Enc}(\cdot)$, $\mathsf{Eval}(\cdot)$, $\mathsf{LocalDec}(\cdot)$, $\mathsf{FinalDec}(\cdot)$.

- $\mathsf{pp} \leftarrow \mathsf{Init}(1^\lambda, 1^L, 1^W, k)$: Let λ be security parameter, L circuit depth, W circuit output length, lattice dimension $n = n(\lambda, L)$, noise distribution χ and χ' over $\mathbb{Z}$ bounded by B_χ and $2^\lambda B_\chi$ respectively, modulus $q = 2^{O(\lambda+L)}B_\chi$, number of partis k, $m = (kn + W + 1)\log q + 2\lambda$, suitable choosing above parameters to make $\mathsf{LWE}_{n,m,q,B_\chi}$ is infeasible. Output the initial parameters $\mathsf{pp} = (k, n, m, q, \chi, \chi')$.
- $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{Gen}(\mathsf{pp})$: Input the initial parameters pp. Let $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{(m-1)\times n}$, $\mathbf{s}_i \leftarrow 0, 1^{m-1}$, $\mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i \mod q$, output $(\mathbf{A}_i, \mathbf{b}_{i,i})$ as the DGSW public key $\mathsf{pk}_i, \mathbf{t}_i = (-\mathbf{s}_i, 1)$ as the DGSW private key sk_i of party i.
- $\mathsf{hk}_i \leftarrow \mathsf{KeyLifting}(\{\mathsf{pk}_i, \mathsf{sk}_i\}_{i\in[k]})$: All parties are engaged in the Algorithm 1, output the hybrid key hk_i.
- $\mathbf{C}_i \leftarrow \mathsf{Enc}(\mathsf{hk}_i, u_i)$: Input hybrid key hk_i, plaintext $u_i \in \{0, 1\}$, output ciphertext

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

  where $\mathbf{R} \leftarrow \mathbb{Z}_q^{n\times ml}$, $l = \lceil \log q \rceil$, $\mathbf{E} \leftarrow \chi^{(m-1)\times ml}$, $\mathbf{e} \leftarrow \chi'^{ml}$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g}$ is a gadget matrix.
- $\mathbf{C}_\mathcal{C} \leftarrow \mathsf{Eval}(S, \mathcal{C})$: Input the ciphertext set $S = \{\mathbf{C}_i\}_{i\in[N]}$ which are encrypted by the hybrid keys $\{\mathsf{hk}_i\}_{i\in[k]}$, the circuit $\mathcal{C}$ with input length N, depth L, output the evaluated ciphertext $\mathbf{C}_\mathcal{C}$.
- $p_i \leftarrow \mathsf{LocalDec}(\mathbf{C}_\mathcal{C}, \mathsf{sk}_i)$: Input the ciphertext $\mathbf{C}_\mathcal{C}$ and the private key sk_i of i, let $\mathbf{C}_\mathcal{C} = \begin{pmatrix} \mathbf{C}_{\mathsf{up}} \\ \mathbf{c}_{\mathsf{low}} \end{pmatrix}$, where $\mathbf{C}_{\mathsf{up}}$ is the first $m-1$ rows of $\mathbf{C}_\mathcal{C}$ and $\mathbf{c}_{\mathsf{low}}$ is last row of $\mathbf{C}_\mathcal{C}$. Party i computes $p_i = -\mathbf{s}_i\mathbf{C}_{\mathsf{up}}\mathbf{G}^{-1}(\mathbf{w}^T)$, where $\mathbf{w} = (0, \ldots, 0, \lfloor q/2 \rceil) \in \mathbb{Z}_q^m$, then broadcast p_i.
- $u_{\mathsf{dec}} \leftarrow \mathsf{FinalDec}(\{p_i\}_{i\in[k]})$: After receiving $\{p_i\}_{i\in[k]}$, let $p = \sum_{i=1}^k p_i + \mathbf{c}_{\mathsf{low}}\mathbf{G}^{-1}(\mathbf{w}^T)$, output $u_{\mathsf{dec}} = \left\lfloor \frac{p}{q/2} \right\rceil$.

4.3 Correctness of distributed decryption

This section verifies the correctness of distributed decryption and gives an expression for p_i (based on the simulated perspective of other $k-1$ parties), which is used to analyze the noise components in p_i. It is easy to see that for the evaluated ciphertext $\mathbf{C}_\mathcal{C}$, decryption with $\bar{\mathbf{t}}$ always satisfies the following equation

$$\bar{\mathbf{t}}\mathbf{C}_\mathcal{C} = \mathbf{e}_\mathcal{C} + u_\mathcal{C}\bar{\mathbf{t}}\mathbf{G}, \tag{8}$$

where $\mathbf{e}_\mathcal{C}$ is the noise accumulated after homomorphic evaluation, and $u_\mathcal{C}$ is the result of homomorphic evaluation. After further multiplication by $\mathbf{G}^{-1}(\mathbf{w}^T)$, we have

$$\bar{\mathbf{t}}\mathbf{C}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) = \mathbf{e}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) + u_\mathcal{C}\left\lfloor \frac{q}{2} \right\rceil, \tag{9}$$

According to the definition of p, it is obvious that $\bar{\mathbf{t}}\mathbf{C}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) = p$. The decryption result u_{dec} can be rewritten as

$$u_{\mathsf{dec}} = \left\lfloor \frac{p}{q/2} \right\rceil = \left\lfloor \frac{\mathbf{e}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) + u_\mathcal{C}\lfloor \frac{q}{2} \rceil}{q/2} \right\rceil = \left\lfloor \frac{\mathbf{e}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) + e_{\mathsf{round}} + u_\mathcal{C}\frac{q}{2}}{q/2} \right\rceil = \left\lfloor \frac{\mathbf{e}_\mathcal{C}\mathbf{G}^{-1}(\mathbf{w}^T) + e_{\mathsf{round}}}{q/2} + u_\mathcal{C} \right\rceil$$

where $|e_{\mathsf{round}}| < 1/2$, as long as $|\mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) + e_{\mathsf{round}}| < q/4$, we have the decryption result u_{dec} equal to the evaluation result $u_{\mathcal{C}}$.

Now that we know the decryption result u_{dec}, let's look at the expression of p_i, substituting $u_{\mathsf{dec}} = u_{\mathcal{C}}$ and $\bar{\mathbf{t}}\mathbf{C}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) = p$ into equation (9) we have $p = \mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) + u_{\mathsf{dec}}\left\lfloor\frac{q}{2}\right\rceil$. From the perspective of the other $k-1$ parties, p_i can be rewritten as

$$p_i = u_{\mathsf{dec}}\left\lfloor\frac{q}{2}\right\rceil + \mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) - \mathbf{c}_{\mathsf{low}}\mathbf{G}^{-1}(\mathbf{w}^T) - \sum_{j\neq i}^{k} p_j \tag{10}$$

Note that $\{p_i\}_{i\in[k]}$ are public and $\mathbf{c}_{\mathsf{low}}$, $\mathbf{G}^{-1}(\mathbf{w}^T)$ are known, anyone can compute $\mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T)$ after obtaining the decryption result u_{dec}. The $\mathbf{e}_{\mathcal{C}}$ is the noise accumulated after the homomorphic evaluation, which is related to the noise in the initial ciphertext and the circuit $\mathcal{C}$. In fact, it also contains the private key $\bar{\mathbf{t}}$. We will introduce the composition of $\mathbf{e}_{\mathcal{C}}$ in detail in the following subsection. Therefore, for security reasons, the traditional approach is to add a large noise to cover up $\mathbf{e}_{\mathcal{C}}$ before publishing p_i.

4.4 Decryption without noise flooding

To illustrate how our approach works, let us first review the noise flooding technique. We note that introducing noise flooding in the partial decryption phase is essential to guarantee the semantic security of initial ciphertext, and noise flooding achieves this by masking the private key and the noise in the initial ciphertext in the partial decryption. For partial decryption to be simulatable, the magnitude of the noise introduced needs to be exponentially larger than the noise after the homomorphic evaluation.

By noise flooding (in [3,8–10,17]). Let $\mathcal{C}$ be the Boolean circuit with input length N, depth L, let $S = \{\mathbf{C}_i\}_{i\in[N]}$ be the initial ciphertext set encrypted by hybrid key $\{\mathsf{hk}_i\}_{i\in[k]}$. Let $\bar{\mathbf{t}}$ be the decryption key defined in Claim 1. Taking S as input, homomorphically evaluation boolean circuit on $\mathcal{C}$ obtain ciphertext $\mathbf{C}_{\mathcal{C}}$. It holds that $\bar{\mathbf{t}}\mathbf{C}_{\mathcal{C}} = \mathbf{e}_{\mathcal{C}} + u_{\mathcal{C}}\bar{\mathbf{t}}\mathbf{G}$. Let $\mathbf{C}_{\mathsf{up}}$ is the first $m-1$ row of $\mathbf{C}_{\mathcal{C}}$ and $\mathbf{c}_{\mathsf{low}}$ is the last row of $\mathbf{C}_{\mathcal{C}}$. By the parameter settings in [17], $B_{smdg} = 2^{\lambda L\log\lambda}B_{\chi}$, $q = 2^{\omega(\lambda L\log\lambda)}B_{\chi}$, the flooding noise which introduced in $\mathsf{LocalDec}(\cdot)$ is $e_i \leftarrow [-B_{smdg}, B_{smdg}]$. The partial decryption result of party i would be $p_i = -\mathbf{s}_i\mathbf{C}_{up}\mathbf{G}^{-1}(\mathbf{w}^T) + e_i$. At this point, from the perspective of the other $k-1$ parties, p_i can be rewritten as

$$p_i = u_{\mathsf{dec}}\left\lfloor\frac{q}{2}\right\rceil + \mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) + e_i - \mathbf{c}_{\mathsf{low}}\mathbf{G}^{-1}(\mathbf{w}^T) - \sum_{j\neq i}^{k} p_j$$

Because $B_{smdg} = 2^{\lambda L\log\lambda}B_{\chi}$, $q = 2^{\omega(\lambda L\log\lambda)}B_{\chi}$, it holds that $|\mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T)/e_i| = \mathsf{negl}(\lambda)$, further $\mathbf{e}_{\mathcal{C}}\mathbf{G}^{-1}(\mathbf{w}^T) + e_i \approx_s e_i$. Therefore, the noise $\mathbf{e}_{\mathcal{C}}$ in p_i is drowned out by e_i.

Our approach (without noise flooding). In this subsection, we first use Lemma 4 to prove the asymmetry of ciphertext multiplication. Thus, by multiplying the initial ciphertext set $S = \{\mathbf{C}_i\}_{i\in[N]}$ by a dummy ciphertext whose plaintext is 1, the noise in the ciphertext after multiplication is independent of the noise in S. Secondly, we point out that the noise obtained by decrypting the ciphertext after homomorphic evaluation is actually composed of the noise in the dummy ciphertext and the private key. Finally, combined with the leakage-resilient property of our scheme, the partial decryption result of party i can be simulated even without the flooding noise.

Let $\mathbf{C}_i$ be an initial ciphertext in S and $\mathbf{C}_{\mathsf{dummy}}$ be the dummy ciphertext with plaintext 1, we have

$$\mathbf{C}_{\mathsf{dummy}} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \mathbf{E} + \mathbf{G}, \qquad \mathbf{C}_i = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R}_i + \mathbf{E}_i + u_i \mathbf{G}.$$

After left-multiplying $\mathbf{C}_i$ by $\mathbf{C}_{\mathsf{dummy}}$, let $\mathbf{C}'_i = \mathbf{C}_{\mathsf{dummy}} \mathbf{G}^{-1}(\mathbf{C}_i) = \Pi_i + \Psi_i + u_i \mathbf{G}$ where

$$\Pi_i = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} \mathbf{G}^{-1}(\mathbf{C}_i) + \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R}_i, \qquad \Psi_i = \mathbf{E} \mathbf{G}^{-1}(\mathbf{C}_i) + \mathbf{E}_i.$$

It holds that $\bar{\mathbf{t}} \Pi_i = \mathbf{0}$, and by Lemma 4, we have $\Psi_i \approx_s \mathbf{E}\mathbf{G}^{-1}(\mathbf{C}_i)$. Thus $\mathbf{C}'_i$ is independent of the noise $\mathbf{E}_i$ in the initial ciphertext $\mathbf{C}_i$. Let $S' = \{\mathbf{C}'_i\}_{i \in [N]}$, then the ciphertext $\mathbf{C}'_{\mathcal{C}}$ obtained by homomorphically evaluation on S' and boolean circuit $\mathcal{C}$ is independent of the initial noise $\{\mathbf{E}_i\}_{i \in [N]}$ in $\{\mathbf{C}_i\}_{i \in [N]}$.

Next, we analyze what information the noise obtained after decrypting $\mathbf{C}'_{\mathcal{C}}$ contains. Similar to the ciphertext of the DGSW scheme, the ciphertext $\mathbf{C}'_{\mathcal{C}}$ can be expressed as $\mathbf{C}'_{\mathcal{C}} = \Pi + \Psi_{\mathcal{C}} + u_{\mathcal{C}} \mathbf{G}$ where Π satisfying $\bar{\mathbf{t}} \Pi = \mathbf{0}$, $\Psi_{\mathcal{C}}$ is the noise depending on $\{\Psi_i\}_{i \in [N]}$ and boolean circuits $\mathcal{C}$. $u_{\mathcal{C}}$ is the result depending on the plaintext set $\{u_i\}_{i \in [N]}$ and $\mathcal{C}$. Decrypting $\mathbf{C}'_{\mathcal{C}}$ we have $\bar{\mathbf{t}} \mathbf{C}'_{\mathcal{C}} = \bar{\mathbf{t}} \Psi_{\mathcal{C}} + u_{\mathcal{C}} \bar{\mathbf{t}} \mathbf{G}$. Let $\mathbf{e}'_{\mathcal{C}} = \bar{\mathbf{t}} \Psi_{\mathcal{C}}$, we call $\mathbf{e}'_{\mathcal{C}}$ the noise obtained by decrypting $\mathbf{C}'_{\mathcal{C}}$. Obviously, $\mathbf{e}'_{\mathcal{C}}$ is composed of the private key $\bar{\mathbf{t}}$ and $\{\Psi_i\}_{i \in [N]}$, and is independent of the initial noise $\{\mathbf{E}_i\}_{i \in [N]}$ in the initial ciphertext set S. At this time, the partial decryption result of party i is (without flooding noise)

$$p_i = u_{\mathsf{dec}} \left\lfloor \frac{q}{2} \right\rceil + \mathbf{e}'_{\mathcal{C}} \mathbf{G}^{-1}(\mathbf{w}^T) - \mathbf{c}_{\mathsf{low}} \mathbf{G}^{-1}(\mathbf{w}^T) - \sum_{j \neq i}^{k} p_j$$

The inner product $\mathbf{e}'_{\mathcal{C}} \mathbf{G}^{-1}(\mathbf{w}^T) \in \mathbb{Z}_q$ only leaks party i's private key $\mathbf{s}_i$ with at most $\log q$ bits. As $\mathcal{C}$ is a boolean circuit with output length W, the partial decryption leaks $W \log q$ bits of $\mathbf{s}_i$.

4.5 Parameter settings

In this subsection, we discuss how to determine some parameters and their impact on security and efficiency. The main parameters are the number of parties k, the dimension of the LWE problem n, the LWE noise B_χ, the key length m, the ciphertext modulus q, the security parameter λ, the circuit depth L and output length W. For the ciphertext modulus q, in subsection 4.3, we give the condition $|\mathbf{e}_{\mathcal{C}} \mathbf{G}^{-1}(\mathbf{w}^T) + e_{\mathsf{round}}| < q/4$ for correct decryption where $\mathbf{e}_{\mathcal{C}}$ is the accumulated noise after homomorphic evaluation bounded by $(m \log q)^L 2^\lambda B_\chi$. $\mathbf{G}^{-1}(\mathbf{w}^T)$ is the binary decomposition of $q/2$, and $|e_{\mathsf{round}}| < 1/2$. Thus $q > 4 \log q (m \log q)^L 2^\lambda B_\chi$ must be satisfied. For the key length m, in order to ensure that the entropy of $\mathbf{s}$ is still large enough after losing $(kn + W) \log q$ bits, satisfying $m - (kn + W) \log q > \log q + 2\lambda$ (the remaining $\log q + 2\lambda$ is used for the DGSW anti-leakage security proof, see Lemma 5), the lower bound of m should be $(kn + W + 1) \log q + 2\lambda$. The security parameter λ is generally set to 128 (note that λ is the security parameter of the statistical distance, not the security level of the LWE problem). The remaining parameters, such as k, L and W depend on the specific application. We can refer to several (n, q) parameter combinations provided in the Homomorphic Encryption Standard [1] for an LWE security level of 128 where the noise $B_\chi \approx 6\sigma$. Therefore, the order of determining parameters should be

1. Determine k, L, W and λ according to the specific application.
2. Refer to the Homomorphic Encryption Standard [1] to select a specific (n, q) pair.
3. Let $m = (kn + W + 1) \log q + 2\lambda$ and verify whether (n, q) determined in step 2 satisfy $q > 4 \log q (m \log q)^L 2^\lambda B_\chi$. If not, select a larger n, q combination until the inequality is satisfied.

For example, selecting parameters $k = 3$, $L = 2$, $W = 2^{24}$ and $\lambda = 128$, we choose the homomorphic encryption standard-recommended parameters ($n = 8192, q = 2^{220}$), which also provides an exact 128-bit security level for LWE problem and $q > 4\log q(m\log q)^L 2^\lambda B_\chi$. It should be noted that when selecting the parameter pair (n, q), the recommended values provided in standard documents serve only as general guidelines. In practical applications, (n, q) should be chosen as small as possible, provided that the decryption inequality is satisfied and the LWE security level remains at least 128 bits (which can be estimated using the Lattice Estimator [2]).

5 Security Proof against Semi-Malicious Adversary

In this section, we first prove that DGSW with partial key leakage is still semantically secure. The proof process is basically the same as Lemma 1 in [8], except that the amount of key leakage is different. Then, we reduce the security of our scheme to the security of DGSW ciphertext.

Lemma 5. *Let k, m, n, q, w, W be the parameters and χ, χ' be the distribution defined in our scheme. Let $\mathbf{A}_1 \leftarrow \mathbb{Z}_q^{(m-1)\times n}$, $\mathbf{s}_1 \leftarrow \{0,1\}^{m-1}$, $\mathbf{b}_{1,1} = \mathbf{s}_1\mathbf{A}_1 \mod q$, the DGSW public key be $(\mathbf{A}_1, \mathbf{b}_{1,1})$, and the ciphertext of 0 be*

$$\mathbf{C}_{DGSW} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix}.$$

Even if $\mathbf{s}_1$ loses $(kn + W)\log q$ bits ($kn\log q$ bits lost in the keylifting phase and $W\log q$ bits lost in the distributed decryption phase), it holds that $\mathbf{C}_{DGSW}$ is still computationally indistinguishable from the uniform distribution.

Proof. Let $\mathbf{C}_0 = \mathbf{A}_1\mathbf{R} + \mathbf{E}_0$, $\mathbf{c}_1 = \mathbf{b}_{1,1}\mathbf{R} + \mathbf{e}_1$, we have $\mathbf{c}_1 = \mathbf{s}_1\mathbf{A}_1\mathbf{R} + \mathbf{e}_1 = \mathbf{s}_1(\mathbf{C}_0 - \mathbf{E}_0) + \mathbf{e}_1 = \mathbf{s}_1\mathbf{C}_0 - \mathbf{s}_1\mathbf{E}_0 + \mathbf{e}_1$. Note that $\mathbf{E}_0$ is sampled from χ and $\mathbf{e}_1$ is sampled from χ'. By Lemma 1, we have $\mathbf{c}_1 \approx_s \mathbf{s}_1\mathbf{C}_0 + \mathbf{e}_1$. Further, according to the LWE assumption, we can replace $\mathbf{C}_0$ with a uniform matrix $\mathbf{U}$ on $\mathbb{Z}_q^{(m-1)\times w}$ and we have $(\mathbf{C}_0, \mathbf{c}_1) \approx_c (\mathbf{U}, \mathbf{s}_1\mathbf{U} + \mathbf{e}_1)$. Note that $m = (kn + W + 1)\log q + 2\lambda$, under a semi-malicious adversary, the *keylifting* phase $\mathbf{s}_1$ will lose $kn\log q$ bits and the distributed decryption phase will lose $W\log q$ bits. At this time $m - (kn + W)\log q > \log q + 2\lambda$, according to the leftover hash lemma, let $\mathbf{U}$ be the seed and $\mathbf{s}_1$ be the source, we have that $(\mathbf{U}, \mathbf{s}_1\mathbf{U})$ is statistically indistinguishable from the uniform distribution. Through the above hybrid argument, we have that the DGSW ciphertext and the uniform distribution are computationally indistinguishable, even if $\mathbf{s}_1$ is lossy. □

We complete the simulation by constructing a reduction from our ciphertext to the DGSW ciphertext.

Theorem 3. *Assume that the first party is the Challenger and the other $k-1$ parties are controlled by the adversary $\mathcal{A}$, if $\mathcal{A}$ can distinguish the ciphertext of our scheme from the uniform distribution, then he can also distinguish the DGSW ciphertext from the uniform distribution with the same (up to negligible) advantage.*

Proof. Consider the following Game

1. Challenger generates $\mathsf{pk}_1 = (\mathbf{A}_1, \mathbf{b}_{1,1} = \mathbf{s}_1\mathbf{A}_1)$ where $\mathbf{A}_1 \leftarrow \mathbb{Z}_q^{(m-1)\times n}$, $\mathbf{s}_1 \leftarrow \{0,1\}^{m-1}$ sends pk_1 to adversary $\mathcal{A}$.
2. After receiving pk_1, $\mathcal{A}$ generates $\{\mathsf{pk}_i\}_{i\in[k]/1}$, where $\mathsf{pk}_i = (\mathbf{A}_i, \mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i)$, sends it to Challenger.
3. After receiving $\{\mathsf{pk}_i\}_{i\in[k]/1}$, Challenger sets $\{\mathbf{b}_{1,i} = \mathbf{s}_1\mathbf{A}_i\}_{i\in[k]/1}$ (the leakage of $\mathbf{s}_1$), sends it to $\mathcal{A}$.
4. After receiving $\{\mathbf{b}_{1,i}\}_{i\in[k]/1}$, $\mathcal{A}$ adaptively chooses $\{\mathbf{s}'_i\}_{i\in[k]/1}$, where $\mathbf{s}'_i \in \{0,1\}^{m-1}$, sets $\{\mathbf{b}_{i,1} = \mathbf{s}'_i\mathbf{A}_1\}_{i\in[k]/1}$, sends it to Challenger.

5. After receiving $\{\mathbf{b}_{i,1}\}_{i\in[k]/1}$, Challenger sets $\mathsf{hk}_1 = (\mathbf{A}_1, \sum_{i=1}^{k} \mathbf{b}_{i,1})$.
6. $\mathcal{A}$ chooses a bit $u \leftarrow \{0,1\}$, sends it to Challenger.
7. Challenger chooses a bit $\alpha \leftarrow \{0,1\}$, if $\alpha = 0$ sets $\mathbf{C} \leftarrow \mathsf{Enc}(\mathsf{hk}_1, u)$, otherwise $\mathbf{C} \leftarrow \mathbb{Z}_q^{m\times ml}$, sends $\mathbf{C}$ to $\mathcal{A}$.
8. After receiving $\mathbf{C}$, $\mathcal{A}$ outputs bit $\bar{\alpha}$, if $\bar{\alpha} = \alpha$, then $\mathcal{A}$ wins.

Obviously the above Game simulates the $\mathsf{KeyLifting}(\cdot)$ and $\mathsf{Enc}(\cdot)$ of our scheme. The first four steps outline the detailed process of $\mathsf{KeyLifting}(\cdot)$ assuming a rushing adversary. After the third step of the above game, $\mathcal{A}$ obtained pk_1 and $\{\mathbf{b}_{1,i}\}_{i\in[k]/1}$ (the leakage of $\mathbf{s}_1$). We use the ciphertext of DGSW to construct $\mathbf{C}$. Let

$$\mathbf{C}_{\mathsf{DGSW}} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{b}_{1,1} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}.$$

be the Dual-GSW ciphertext generated by pk_1, which is semantically secure by Lemma 5, even if $\mathbf{s}_1$ is lossy. Let $\mathbf{s}' = \sum_{i=2}^{k} \mathbf{s}'_i$ are adaptively chosen by $\mathcal{A}$ after seeing pk_1 and $\{\mathbf{b}_{1,i}\}_{i\in[k]/1}$. Let

$$\mathbf{C}' = \mathbf{C}_{\mathsf{DGSW}} + \begin{pmatrix} \mathbf{0} \\ \mathbf{s}'\mathbf{C}_0 \end{pmatrix},$$

it holds that $\mathbf{s}'\mathbf{C}_0 = \mathbf{s}'(\mathbf{A}_1\mathbf{R} + \mathbf{E}_0) = \sum_{i=2}^{k} \mathbf{b}_{i,1}\mathbf{R} + \mathbf{s}'\mathbf{E}_0$ and

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

Because $\mathbf{e}_1$ is uniform over $[-2^{\lambda}B_{\chi}, -2^{\lambda}B_{\chi}]$ and $||\mathbf{s}'\mathbf{E}_0||_{\infty} < kmB_{\chi}$, then $||\mathbf{s}'\mathbf{E}_0/\mathbf{e}_1||_{\infty} = \mathsf{negl}(\lambda)$. By Lemma 1, it holds that $\mathbf{C}' \approx_s \mathbf{C}$. Let $\mathsf{Adv} = ||Pr[\bar{\alpha} = \alpha] - \frac{1}{2}||$ denote $\mathcal{A}$'s advantage in winning the game, if $\mathcal{A}$ can distinguish between $\mathbf{C}$ and the uniform distribution by advantage Adv, then he can also distinguish between $\mathbf{C}_{\mathsf{DGSW}}$ and the uniform distribution with the same (up to negligible) advantage. □

6 Performance Analysis

We give the complexity analysis of our scheme, focusing on the public key size, ciphertext size, the number of communication rounds during the initialization phase, the underlying security assumptions, and the computational model adopted. Furthermore, we compare our approach with existing state-of-the-art schemes to highlight the improvements.

In our scheme, the public key is the Dual-GSW public key $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} \in \mathbb{Z}_q^{(m-1)\times n}$ and $\mathbf{b} \in \mathbb{Z}_q^n$ with $m = (kn + W + 1)\log q + 2\lambda$, $q = 2^{O(\lambda+L)}$. Therefore, the public key size is $O\left(n(kn+W)(\lambda+L)^2\right)$. The ciphertext is a matrix over $\mathbb{Z}_q^{m\times m\log q}$ with a size of $O\left((kn+W)^2(\lambda+L)^4\right)$. Additionally, to generate the hybrid key, we introduce two rounds of interaction in the $\mathsf{KeyLifting}(\cdot)$ algorithm. We summarize a comparison with other multi-key FHE schemes in Table 2. For fairness, all the compared schemes are LWE-based.

Here, we focus on explaining why the public key and evaluation key sizes in [3] and [8] are relatively large. This is primarily because both schemes follow the *ciphertext expansion* paradigm introduced in [17]. In brief, in schemes [17], [8] and [3], the initial ciphertexts do not support homomorphic evaluation directly as they are generated by different public keys. To enable homomorphic evaluation, the ciphertexts must first undergo an *expansion* process, which relies on the evaluation keys. The evaluation keys, in essence, is the ciphertexts encoding of the random matrix embedded in the initial

Table 2: Performance Analysis

Scheme	PubKey + EvalKey	Ciphertext	Round	Assumption	Adversary	Model
MKFHE [20]	$O(n^4(k+L)^4)$	$O(n^2k^2(\lambda L)^2)$	–	LWE	semi-honest	CRS
MKFHE [8]	$O(k^3n^4(\lambda L)^6)$	$O(k^3n^2(\lambda L)^4)$	2	LWE	semi-malicious	–
MKFHE [3]	$O(k^3n^4(\lambda L)^6)$	$O(k^4n^2(\lambda L)^4)$	2	LWE	malicious-client	–
Our scheme	$O(n(kn+W)(\lambda+L)^2)$	$O((kn+W)^2(\lambda+L)^2)$	2	LWE	semi-malicious	–

n denotes the LWE dimension, k is the number of parties, L represents the circuit depth, λ is the security parameter, and W denotes the output length of the circuit. The PubKey + EvalKey column indicates the combined size of the public key and the evaluation key, while the Ciphertext column shows the ciphertext size, measured in bits. The Round column specifies the number of communication rounds required during the initialization phase. The Assumption column lists the underlying security assumptions of each scheme. The Adversary column shows the adversary model. The Model column indicates the computational model adopted. CRS stands for "Common Random String", which is generated by a trusted third party and distributed to all participants for use during key initialization.

ciphertext. For instance, in [3], given an initial ciphertext $\mathbf{C} \in \mathbb{Z}_q^{m \times m \log q}$, *expansion* requires encrypting each entry of the random matrix $\mathbf{R} \in \mathbb{Z}_q^{n \times m \log q}$ in $\mathbf{C}$. As a result, the evaluation keys consist of $nm \log q$ ciphertexts over $\mathbb{Z}_q^{m \times m \log q}$ with total size $O(k^3n^4(\lambda L)^6)$. For further details, please refer to Section 3.2 of [3], specifically the $\mathsf{Enc}_{\mathsf{BHP}}(\cdot)$ and $\mathsf{Expand}_{\mathsf{BHP}}(\cdot)$ algorithms. Table 2 shows that the ciphertext in scheme [3] is k times larger than that in [8]. Although both schemes expand the ciphertext into matrices over $\mathbb{Z}_q^{km \times km \log q}$, the actual ciphertext is sparse, containing only $2k-1$ matrices over $\mathbb{Z}_q^{m \times m \log q}$, with the remaining positions being zeros. To protect against malicious clients, the scheme [3] fills these zero positions with dummy ciphertexts, resulting in an overall ciphertext size that is k times larger than that of scheme [8]. For more details, refer to Section 3.3 of [3].

7 Conclusion

In this work, we presented a Multi-Key Fully Homomorphic Encryption (MKFHE) scheme that eliminates the need for exponential noise flooding during distributed decryption—a major bottleneck in prior constructions. By leveraging the asymmetric noise-masking properties of Dual-GSW ciphertext multiplication and proving a new smudging lemma for discrete Gaussian distributions, we demonstrated that the initial ciphertext noise can be decoupled from the decryption process. Leveraging the leakage-resilience of our encryption scheme, we are able to eliminate the use of flooding noise in distributed decryption, thereby reducing the ciphertext modulus q from $2^{\omega(\lambda L \log \lambda)}$ to $2^{O(\lambda+L)}$, aligning it with single-key FHE schemes while preserving security. Our approach also introduces a $\mathsf{KeyLifting}(\cdot)$ technique that removes the need for *ciphertext expansion*, streamlining homomorphic evaluation across multiple keys. The scheme achieves security against semi-malicious adversaries in the plain model, relying solely on the standard LWE assumption. Compared to state-of-the-art MKFHE constructions, our work improves efficiency, as evidenced by the reduced ciphertext size and public key complexity. Future work will focus on extending the proposed approach to MKFHE schemes based on polynomial rings, with the goal of enabling practical implementation and deployment in general-purpose applications.

8 Declarations

8.1 Availability of data and materials

No data were generated or analyzed in this study.

8.2 Conflict of Interest

The authors declare no conflict of interest.

9 Acknowledgments

The authors would like to thank anonymous reviewers for their helpful comments and suggestions. This study was funded by Academicians in Chongqing lead the scientific and technological innovation guidance project (2022YSZX-JCX0011CSTB, CSTB2023YSZX-JCX0008), National Key Research and Development Project (2020YFA0712300), Chongqing Talents Plan Youth Top-Notch Project (2021000263), Western Young Scholars Program of CAS. A preprint of this work has published on the Cryptology ePrint Archive [11].

References

1. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.: Homomorphic Encryption Standard, pp. 31–62. Springer International Publishing, Cham (2021), https://doi.org/10.1007/978-3-030-77287-1_2
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology 9(3), 169–203 (2015), https://doi.org/10.1515/jmc-2015-0016
3. Attrapadung, N., Hanaoka, G., Hiromasa, R., Matsuda, T., Schuldt, J.C.: Maliciously circuit-private multi-key fhe and mpc based on lwe. Designs, Codes and Cryptography 91(5), 1645–1684 (2023)
4. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen 296, 625–635 (1993)
5. Bensimhoun, M.: N-dimensional cumulative function, and other useful facts about gaussians and normal densities. Jerusalem, Israel, Tech. Rep pp. 1–8 (2009)
6. Bourse, F., Pino, R.D., Minelli, M., Wee, H.: FHE circuit privacy almost for free. Cryptology ePrint Archive, Paper 2016/381 (2016), https://eprint.iacr.org/2016/381
7. Brakerski, Z., Döttling, N.: Two-message statistically sender-private ot from lwe. In: Beimel, A., Dziembowski, S. (eds.) Theory of Cryptography. pp. 370–390. Springer International Publishing, Cham (2018)
8. Brakerski, Z., Halevi, S., Polychroniadou, A.: Four round secure computation without setup. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography. pp. 645–677. Springer International Publishing, Cham (2017)
9. Brakerski, Z., Perlman, R.: Lattice-based fully dynamic multi-key fhe with short ciphertexts. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016. pp. 190–213. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
10. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. Cryptology ePrint Archive, Paper 2014/798 (2014), https://eprint.iacr.org/2014/798
11. Dai, X., Wu, W., Feng, Y.: Key lifting : Multi-key fully homomorphic encryption in plain model without noise flooding. Cryptology ePrint Archive, Paper 2022/055 (2022), https://eprint.iacr.org/2022/055
12. Ducas, L., Stehlé, D.: Sanitization of FHE ciphertexts. Cryptology ePrint Archive, Paper 2016/164 (2016), https://eprint.iacr.org/2016/164
13. Fiore, D., Mitrokotsa, A., Nizzardo, L., Pagnin, E.: Multi-key homomorphic authenticators. IET Information Security 13, 618–638 (2019), https://digital-library.theiet.org/doi/abs/10.1049/iet-ifs.2018.5341
14. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013. pp. 75–92. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
15. Kluczniak, K.: Circuit privacy for FHEW/TFHE-style fully homomorphic encryption in practice. Cryptology ePrint Archive, Paper 2022/1459 (2022), https://eprint.iacr.org/2022/1459
16. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing. pp. 1219–1234. STOC'12, Association for Computing Machinery, New York, NY, USA (2012), https://doi.org/10.1145/2213977.2214086

17. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key fhe. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016. pp. 735–763. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)

18. Passelègue, A., Stehlé, D.: Low communication threshold fully homomorphic encryption. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024. pp. 297–329. Springer Nature Singapore, Singapore (2025)

19. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. pp. 333–342. STOC '09, Association for Computing Machinery, New York, NY, USA (2009), https://doi.org/10.1145/1536414.1536461

20. Peikert, C., Shiehian, S.: Multi-key fhe from lwe, revisited. In: Hirt, M., Smith, A. (eds.) Theory of Cryptography. pp. 217–238. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)

21. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. pp. 84–93. STOC '05, Association for Computing Machinery, New York, NY, USA (2005), https://doi.org/10.1145/1060590.1060603

22. Slotani, M.: Tolerance regions for a multivariate normal population. Annals of the Institute of Statistical Mathematics 16(1), 135–153 (1964)

23. Zhou, T., Li, N., Lai, Q., Yang, X., Han, Y., Liu, W.: Efficient multi-key fully homomorphic encryption over prime cyclotomic rings with fewer relinearisations. IET Information Security 15(6), 472–486 (2021), https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ise2.12036

Appendix

A The proof of Lemma 2 and Theorem 2

Recall that the integral of $\rho_\Sigma(\mathbf{x})$ is $\det(\Sigma)$, thus the Fourier transform of $\rho_\Sigma(\mathbf{x})$ is $\hat{\rho}_\Sigma(\mathbf{k}) = \det(\Sigma)\rho_{\Sigma^{-1}}(\mathbf{k})$, and the Poisson summation formula of $\rho_\Sigma(\mathbf{x})$ is $\rho_\Sigma(\Lambda) = \det(\Sigma)\det(\Lambda^*)\rho_{\Sigma^{-1}}(\Lambda^*)$.

A.1 The proof of Lemma 2

By the Poisson summation formula, we have

$$\begin{aligned}\rho_{\Sigma_1\Sigma_2} &= \det(\Sigma_1)\det(\Sigma_2)\det(\Lambda^*)\rho_{(\Sigma_1\Sigma_2)^{-1}}(\Lambda^*),\\ \det(\Sigma_1)\rho_{\Sigma_2} &= \det(\Sigma_1)\det(\Sigma_2)\det(\Lambda^*)\rho_{\Sigma_2^{-1}}(\Lambda^*).\end{aligned}$$

If $\rho_{\Sigma_2^{-1}}(\Lambda^*) > \rho_{(\Sigma_1\Sigma_2)^{-1}}(\Lambda^*)$, then we done. For $\rho_{\Sigma_2^{-1}}(\mathbf{x}) = e^{-\pi\mathbf{x}\Sigma_2\mathbf{x}^T}$, $\rho_{(\Sigma_1\Sigma_2)^{-1}}(\mathbf{x}) = e^{-\pi\mathbf{x}\Sigma_1\Sigma_2\mathbf{x}^T}$, if $\Sigma_1\Sigma_2 - \Sigma_2$ is positive semi-definite, then we have $\rho_{\Sigma_2^{-1}}(\mathbf{x}) > \rho_{(\Sigma_1\Sigma_2)^{-1}}(\mathbf{x})$, thus $\rho_{\Sigma_2^{-1}}(\Lambda^*) > \rho_{(\Sigma_1\Sigma_2)^{-1}}(\Lambda^*)$.

A.2 The proof of Theorem 2

Let $\mathcal{E}(k) = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{x}\Sigma_2^{-1}\mathbf{x}^T < k\}$ be the ellipsoid with "shape" Σ_2 and radius k, and positive definite matrix Σ_1, Σ_2, we have

$$\begin{aligned}
\rho_{\Sigma_1\Sigma_2}(\Lambda) &\geq \rho_{\Sigma_1\Sigma_2}(\Lambda\backslash\mathcal{E}(k)) \\
&= \sum_{\mathbf{x}\in(\Lambda\backslash\mathcal{E}(k))} e^{-\pi\mathbf{x}(\Sigma_1\Sigma_2)^{-1}\mathbf{x}^T+\pi\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T} \cdot e^{-\pi\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T} \\
&= \sum_{\mathbf{x}\in(\Lambda\backslash\mathcal{E}(k))} e^{\frac{1}{2}\pi\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T} \cdot e^{-\pi\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T} \qquad (\text{let } \Sigma_1 = 2\mathbf{I}) \\
&\geq \sum_{\mathbf{x}\in(\Lambda\backslash\mathcal{E}(k))} e^{\frac{1}{2}\pi k} \cdot e^{-\pi\mathbf{x}\Sigma_2^{-1}\mathbf{x}^T} \\
&= e^{\frac{\pi}{2}k} \cdot \rho_{\Sigma_2}(\Lambda\backslash\mathcal{E}(k)).
\end{aligned}$$

By Lemma 2 we have $2^m \cdot \rho_{\Sigma_2}(\Lambda) \geq \rho_{2\Sigma_2}(\Lambda)$ and $e^{\frac{\pi}{2}} > 4$, thus $\rho_{\Sigma_2}(\Lambda\backslash\mathcal{E}(k)) < 2^{m-2k} \cdot \rho_{\Sigma_2}(\Lambda)$.