Summation rather than Concatenation: a more efficient MKFHE scheme in the plain model

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

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

Abstract. For Multi-key Fully Homomorphic scheme(MKFHE) based on the Learning With Error(LWE) problem, in order to enable multi-key function, ciphertext expansion is required. In order to achieve ciphertext expansion, the random matrix used in encryption must be encrypted. For an boolean circuit with input length N, multiplication depth L, security parameter λ , the number of additional encryptions introduced to achieve ciphertext expansion is $O(N\lambda^6L^4)$, which is a lot of overhead for computationally sensitive local users. In order to alleviate this overhead, we proposed a weak version of the MKFHE, using the leakage resilient property of Leftover Hash Lemma(LHL), the first weak version of the MKFHE scheme is constructed under plain model. The total private key is the sum of all participant keys. We note that previous MKFHE schemes with this key structure are all based on Common Reference String Model(CRS). Our scheme is simpler and more efficient in construction: we dont need to encrypt the random matrix, so the extra overhead $O(N\lambda^6L^4)$ is reduced to $O(N)$.
For MKFHE based on Ring Learing With Error(RLWE) problem, since the Regularity Lemma on rings does not have the corresponding leakage resilient property, we can only construct the weak-MKFHE scheme under the random oracle model.

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

1 Introduction

Fully Homomorphic Encryption(FHE). The concept of Fully homomorphic encryption(FHE) was proposed by Rivest et al. [RAD+78], within a year of publishing of the RSA scheme [RSA78]. It was not until 2009 that Gentry gave the first truly FHE scheme in his doctoral dissertation [Gen09a]. Based on Gentry's ideas, a series of FHE schemes have been proposed [Gen09b] [vGHV10] [BGV12] [FV12] [GSW13] [CGGI16] [CKKS17], and their security and efficiency have been continuously improved. FHE is suitable to the problem of unilateral outsourcing computations. However in the case of multiple data providers, in order to support homomorphic evaluation, data must be encrypted by a common public key. Due to privacy of data, it is unreasonable to require participants to use other people's public keys to encrypt their own data.

Multiparty Computation (MPC). This problem was initialized by Yao in [Yao82] [Yao86], who considered two-party scenarios and gave a solution. Later, Goldreich, Micali and Wigderson extended the model to k participants with malicious adversaries in [MGW87]. Compared to FHE, MPC is more mature, as [Orl11] mentioned, generic MPC is a fast moving field: In 2009 the first implementation of MPC for 2 parties with active security [PSSW09], was able to evaluate a circuit of 3×10^4 gates in 10^3 seconds. Only two years after, the same circuit is evaluated in less than 5 seconds [NNOB12]. Subsequently, for active attacks, the literature [LP07] [Lin13] made a series of improvements.The first large-scale and practical application of multi-party computation (demonstrated on an actual auction problem) took place in Denmark in January 2008. However, MPC also has disadvantages: it suffers from high communication overhead and is vulnerable to attacks by corrupt participants, although these problems can be solved, but at the cost of high computational overhead.

Multi-key Fully Homomorphic Encryption(MKFHE). In order to solve this dilemma, López-Alt et al. proposed the concept of MKFHE in [LTV12] and construct the first MKFHE scheme based on modified-NTRU [SS11]. Conceptually, it is an enhancement of the FHE on function. MKFHE allows data provider to encrypt data independent from other participants, its key generation and

data encryption are done locally. To get the evaluated result, all participants are required to execute a round of threshold decryption protocol.

After López-Alt et al. proposed the concept of MKFHE, many schemes were proposed. In 2015, Clear and McGoldrick [CM15] constructed a GSW [GSW13] type multi-key fully homomorphic scheme based on LWE. This scheme defined the total key as the concatenation of all keys, introduced CRS and circular security assumptions, and constructed a masking scheme to converts the ciphertext under single key to the ciphertext under total key,which only supports single-hop computation. In 2016, Mukherjee and Wich [MW16], Perkert and shiehian [PS16], Brakerski and Perlman [BP16] constructed MKFHE scheme based on GSW respectively. [MW16] simplified the mask scheme of [CM15], and focused on constructing a two-round secure multi-party computing protocol. The work of [PS16] and [BP16] is dedicated to constructing a multi-hop multi-key fully homomorphic encryption scheme, but their methods are different. [BP16] introduces bootstrapping to realize ciphertext expansion, thereby realizing the multi-hop function. [PS16] realize multi-hop function through ingenious construction. It is worth mentioning that all MKFHE schemes constructed based on the GSW scheme require a ciphertext expansion procedure, so the random matrix corresponding to ciphertext needs to be encrypted, which leads to unsatisfactory efficiency of the GSW-type multi-key scheme.

1.1 Motivation

Why do we choose to construct a multi-key homomorphic scheme based on LWE? Now there are many multi-key homomorphic scheme based on RLWE problem, such as [CDKS19] [MTBH21], and because of the smaller public key, compact structure and fast arithmetic operation over rings, it is generally believed that the homomorphic scheme based on RLWE is more efficient than the scheme based on LWE . This is because the LWE-based MKFHE scheme can use the leakage resilient property of LHL over the Integers $\mathbb{Z}$ to remove CRS, which is incomparable to the RLWE-based homomorphic scheme.

Structure is a double-edged sword: due to the more compact structure on polynomial ring and various efficient ring algorithms, it is generally believed that FHE scheme based on RLWE is more efficient than the homomorphic scheme based on LWE. This is the reason why most current homomorphic schemes are constructed based on RLWE, but LHL lemma over integer ring $\mathbb{Z}$ enjoys the leakage resilient property : It can transform an average quality random sources into higher quality [ILL89], which is incomparable to general polynomial ring $R : \mathbb{Z}[x]/f(x), f(x) = x^d + 1$. Thanks to this property, the LWE-based multi-key homomorphic scheme can remove CRS, but the RLWE-based MKFHE scheme can't, because the regularity lemma [LPR13] over polynomial ring does not have this property: As [DSGKS21]mentioned if the j-th Number theoretical Transfer(NTT) coordinate of each ring element in $\mathbf{x} = (x_1, \ldots, x_l)$. is leaked, then the j-th NTT coordinate of $a_{l+1} = \sum a_i x_i$ is defined, and so a_{l+1} is very far from uniform: Yet this is only a 1/n leakage rate.

Therefore, no matter how efficient the RLWE multi-key homomorphism scheme is, it has not been possible to get rid of CRS so far. In some specific scenarios, such as the data provider questioning the randomness of the common public string, or challenging the fairness of a trusted third party, to deal with this dilemma, we can only choose the MKFHE based on LWE assumption. However, The current LWE-based MKFHE efficiency is hardly satisfactory

For the multi-key homomorphism scheme based on LWE, the ciphertext under different keys needs to be expanded, so it is necessary to encrypt the random matrix $\mathbf{R} \in \mathbb{Z}_q^{m \times m}$ of each ciphertext, for parameters $m = n \log q + \omega(\log \lambda)$, $q = 2^{\lambda L} B_\chi$, circuit input length N , the additional encryption operation introduced is $O(N\lambda^6 L^4)$, which for single-key FHE is $O(N)$. For computing-sensitive participants, this is a lot of overhead. Therefore, we propose the concept of weak-MKFHE, which allows participants to conduct two rounds of interaction and pull encryption operation of participant back to $O(N)$.

As the reason mentioned above, we cannot apply the LWE construction method trivially to RLWE-type MKFHE. As a compromise, we introduce a round of bit commitment protocol to guarantee the independence of each participants, construct the corresponding weak-MKFHE based on ROM, and optimize the re-linear algorithm of ciphertext. For k participants, previous schemes with concatenation key structure, the ciphertext after tensor product is $O(k^2)$ dimension, so the complexity of the re-linearization algorithm depends on k. If the sum key structure is adopted, the ciphertext after tensor product is only 4 dimensions, we can pull the ciphertext back to the initial dimension by one shot.

1.2 Our Results

In order to eliminate the extra overhead caused by the encryption of random matrix, we proposed the concept of weak-MKFHE, using the leak resilient property of LHL to construct the first weak version of the MKFHE scheme under the plain model. The total private key is the sum of the private keys of all participants. We note that previous MKFHE schemes adopt this key structure are all based on the CRS model. Not only is the CRS model removed, our solution is simpler and more efficient in construction: we don't need to encrypt the random matrix. For MKFHE based on the RLWE assumption, since regularity lemma [LPR13] on rings has no corresponding leakage resilient properties, we can only construct the MKFHE scheme under the random oracle model. We give a review of our two scheme below.

Scheme#1: LWE-based weak MKFHE under plain model:
The security of *Scheme*#1 based on the Decision-LWE assumption. For a circuit with an input length N, our scheme requires local participants to perform $O(N)$ encryption operations, in contrast, for those schemes that require ciphertext expansion, the required encryption operations is $O(N\lambda^6 L^4)$. In order to ensure the semantic security of encryption and make the threshold decryption procedure simulatable, we have to choose a larger smuging error to conceal the local decryption result. At the same time, we bounded the participants k by $\mathsf{poly}(\lambda)$, because a larger k will lead to a larger smuging error, which further leads to a larger q. After estimation, we choose $q = 2^{\lambda L} B_\chi$, for such q, the approximate factor of the GapSVP problem on lattice is $\tilde{O}(2^{\lambda L})$. For detailed security and parameters, please refer to Section 4.

We give the efficiency comparison with the scheme [PS16] in Table1. Since we have no ciphertext expansion, our scheme has lower computational overhead.

Scheme	Space		Time
	Public key	Ciphertext	EvalkeyGen
[PS16]	$\tilde{O}(\lambda^6 L^4(k+N\lambda^3 L^2))$	$\tilde{O}(Nk^2\lambda^6 L^4)$	$\tilde{O}(N\lambda^{14} L^9)$
Scheme#1	$\tilde{O}(k^2\lambda^6 L^4)$	$\tilde{O}(Nk^2\lambda^8 L^6)$	-

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

Scheme#2: RLWE-based weak MKFHE under ROM:
Scheme#2 is based on circular RLWE. Our approach is very simple. We introduce a bit commitment protocol to guarantee the randomness of each participant's public key. Due to the sum key structure, the dimension of $\mathbf{t} \otimes \mathbf{t}$ is independent of k, so the re-linear algorithm pull the ciphertext after tensor product back to initial dimension by one shot, in addition, the "one shot re-linear algorithm" introduces less noise. We compared with [CDKS19] in terms of memory and computational overhead, the results are shown in Table2.

Scheme	Space		Time	
	Evalkey	Ciphertext	Relinear	Mult
[CDKS19]	$\tilde{O}(kd)$	$\tilde{O}(kd)$	$\tilde{O}(k^2 d)$	$\tilde{O}(k^2 d)$
Scheme#2	$\tilde{O}(kd)$	$\tilde{O}(d)$	$O(1)$	$\tilde{O}(d)$

Table 2. Memory (bit-size) and computational overhead (number of scalar operations over $\mathbb{Z}_q$). The notation $\tilde{O}$ hides logarithmic factors. k denotes the number of participants; d denotes the dimension of the RLWE problem.

1.3 Related works

Unlike our scheme, [CM15] [PS16] [MW16] [BP16] used the concatenation of all private key as the total key structure, and the common reference string are introduced. [AJL+12] is the first scheme that introduce the summation of all private key as the total key,which is also under common reference string. [BHP17] is the first scheme using the leakage resilient property of LHL to get rid of the common reference string, which has the concatenation total key structure, and random matrix must be encrypted by local parties for ciphertext expansion. To be honest, our scheme are the combination of those two schemes.

1.4 Overview of our construciton

***Scheme*#1:** Similar to [BHP17], *Scheme*#1 is based on the Dual-GSW scheme. First, let's review the Dual-GSW scheme:
Let public key $pk = (\mathbf{A}, \mathbf{b} = \mathbf{sA})$, private key $sk = \mathbf{t} = (\mathbf{s}, 1)$, ciphertext:

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \mathbf{E} + u\mathbf{G}, \quad \text{Obviously }, \mathbf{tC} \approx u\mathbf{tG} \text{ (omit small noise)}$$

For the convenience of explanation, we assume that there are only two participants p_1, p_2, naturally, the whole process can be extended to N participants.

Key Generation: The interactive key generation includes the following three steps

- p_1 generates $(\mathbf{A}_1, \mathbf{s}_1)$, set $\mathbf{s}_1$ as a private key and then broadcasts $\mathbf{A}_1$ (p_2 performs the same operation)
- After receiving $\mathbf{A}_2$(sent by p_2), p_1 computes $\mathbf{b}_{11} = \mathbf{s}_1\mathbf{A}_1$, $\mathbf{b}_{12} = \mathbf{s}_1\mathbf{A}_2$, and discloses $\mathbf{b}_{11}, \mathbf{b}_{12}$.
- After receiving $\mathbf{b}_{21} = \mathbf{s}_2\mathbf{A}_1$(sent by p_2), p_1 generates public key pk_1=$(\mathbf{A}_1, \mathbf{b}_1)$, where $\mathbf{b}_1 = \mathbf{b}_{11} + \mathbf{b}_{21}$ (p_2 performs the same operation)

Encryption: Let the plaintext of p_1 and p_2 be u_1, u_2, the corresponding ciphertexts under public keys pk_1 and pk_2 are

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

Decryption: Let $\mathbf{t} = (-\mathbf{s}_1 - \mathbf{s}_2, 1)$, obviously $\mathbf{tC}_1 \approx u_1\mathbf{tG}$, $\mathbf{tC}_2 \approx u_2\mathbf{tG}$ that is, although $\mathbf{C}_1$ and $\mathbf{C}_2$ are encrypted by different public key, they are both ciphertexts under private key $\mathbf{t}$. Thus, ciphertexts under different public keys can also perform homomorphic evaluation without ciphertext expansion, and participants do not need to encrypt the random matrix $\mathbf{R}$ to prepare for ciphertext expansion. The subsequent processes, such as module reduction, bootstrapping, etc. are the same as the single-key FHE scheme. For a more detailed description, please refer to section 4.

***Scheme*#2:** *Scheme*#2 is quite simple. Compared with [CDKS19], it has one more round of bit commitment protocol. In addition, due to the sum key structure, re-linear algorithm can pull the ciphertext after the tensor product back to initial dimension by one shot. For a more details, please refer to section 5.

2 Preliminaries

2.1 Notation:

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

In order to decompose $\mathbb{Z}_q$ into binary, we review the Gadget matrix [MP12] [AP14] here, let $\mathbf{G}^{-1}$ be the computable function that for any

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

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

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

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

In order to prove the security of our scheme under plain model and enable the simulatability of threshold decryption, we need the following lemma:

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

2.2 The Learning With Error Problem(LWE)

The Learning With Error problem was introduced by Regev [Reg05].

Definition 3. *Let λ be security parameter. For parameters $n = n(\lambda), q = q(\lambda) > 2$,and a distribution $\chi = \chi(\lambda)$ over $\mathbb{Z}$, the $LWE_{n,q,\chi}$ problem is to distinguish the following distribution:*

- *Distribution 0: the jointly distribution $(\mathbf{A}, \mathbf{b}) \in (\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n)$ is computed by $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})\quad \mathbf{b} \leftarrow U(\mathbb{Z}_q^n)$*
- *Distribution 1: the jointly distribution $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^n$ is computed by $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})\quad \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\quad \mathbf{s} \leftarrow U(\mathbb{Z}_q^n)\quad \mathbf{e} \leftarrow \chi^m$*

Regev proved that the $LWE_{n,q,\chi}$ problem is true as long as certain worst case lattice problems are hard to solve using a quantum algorithm, so we have the following theorem which is implicit in [Reg05]

Theorem 4. *Let λ be security parameter, $n = n(\lambda)$, $q = q(\lambda)$ be integer and let $\chi = \chi(\lambda)$ be distribution over $\mathbb{Z}$, we have the jointly distribution $(\mathbf{A}, \mathbf{b})$ is computational indistingquishable from uniform random:*

$$(\mathbf{A}, \mathbf{b}) \overset{comp}{\approx} (\mathbf{A}, \mathbf{z})$$

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

2.3 The Ring Learning With Error Problem(RLWE)

Lyubaskevsky, Peikert and Regev defines The RLWE problem in [LPR10] as follows:

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

- *Distribution 0: the jointly distribution $(a, b) \in R_q^2$ is sampled by $(a, b) \leftarrow U(R_q^2)$.*
- *Distribution 1: the jointly distribution $(a, b) \in R_q^2$ is computed by $a \leftarrow U(R_q),\ b = as + e$, where $s \leftarrow U(R_q),\ e \leftarrow \chi$.*

[LPR10] proved that the $RLWE_{n,q,\chi}$ problem is infeasible as long as the approximate worst case shortest vector problem(SVP) over ideal lattice are hard to solve, so we have the following result which is implicit in [LPR10]

$$(a, b) \overset{\mathsf{comp}}{\approx} (a, z)$$

where $(a, b) \leftarrow U(R_q^2)$, $z = as + e$, $s \leftarrow U(R_q)$, $e \leftarrow \chi$.

Specially, [LPR10] indicated that The $RLWE_{n,q,\chi}$ problem is also infeasible when s is sampled from nosie distribution χ. In homomorphic encryption, this property is especially popular, because the low-norm s introduces less noise during homomorphic computation.

2.4 Dual-GSW Encryption scheme

The Dual-GSW encryption scheme and GSW encryption scheme is similar to Dual Regev scheme and Regev scheme : public key and encryption structure are just the opposite. Dual-GSW scheme is defined as follows:

- $pp \leftarrow Dual-GSW.setup(1^\lambda, 1^L)$: For a given security parameter λ, circuit depth L, Choose a appropriate lattice dimension $n = n(\lambda, L)$, $m = n\log q + \omega(\lambda)$, a discrete Gaussian distribution $\chi = \chi(\lambda, L)$ over $\mathbb{Z}$ which is bouned by B_χ, module $q = \mathsf{poly}(n)B_\chi$ to meet the LWE_{n,q,χ,B_χ}, Output $pp = (n, m, q, \chi, B_\chi)$ as the initial parameters.
- $(pk, sk) \leftarrow Dual-GSW.keyGen(pp)$: Let private key $\mathbf{s} \leftarrow U\{0,1\}^{m-1}$, public key $pk = (\mathbf{A}, \mathbf{b})$, where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m-1\times n})$, $\mathbf{b} = \mathbf{sA} \mod q$
- $C \leftarrow Dual-GSW.enc(pk, u)$: Choose a random Matrix $\mathbf{R} \leftarrow U(\mathbb{Z}_q^{n\times w})$, $w = m\log q$ and an error matrix $\mathbf{E} \leftarrow \chi^{n\times w}$, Output the ciphertext :

$$\mathbf{C} = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \mathbf{E} + u\mathbf{G}, \text{ where } \mathbf{G} \text{ is a gadget Matrix.}$$

- $u \leftarrow Dual-GSW.decrypt(sk, C)$: Let $\mathbf{t} = (-\mathbf{s}, 1)$, $\mathbf{v} = \mathbf{tC} = \mathbf{tE} + u\mathbf{tG}$, check the value of $\mathbf{v}$ output 0 if it close to 0, or 1 otherwise.

Homomorphic addition and multiplication: For ciphertext $\mathbf{C}_1$, $\mathbf{C}_2 \in \mathbb{Z}_q^{m\times w}$ let $\mathbf{C}_{add} = \mathbf{C}_1 + \mathbf{C}_2$, $\mathbf{C}_{mult} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$ It is easy to verify that$\mathbf{C}_{aad}$ and$\mathbf{C}_{mult}$ are plaintext of $u_1 + u_2$ and u_1u_2, respectively.

For the security and correctness of the Dual-GSW scheme, please refer to [cited]. Compared with the GSW scheme, Dual-GSW scheme has bigger ciphertext, which is $O(n^2\log^3 q)$, while $O(n^2\log q)$ for GSW scheme. As [BHP17] mentioned, the Dual-GSW scheme makes it more convenient to use the leakage resilient property of LHL to remove CRS.

2.5 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 weak-MKFHE we proposed. The main difference is that in MKFHE, each participant is required in key generation and encryption phase independently generates their own keys and completes the encryption operation without interaction between participants. These two phases are similar to single-key homomorphic encryption, the computational overhead is independent of k and only related to λ and L, only in the decryption phase, interaction is involved when participants perform a round of decryption protocol.

However, In the definition of weak-MKFHE, we do not have such restrictions. We allow interaction in the key generation phase, and the computational overhead depends on the participants, which is helpful to reduce the computational overhead and eliminate CRS. We review the definition of MKFHE first, and weak-MKFHE will be defined later.

Definition 6. *Let λ be the security parameter, L be the circuit depth, and k be the number of participants. A Leveled multi-key fully homomorphic encryption scheme consists of a tuple of efficient probabilistic polynomial time algorithms MKFHE=(Init, MKgen, MKenc, MKexpand, MKEval, Decrypt)*

- *$params \leftarrow Init(1^\lambda, 1^L)$: Input security parameter λ, circuit depth L, Output system paremeter params. We assume that all algorithm take params as input.*
- *$(pk_i, sk_i) \leftarrow$ MKGen(params, id):On input params, identity id, the key generation algorithm output a key pair for participant p_i..*
- *$c_i \leftarrow MKenc(pk_i, u_i)$:Input pk_i and u_i, output ciphertext c_i.*
- *$\bar{c}_i \leftarrow MKEpand(pk, c_i)$:Input the ciphertext c_i of participant p_i, the public key set $pk = \{pk_i\}_{i\in[k]}$ of all participants, output expanded ciphertext $\bar{c}_i$ which is under $f(sk_i, \ldots sk_k)$ whose structure is undefined.*
- *$\bar{c}_{eval} \leftarrow MK.Eval(\bar{c}, D_c)$:Input the description D_c of circuit, the set of all ciphertext $\bar{c} = \{\bar{c}_1 \ldots \bar{c}_N\}$ while N is the input length of circuit, output evaluated ciphertext $\bar{c}_{eval}$*

– $u \leftarrow MKDec(\bar{c}_{eval}, f(sk_1 \dots sk_k))$: *Input evluated ciphertext* $\bar{c}_{eval}$*, total private key function* $f(sk_1 \dots sk_k)$*, output* u

Note:

1. The Expand algorithm is not necessary. For example, in the RLWE-based MKFHE scheme, the ciphertext expansion process is trivial, but in the LWE-based MKFHE scheme, the ciphertext expansion is a complicated and time-consuming process.
2. The ciphertext structure function $f(sk_1 \dots sk_k)$ represents an organization form, or a certain function, which is not unique. For example, it can be the concatenation of all keys or the sum of all keys.

Correctness and Compactness : a leveled MKFHE scheme is correct, if for a given security parameter λ, circuit depth L, and participants k, we have:

$$\Pr\left[\, Decrypt(f(sk_1 \dots sk_k), \bar{c}_{eval}) \neq C(u_1 \dots u_N) \,\right] = \mathsf{negl}(\lambda).$$

where C is a circuit with input length N and depth less than L

3 weak-MKFHE scheme

Here we first give the definition of weak-MKFHE, and then construct a weak-MKFHE based on the Dual-GSW scheme.

3.1 The definition of weak-MKFHE

The properties implicited in MKFHE: In the formal definition of the MKFHE [LTV12], the key generation and encryption phase are both localized operations. The computation overhead of each participant only depends on security parameters, circuit depth, and is independent from other parameters, which means there is no interaction between participant. Different from the standard MKFHE, in our weak-MKFHE, similar with [BHP17] for the purpose to remove CRS, we allow interaction which is constant round between participants.

Definition 7. *A weak-MKFHE scheme is a tuple of probabilistic polynomial time algorithm weak-MKFHE=(Init, Constant Round KeyGen, MKenc, MKeval, Dec), which can be divided into two phases, online phase: Constant Round KeyGen and Decryption, where interaction is allowed between participants, but the interaction rounds should be constant and independent from other parameters, local phase : Init, MKenc, and MKeval, whose operations do not involve interaction. These five algorithms are described as follows:*

– $pp \leftarrow Init(1^\lambda, 1^L)$*:Input security parameter* λ*, circuit depth* L*, output public parameters* pp*.*
– $(pk_i, sk_i) \leftarrow KeyGen(pp, id)$*:Input public parameter* pp*, identity* id*, output the key pair of particpant* p_i
– $c_i \leftarrow wmkEnc(pk_i, u_i)$*: Input* u_i *and* pk_i*, output ciphertext* c_i
– $\hat{c} \leftarrow wmkEval(C, S)$*: Input circuit* C*, ciphertext set* $S = \{c_i\}_{i\in[N]}$ *, output ciphertext* $\hat{c}$
– $u \leftarrow wmkDec(\hat{c}, f(sk_1 \dots sk_k))$*: Input evaluated ciphertext* $\hat{c}$*,* $f(sk_1 \dots sk_k)$*, output* u*.*

We note that: weak-MKFHE does not have a ciphertext expansion procedure, indeed the inputed ciphertext in $wmkEnc(pk_i, u_i)$ is encrypted by participants under their own public key, however, which still supports homomorphic operations. Our construction below details it.

Similar to MKFHE, we require weak-MKFHE to satisfy the following properties: Here we just briefly review it, refer to [LTV12] for details:

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

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

Correctness and Compactness A leveled wkMKFHE scheme is correct if for a given security parameter λ, circuit depth L, participants k, we have the following

$$\Pr\left[\, Decrypt(f(sk_1 \dots sk_k), \hat{c}) \neq C(u_1 \dots u_N) \,\right] = \mathsf{negl}(\lambda).$$

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

4 Scheme#1:a weak-MKFHE scheme from Dual-GSW

Scheme#1 is different from the Dual-GSW scheme only in the public key: Let $\mathbf{s} = \sum_{i=1}^{k} \mathbf{s}_i$, for participant p_i, $pk_i = (\mathbf{A}_i, \mathbf{b}_i)$, where $\mathbf{b}_i = \mathbf{s}\mathbf{A}_i$. We will point out later, however, this structure will drew some security concerns, but more attention should be paid to its advantages. For any public key pk_i, the corresponding private key is $\mathbf{t} = (-\mathbf{s}, 1)$, that is to say, for ciphertexts under different public keys, homomorphic evaluation are supported without ciphertext expansion. Let $\mathbf{C}_1$ and $\mathbf{C}_2$ be the ciphertext of $u_1,\ u_2$ under $pk_1,\ pk_2$ respectively, Obviously $\mathbf{t}\mathbf{C}_1 \approx u_1\mathbf{t}\mathbf{G},\ \mathbf{t}\mathbf{C}_2 \approx u_2\mathbf{t}\mathbf{G}$, and for $\mathbf{C}_{add} = \mathbf{C}_1 + \mathbf{C}_2, \mathbf{C}_{mult} \approx \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$, we have $\mathbf{t}\mathbf{C}_{add} \approx (u_1 + u_2)\mathbf{t}\mathbf{G}$, $\mathbf{t}\mathbf{C}_{mult} \approx u_1u_2\mathbf{t}\mathbf{G}$, we detail our construction as follows:

- $params \leftarrow Init(1^\lambda, 1^L)$:Let λ be security parameter, L be circuit depth, lattice dimension $n = n(\lambda, L)$, noise distribution χ over $\mathbb{Z}$, and $e \leftarrow \chi$, where $|e|$ is bounded by B_χ, modulus $q = 2^{\lambda L}B_\chi$, $k = \mathsf{poly}(\lambda)$, $m = kn\log q + \lambda$, suitable choosing above parameters makes LWE_{n,q,B_χ} is infeasible. Output $params = (k, n, m, q, \chi, B_\chi)$
- *constant round keyGen*: p_i generates $\mathbf{A}_i \leftarrow U(\mathbb{Z}_q^{m-1\times n})$, $\mathbf{s}_i \leftarrow U\{0,1\}^{m-1}$, and let $\mathbf{b}_{i,i} = \mathbf{s}_i\mathbf{A}_i \mod q$
  - First round: p_i broadcasts $(\mathbf{A}_i,\ \mathbf{b}_{i,i})$ and receives all $\{\mathbf{A}_j,\ \mathbf{b}_{j,j}\}_{j\in[k]/i}$
  - Second round: p_i generates and discloses $\{\mathbf{b}_{ij}\}_{j\in[k]}$, where $\mathbf{b}_{ij} = \mathbf{s}_i\mathbf{A}_j \mod q$

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

$$\text{let } \mathbf{b}_i = \sum_{j=1}^{k} \mathbf{b}_{ji},\ p_i \text{ output } pk_i = (\mathbf{A}_i, \mathbf{b}_i) \text{ as public key}$$

- $\mathbf{C}_i \leftarrow wMK.Enc(pk_i, u_i)$: Input public key pk_i, plaintext u_i, output ciphertext $\mathbf{C}_i = \begin{pmatrix}\mathbf{A}_i\\ \mathbf{b}_i\end{pmatrix}\mathbf{R} + \mathbf{E} + u_i\mathbf{G}$, where $\mathbf{R} \leftarrow \chi^{n\times ml}$, $\mathbf{E} = \begin{pmatrix}\mathbf{E}_0\\ \mathbf{e}_1\end{pmatrix}$, $\mathbf{E}_0 \leftarrow \chi^{(m-1)\times ml}$, $\mathbf{e}_1 \leftarrow \chi'^{\,ml}$, χ' is a distribution over $\mathbb{Z}$, satisfying $|\mathbf{e}_1|$ is bounded by $2^{\lambda^{\epsilon_1}}B_\chi$, $\epsilon_1 \in (0, \frac{1}{2})$, $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{g}$ is a gadget matrix.
- $\hat{\mathbf{C}} \leftarrow wMK.Eval(S, C)$: Input set $S = \{\mathbf{C}_i\}_{i\in[N]}$ which are ciphertext under different public key, circuit C, output $\hat{\mathbf{C}}$.

Homomorphic addition and multiplication

- $\mathbf{C}_{add} \leftarrow wMK.add(\mathbf{C}_1, \mathbf{C}_2)$: Input ciphertext $\mathbf{C}_1,\ \mathbf{C}_2$, output $\mathbf{C}_{add} = \mathbf{C}_1 + \mathbf{C}_2$, Obviously $\mathbf{t}\mathbf{C}_{add} \approx (u_1 + u_2)\mathbf{t}\mathbf{G}$
- $\mathbf{C}_{mult} \leftarrow wMK.mult(\mathbf{C}_1, \mathbf{C}_2)$: Input ciphertext $\mathbf{C}_1,\ \mathbf{C}_2$, output $\mathbf{C}_{mult} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$, Obviously $\mathbf{t}\mathbf{C}_{mult} \approx u_1u_2\mathbf{t}\mathbf{G}$

Distributed decryption Similar to [MW16], the decryption procedure is a distributed protocol:

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

4.1 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 our weak-MKFHE scheme is the same as single-key homomorphic scheme: After the interactive key generation, participant p_i uses its own public key pk_i to encrypt s_i to obtain evaluation key evk_i, which is appended to the public key. Because evk_i and $\hat{\mathbf{C}}$ are both ciphertexts under $\mathbf{t} = (\sum_{i=1}^{k} \mathbf{s}_i, 1)$, homomorphic decryption can be computed directly when $\hat{\mathbf{C}}$ are need to be refresh. Therefore, in order to evaluate any depth circuit, we only need to set the initial parameters to satisfy the homomorphic evaluation of the decryption circuit.

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

4.2 Correctness analysis

In order to illustrate the correctness of our scheme, we first study the accumulation of noise:

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

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

After L depth circuit evaluation let $\mathbf{e}_L = (ml)^L \mathbf{e}_{init}$,

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

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

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

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

4.3 Security analysis

We first prove the semantic security of wMKFHE. Goldwasser et al. proved that the dual regev scheme is leakage resilient in [DGK+10], and similarly, Brakerski et al. [BHP17] proved that the Dual GSW scheme is leakage resilient. We prove the security by constructing a reduction from our scheme to the Dual GSW scheme. Consider the following game:

1. Challenger generates $pk_{Dual-GSW} = (\mathbf{A}, \mathbf{b}_1)$ where $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m-1\times n})$, $\mathbf{b}_1 = \mathbf{s}_1\mathbf{A}$, $\mathbf{s}_1 \leftarrow U\{0,1\}^m$ and send $pk_{Dual-GSW}$ to adversary $\mathcal{A}$
2. $\mathcal{A}$ generates $\{\mathbf{b}_i|\mathbf{b}_i = \mathbf{s}_i\mathbf{A}\}_{i\in[k]/1}$, choose a bit $u \in \{0,1\}$ and set $pk_{scheme1} = (\mathbf{A}, \mathbf{b})$, where $\mathbf{b} = \sum_{i=1}^{k} \mathbf{b}_i$, then send $pk_{scheme1}$, u to challenger.
3. Challenger choose a bit $\alpha \in \{0,1\}$, if $\alpha = 0$, set $\mathbf{C}_{scheme1} \leftarrow Scheme1.Enc(pk_{scheme1}, u)$, otherwise $\mathbf{C}_{scheme1} \leftarrow U(\mathbb{Z}_q^{m\times ml})$, and send $\mathbf{C}_{scheme1}$ to $\mathcal{A}$
4. After received $\mathbf{C}_{scheme1}$, $\mathcal{A}$ output bit $\bar{\alpha}$, if $\bar{\alpha} = \alpha$, then $\mathcal{A}$ wins.

Lemma 8. *Let $\mathcal{A}dv = |Pr[\bar{\alpha} = \alpha] - \frac{1}{2}|$ denote $\mathcal{A}$'s advantage in winning the game, If A can win the game with advantage* Adv*, then $\mathcal{A}$ can distinguish between the ciphertext distribution of Dual-GSW and the uniform random distribution with the same advantage.*

Proof. We construct $Scheme1.Enc(pk_{scheme1}, 0)$ by $Dual-GSW.Enc(pk_{Dual-GSW}, 0)$:

1. First, Challenger generates $pk_{Dual-GSW}$ like Game1, set $\mathbf{C}_{Dual-GSW} = Dual{-}GSW.Enc(pk_{Dual-GSW}, 0)$ send the both to $\mathcal{A}$.
2. $\mathcal{A}$ generates $\{\mathbf{s}_i\}_{i\in[k]/1}$, let $\mathbf{s}' = \sum_{i=2}^{k} \mathbf{s}_i$, $\mathbf{C}_{Dual-GSW} = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 \end{pmatrix}$, $\mathbf{c}'_1 = \mathbf{s}'\mathbf{C}_0$, $\mathbf{C}' = \begin{pmatrix} \mathbf{C}_0 \\ \mathbf{c}_1 + \mathbf{c}'_1 \end{pmatrix}$, obviously $\mathbf{C}' = \begin{pmatrix} \mathbf{A} \\ \mathbf{b} \end{pmatrix} \mathbf{R} + \begin{pmatrix} \mathbf{E}_0 \\ \mathbf{e}_1 + \mathbf{s}'\mathbf{E}_0 \end{pmatrix}$.

For our parameter settings $|\mathbf{e}_1| < 2^{\lambda^{\epsilon_1}} B_\chi$, $|\mathbf{s}'\mathbf{E}_0| < kmB_\chi$, thus $\mathbf{e}_1/\mathbf{s}'\mathbf{E}_0 = \mathsf{negl}(\lambda)$, we have $\mathbf{C}' \overset{\mathsf{stat}}{\approx} Scheme1.Enc(pk_{sheme1}, 0)$, if $\mathcal{A}$ can distinguish between $Scheme1.Enc(pk_{sheme1}, 0)$ and uniform random distribution by advantage Adv,then he can distinguish between $Dual-GSW.enc(0)$ and the uniform random distribution with the same advantage.

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

4.4 Simulatability of distributed decryption procedure

Similar to [MW16], we get a weak simulation of the distributed decryption procedure: input all private keys $\{sk_j\}_{j\in[k]/i}$ except sk_i, evaluated result u_{eval}, ciphertext $\hat{\mathbf{C}}$, we can simulate the local decryption result γ_i. For stronger security requirements : Input any private keys set $\{sk_j\}_{j\in S}$, S is any subset of $[k]$, evaluated result u_{eval} and ciphertext $\hat{\mathbf{C}}$, to simulate $\{\gamma_i\}_{i\in U,\ U=[k]-S}$, we don't know how to achieve it.

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

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

For simulator $\mathcal{S}$, input $\{sk_j\}_{j\in[k]/i}$, evaluated result u_{eval}, ciphertext $\hat{\mathbf{C}}$, output simulated γ'_i

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

For our parameter settings, we have :

$$|\sum_{i=1}^{k} e''_i| < k2^{L\lambda^{\epsilon_2}} B_\chi$$

$$e_{final} < kn(L\lambda)^{(2L+1)}(2^{\lambda^{\epsilon_2}} + k^2 nL\lambda)B_\chi = 2^{O(L\lambda^{\epsilon_1})} B_\chi$$

$$\text{thus } |e_{final}/\sum_{i=1}^{k} e''_i| = k2^{-\omega(L\lambda^{\epsilon_2} - L\lambda^{\epsilon_1})} = \mathsf{negl}(\lambda)$$

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

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

It is regrettable that the regularity lemma on the general ring cannot enjoy the leak resilient property of the leftover hash lemma on the integer ring $\mathbb{Z}$. This means that we cannot transplant the above construction process trivially to RLWE-based FHE. Indeed, [DSGKS21] pointed out that for $\mathbf{x} = (x_1 \dots x_l) \in R^l$, if the $j-th$ NTT coordinate of each $x_{i,i\in[l]}$ is leaked, then the $j-th$ NTT coordinate of $a_{l+1} = \sum_{i=1}^{l} a_i x_i$ is defined, thus a_{l+1} is far from random, although the leakage ratio is only 1/n. We also noticed a trivial solution: for $\mathbf{a}, \mathbf{s} \in R_q^l$, $b = \langle \mathbf{a}, \mathbf{s} \rangle \in R_q$, b leaks information about $\mathbf{s}$ at most $n \log q$ bits, therefore, as long as we set l long enough, for example, $l = l + n \log q$, then obviously b is close to uniformly random, but this will result in a extremely large key, thus it is not practical.

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

Lemma 9. *For a given $\delta \in \{0,1\}^\lambda$, k probabilistic polynomial time(ppt) adversary $\mathcal{A}$, Each $\mathcal{A}$ makes $q_{ro} = \mathsf{poly}(\lambda)$ queries to H, let $\overline{\mathsf{Adv}}$ denotes the probability of finding a collision, then: $\overline{\mathsf{Adv}} = \mathsf{negl}(\lambda)$*

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

Key generation with one round bit commitment.

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

1. $params \leftarrow Init(1^\lambda, 1^L)$:Input security parameter λ, circuit depth L, output $params = (d, q, \chi, B_\chi)$, which χ is an noise distribution over $R : \mathbb{Z}[x]/x^d + 1$, satisfying $e \leftarrow \chi$, $|e|_\infty^{can}$ is bounded by B_χ, and $RLWE_{d,q,\chi,B_\chi}$ is infeasible.
2. p_i generates $a_i \leftarrow U(R_q)$, $\mathbf{d}_i \leftarrow U(R_q^l)$, $\mathbf{f}_i \leftarrow U(R_q^l)$, computes $\delta_i = H(a_i)$, $\epsilon_i = H(\mathbf{d}_i)$, $\zeta_i = H(\mathbf{f}_i)$ and broadcast δ_i, ϵ_i, ζ_i
3. After all $\{\delta_i,\ \epsilon_i,\ \zeta_i\}_{i\in[k]}$ are public, p_i discloses $\{a_i,\ \mathbf{d}_i,\ \mathbf{f}_i\}$.
4. After receiving $\{a_j, \mathbf{d}_j, \mathbf{f}_j\}_{j\in[k]/i}$, p_i broadcast $\{b_i,\ \mathbf{h}_i\}$, where $b_i = as_i + e_1$, $\mathbf{h}_i = \mathbf{d}s_i + \mathbf{e}_2$, $a = \sum_{i=1}^k a_i,\ \mathbf{d} = \sum_{i=1}^k \mathbf{d}_i,\ (s_i,\ e_1,\ \mathbf{e}_2) \leftarrow \chi^{l+2}$.

   After receiving $\{b_j,\ \mathbf{h}_j\}_{j\in[k]/i}$, p_i output $pk_i = (a, b)$ and $EvalKey_i = (\mathbf{h}_i, \eta_i, \theta_i)$

$$b = \sum_{i=1}^{k} b_i \qquad \eta_i = \mathbf{d}r_i + \mathbf{e}_3 + s_i\mathbf{g}$$
$$\theta_i = \mathbf{f}s_i + \mathbf{e}_4 + r_i\mathbf{g} \qquad (r_i,\ \mathbf{e}_3,\ \mathbf{e}_4) \leftarrow \chi^{2l+1}$$

Re-linearization ciphertext

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

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

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

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

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

$\mathbf{c}_{linear} \leftarrow Relinear(\mathbf{c}_{mult},\ \{Evalkey_i\}_{i\in[k]})$: Input $\mathbf{c}_{mult} \in R_q^4$, evaluation key $\{Evalkey_i\}_{i\in[k]}$, perform Relinear as follows, output $\mathbf{c}_{linear} = (c_0', c_1')$. Due to the sum structure of keys, the dimension of $\mathbf{t} \otimes \mathbf{t}$ is independent of participants k, thus above algorithm pulls the tensor product ciphertext back to initial dimension by one shot, and introduces less noise than those keys with concatenation structure.

Ciphertext Relinearization

Input: $\mathbf{c}_{mult} = (c_0, c_1, c_2, c_3) \in R_q^4,\ \{Evalkey_i\}_{i\in[k]} = \{\mathbf{h}_i,\ \eta_i,\ \theta_i\}_{i\in[k]}$
Output: $\mathbf{c}_{linear} = (c_0', c_1') \in R_q^2$
1: $\eta \leftarrow \sum_{i=1}^{k} \eta_i,\ \theta \leftarrow \sum_{i=1}^{k} \theta_i,\ \mathbf{h} \leftarrow \sum_{i=1}^{k} \mathbf{h}_i$
2: $\mathbf{k}_0 \leftarrow -\theta\mathbf{g}^{-1}(\mathbf{h}),\ \mathbf{k}_1 \leftarrow \eta + \mathbf{f}\mathbf{g}^{-1}(\mathbf{h})$
3: $c_0' \leftarrow c_0 + \mathbf{k}_0\mathbf{g}^{-1}(c_3),\ c_1' \leftarrow c_1 + c_2 + \mathbf{k}_1\mathbf{g}^{-1}(c_3)$
4: **Output:** (c_0', c_1')
5: **End**.

6 Conclusions

For the LWE-based multi-key homomorphism scheme, in order to alleviate the overhead of the local participants, we proposed the concept of weak-MKFHE, combining the methods of [BHP17] and [AJL+12] to construct a Dual GSW style weak-MKFHE under the plain model. Our *Scheme*#1 is more friendly to local participants than previous scheme, since there is no ciphertext expansion. However, to support semantic security and threshold decryption, module q is required to be $O(2^{\lambda L})$, such a large q results in high overhead of ciphertext evaluation. Reducing q while ensuring security is the future direction.

For the multi-key homomorphic scheme based on RLWE, although the computation overhead of the local participants is not large: to perform re-linearization, only one ring element needs to be encrypted, but the common random string is always an insurmountable hurdle. Constructing RLWE-type MKFHE under plain model is the future direction.

References

AJL+12. G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *EUROCRYPT 2012*, *LNCS* 7237, pages 483–501. Springer, Heidelberg, April 2012.

AP14. J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *CRYPTO 2014, Part I*, *LNCS* 8616, pages 297–314. Springer, Heidelberg, August 2014.

BGV12. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012*, pages 309–325. ACM, January 2012.

BHP17. Z. Brakerski, S. Halevi, and A. Polychroniadou. Four round secure computation without setup. In *TCC 2017, Part I*, *LNCS* 10677, pages 645–677. Springer, Heidelberg, November 2017.

BP16. Z. Brakerski and R. Perlman. Lattice-based fully dynamic multi-key FHE with short ciphertexts. In *CRYPTO 2016, Part I*, *LNCS* 9814, pages 190–213. Springer, Heidelberg, August 2016.

CDKS19. H. Chen, W. Dai, M. Kim, and Y. Song. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In *ACM CCS 2019*, pages 395–412. ACM Press, November 2019.

CGGI16. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT 2016, Part I*, *LNCS* 10031, pages 3–33. Springer, Heidelberg, December 2016.

CKKS17. J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT 2017, Part I*, *LNCS* 10624, pages 409–437. Springer, Heidelberg, December 2017.

CM15. M. Clear and C. McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In *CRYPTO 2015, Part II*, *LNCS* 9216, pages 630–656. Springer, Heidelberg, August 2015.

DGK+10. Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC 2010*, *LNCS* 5978, pages 361–381. Springer, Heidelberg, February 2010.

DSGKS21. D. Dachman-Soled, H. Gong, M. Kulkarni, and A. Shahverdi. Towards a ring analogue of the leftover hash lemma. *Journal of Mathematical Cryptology*, 15(1):87–110, 2021.

FV12. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. https://eprint.iacr.org/2012/144.

Gen09a. C. Gentry. *A fully homomorphic encryption scheme*. Stanford university, 2009.

Gen09b. C. Gentry. Fully homomorphic encryption using ideal lattices. In *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

GSW13. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013, Part I*, *LNCS* 8042, pages 75–92. Springer, Heidelberg, August 2013.

ILL89. R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

Lin13. Y. Lindell. Fast cut-and-choose based protocols for malicious and covert adversaries. In *CRYPTO 2013, Part II*, *LNCS* 8043, pages 1–17. Springer, Heidelberg, August 2013.

LP07. Y. Lindell and B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *EUROCRYPT 2007*, *LNCS* 4515, pages 52–78. Springer, Heidelberg, May 2007.

LPR10. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT 2010*, *LNCS* 6110, pages 1–23. Springer, Heidelberg, May / June 2010.

LPR13. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-lwe cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 35–54. Springer, 2013.

LTV12. A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *44th ACM STOC*, pages 1219–1234. ACM Press, May 2012.

MGW87. S. Micali, O. Goldreich, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*, pages 218–229. ACM, 1987.

MP12. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, *LNCS* 7237, pages 700–718. Springer, Heidelberg, April 2012.

MTBH21. C. Mouchet, J. R. Troncoso-Pastoriza, J.-P. Bossuat, and J.-P. Hubaux. Multiparty homomorphic encryption from ring-learning-with-errors. *PoPETs*, 2021(4):291–311, October 2021.

MW16. P. Mukherjee and D. Wichs. Two round multiparty computation via multi-key FHE. In *EUROCRYPT 2016, Part II*, *LNCS* 9666, pages 735–763. Springer, Heidelberg, May 2016.

NNOB12. J. B. Nielsen, P. S. Nordholt, C. Orlandi, and S. S. Burra. A new approach to practical active-secure two-party computation. In *CRYPTO 2012*, *LNCS* 7417, pages 681–700. Springer, Heidelberg, August 2012.

Orl11. C. Orlandi. Is multiparty computation any good in practice? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5848–5851. IEEE, 2011.

PS16. C. Peikert and S. Shiehian. Multi-key fhe from lwe, revisited. In *Theory of Cryptography Conference*, pages 217–238. Springer, 2016.

PSSW09. B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure two-party computation is practical. In *ASIACRYPT 2009*, *LNCS* 5912, pages 250–267. Springer, Heidelberg, December 2009.

RAD+78. R. L. Rivest, L. Adleman, M. L. Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

Reg05. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

RSA78. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

SS11. D. Stehlé and R. Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 27–47. Springer, 2011.

vGHV10. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT 2010*, *LNCS* 6110, pages 24–43. Springer, Heidelberg, May / June 2010.

Yao82. A. C.-C. Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982.

Yao86. A. C.-C. Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.