IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 March 2024, accepted 29 March 2024, date of publication 2 April 2024, date of current version 11 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384247

RESEARCH ARTICLE

Concise and Efficient Multi-Identity Fully Homomorphic Encryption Scheme

GUANGSHENG TU[1], WENCHAO LIU[2,3], TANPING ZHOU[2,3], XIAOYUAN YANG[2,3], AND FAN ZHANG[1]

[1]Non-Commissioned Officer Academy of People's Armed Police, Hangzhou 311400, China
[2]Key Laboratory of Network and Information Security of People's Armed Police, Xi'an 710086, China
[3]School of Cryptographic Engineering, Engineering University of People's Armed Police, Xi'an 710086, China

Corresponding author: Xiaoyuan Yang (tugs@ldy.edu.rs)

This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB3106100 and Grant 2021YFB3100100, in part by the National Natural Science Foundation of China under Grant 62172436 and Grant 62102452, and in part by the Natural Science Foundation of Shaanxi Province under Grant 2023-JC-YB-584.

ABSTRACT Combining multi-key fully homomorphic encryption (MKFHE) and identity-based encryption (IBE) to construct multi-identity based fully homomorphic encryption (MIBFHE) scheme can not only realize homomorphic operations and flexible access control on identity ciphertexts but also reduce the burden of public key certification management. However, MKFHE schemes used to construct MIBFHE usually have complex construction and large computational complexity, which also causes the same problem for MIBFHE schemes. To solve this problem, we construct a concise and efficient MIBFHE scheme based on the learning with errors (LWE) problem. Firstly, we construct an MKFHE scheme using a new method called ''the decomposition method''. Secondly, we make a suitable deformation of the current IBE scheme. Finally, we combine the above MKFHE scheme with IBE scheme to construct our MIBFHE scheme and prove its IND-sID-CPA security under the LWE assumption in the random oracle model. The analysis results show that our MIBFHE scheme can generate the extended ciphertext directly from the encryption algorithm, without generating fresh ciphertext in advance. In addition, the noise expansion rate is reduced from the polynomial of lattice dimension n and modulus q to the constant K of the maximum number of users. The scale of introduced auxiliary ciphertexts is reduced from $\tilde{O}(n^4L^4)$ to $\tilde{O}(n^2L^4)$ when generating the extended ciphertext.

INDEX TERMS Multi-key, multi-identity, fully homomorphic encryption, identity-based encryption.

I. INTRODUCTION

Cloud computing can link a large number of computing, storage and software resources together to form a large scale of shared virtual IT pools, providing remote computer users with efficient, fast and diverse information services. However, cloud computing also has to face security problems, which has become an important reason restricting its development.

In 1978, Rivest et al. [1] first proposed the notion of ''privacy homomorphism'' which allows ciphertexts to be operated on without preliminary decryption of the operands. This property can be naturally applied to solve the security problem of cloud computing, and how to construct a fully homomorphic encryption (FHE) scheme has become an open issue in cryptography. In 2009, Gentry [2] proposed the first FHE scheme based on ideal lattice, and made it bootstrappable by using the technique of ''squash the decryption circuit''. However, such FHE schemes [1], [2], [3], [4] can only perform homomorphic operations on ciphertexts encrypted under the same public key. In 2012, López-Alt et al. [5] first proposed the notion of multi-key FHE (MKFHE) and constructed an MKFHE scheme based on the NTRU cryptosystem. Cryptographers have proposed a series of MKFHE schemes [6], [7], [8], [9], [10], [11], [12]. As an extension of FHE, MKFHE allows homomorphic operations on ciphertexts encrypted under different public keys. However, the

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

public key encryption (PKE) system needs to rely on public key certificates to realize identity authentication, which inevitably increases the overhead related to certificate management, especially FHE and MKFHE have a large public key size.

Identity-based encryption (IBE) is another cryptographic primitive that was first proposed by Shamir [13] in 1984. Its central idea is to generate a pair of master public key (*MPK*) and master secret key (*MSK*) from a trusted authority. User's public key is then generated by the unique identity strings and *MPK*, without the need to determine and manage the public key for each user. The corresponding secret key is generated by the private key generator (PKG) from the identity strings and *MSK*. In 2008, Gentry et al. [14] constructed the first IBE scheme based on the learning with errors (LWE) in the random oracle model. Compared with PKE, IBE eliminates the calculation, distribution, storage, update, management and other operations related to public key certificates, and manages keys more effectively.

According to the above analysis, combining FHE and MKFHE with IBE to construct the identity-based FHE (IBFHE) scheme and multi-identity based FHE (MIBFHE) scheme seems to be beneficial. There are also many related results [3], [6], [15], [16], [17], [18], [19], [20], [21], [22].

About IBFHE. In 2013, Gentry et al. (GSW scheme) [3] described a leveled FHE scheme based on LWE problem with the approximate eigenvector method. Using this FHE scheme and the IBE scheme satisfying specific properties, they also constructed the first IBFHE scheme without breaking the non-interactivity. However, this IBFHE scheme can only perform homomorphic operations on ciphertexts encrypted under the same identity.

About MIBFHE. In 2014, Clear and McGoldrick [15] constructed an MIBFHE scheme based on the indistinguishability obfuscation (IO) [23]. The next year, they obtained a new MIBFHE scheme [6] based on LWE problem and demonstrated its selective security in the random oracle model. In 2017, Canetti et al. [16] used the MIBFHE scheme to construct three FHE schemes that were secure against non-adaptive chosen ciphertext attacks (CCA1) and gave two instantiated MIBFHE schemes. However, both MIBFHE schemes have weak compactness, that is, the size of the evaluated ciphertext depends on the size of the computing circuits, but is independent of the number of users involved in the computation. In 2018, Wang et al. [17] constructed a more compact MIBFHE scheme compared to [16] using IO and witness pseudorandom function (WPRF) which is not a standard assumption. In 2019, Shen et al. [18] used the MKFHE scheme proposed by Mukherjee and Wichs [7] and the IBE scheme proposed by Micciancio and Peikert [24] to construct an efficient MIBFHE scheme. In 2020, Pal and Dutta [19] extended IBE to a CCA1 secure MIBFHE scheme using WPRF. In 2021, Shen et al. [20] constructed a compressible MIBFHE scheme using a new compressible ciphertext extension technique. In 2022, Liu et al. [21] proposed a leveled multi-hop MIBFHE scheme based on GPV-FHE scheme in [14] and [25] and MKFHE scheme in [8]. This is a multi-hop MIBFHE scheme in which the resulting ciphertexts encrypted under a set of identities can be further evaluated with other ciphertexts encrypted under additional identities. In 2023, Fan et al. [22] proposed an efficient MIBFHE scheme on lattice. They improved the IBE scheme proposed by Agrawal et al. [26] using the transformation mechanism of [24] and reconstructed a Link-Mask system based on the MKFHE scheme in [7]. However, both schemes have the problem of complex construction and large computational complexity.

In summary, the FHE and MKFHE developed from the GSW scheme [3] eliminate the evaluation key without using the bootstrapping technique and can be combined with IBE to construct IBFHE and MIBFHE scheme. Whereas, MKFHE schemes commonly used to construct MIBFHE schemes have complex construction and large computational complexity, which leads to the same problem in the MIBFHE scheme. Compared with IBE, the main factors restricting the construction and efficiency of IBFHE and MIBFHE are FHE and MKFHE. Current research on MIBFHE mainly focuses on the optimization of IBE, while the optimization of MKFHE is less and the effect is not obvious. Therefore, it is necessary to start from the optimization of MKFHE, and then construct a more concise and efficient MIBFHE scheme. In this paper, we focus on the leveled FHE and MKFHE developed from the GSW scheme [3], so we omit the term ''leveled'' in the context.

A. OUR CONTRIBUTION

To solve the above problems, we want to construct an MIBFHE scheme, so the scheme construction is relatively simple and the computational complexity can be significantly reduced. We make the contributions as follows:

1) A new MKFHE scheme. We construct an MKFHE scheme using a new method called ''the decomposition method''. In our MKFHE scheme, users can directly generate the extended ciphertext for homomorphic operations through the encryption algorithm, without generating the fresh ciphertext in advance. The noise expansion rate is significantly reduced from $poly(n, \ell)$ to 2, and the scale of auxiliary ciphertexts is reduced from $\tilde{O}(n^4L^4)$ to $\tilde{O}(n^2L^2)$.
2) A deformation of the IBE scheme in [22]. We introduce a hash function and adjust the construction form of the master public key matrix. The parameters, safety and efficiency of the IBE scheme remain unchanged.
3) An MIBFHE scheme. Using the above MKFHE and IBE scheme, we construct our MIBFHE scheme.
    - No fresh ciphertext is needed in advance. Inheriting the advantages of our MKFHE scheme, our MIBFHE scheme can also directly generate the extended ciphertext, without generating the fresh ciphertext in advance.

- Smaller noise growth and public parameters. The noise expansion rate of our scheme is reduced from $poly(n, \ell)$ to K, where K is determined in advance. The scale of introduced auxiliary ciphertexts is reduced from $\tilde{O}(n^4L^4)$ to $\tilde{O}(n^2L^4)$ when generating the extended ciphertext. Therefore, our scheme has smaller public parameters.

B. TECHNICAL OVERVIEW

We construct an MKFHE scheme and then construct an MIBFHE scheme based on it. The efficiency of MKFHE scheme directly determines the efficiency of MIBFHE scheme. Here, we show the technical overview of constructing MKFHE scheme. In contrast, we call the previous method of constructing MKFHE scheme ''the integral method'' and the method of constructing our MKFHE scheme ''the decomposition method''.

The integral method. In previous MKFHE schemes [6], [7], [8], [9], [10], [11], [12], a fresh ciphertext $\boldsymbol{C}$(usually a matrix) needs to be generated by the encryption algorithm, and then the fresh ciphertext $\boldsymbol{C}$ can be transformed into the extended ciphertext $\hat{\boldsymbol{C}}$ (usually a block matrix) by the ciphertext extension algorithm. The structure of the extended ciphertext is similar and can be defined as

$$\hat{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C}_i & \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_i & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{D}_1 & \boldsymbol{D}_2 & \dots & \boldsymbol{C}_i & \dots & \boldsymbol{D}_K \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{C}_i \end{bmatrix} \tag{1}$$

or

$$\hat{\boldsymbol{C}} = \begin{pmatrix} \boldsymbol{C}_i & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{F} \end{pmatrix} \tag{2}$$

where the matrices $\boldsymbol{D}$ and $\boldsymbol{F}$ are auxiliary ciphertexts. The common thought to achieve ciphertext extension in current schemes is to regard the fresh ciphertext as a whole and as parts of the extended ciphertext. The difference lies in different ways of constructing auxiliary ciphertexts $\boldsymbol{D}$ and $\boldsymbol{F}$. We call this method ''the integral method''.

The decomposition method. Now, we think about the ciphertext extension in a different way. We use the variant FHE in [3], and the fresh ciphertext can be defined as $\boldsymbol{C} = \boldsymbol{A} \cdot \boldsymbol{R} + \mu \boldsymbol{G}_n$. We decompose the fresh ciphertext into two parts. The first part is the combination of public key $\boldsymbol{A}$ and randomness matrix $\boldsymbol{R}$, and the second part is the combination of plaintext μ and $\boldsymbol{G}_n = \boldsymbol{I}_n \otimes \boldsymbol{g}$. Analysis of the decryption of fresh ciphertext $\boldsymbol{tC} = \boldsymbol{tAR} + \mu \boldsymbol{tG}_n = \mu \boldsymbol{tG}_n + \boldsymbol{eR}$, we know that the second part of the fresh ciphertext can be directly extended, that is, $\mu \boldsymbol{G}_n \to \mu \boldsymbol{G}_{Kn}$. Correspondingly, the decryption process also changes from $\mu \boldsymbol{tG}_n$ to $\mu \hat{\boldsymbol{t}} \boldsymbol{G}_{Kn}$. Assuming that the first part of the fresh ciphertext also has the same property when it is decrypted, that is, $\boldsymbol{e} \to \hat{\boldsymbol{e}}$, then the extended ciphertext can also be correctly decrypted. The method of ciphertext extension also changes from the whole expansion $\boldsymbol{C} \to \hat{\boldsymbol{C}}$ to the separate expansion of the first part and the second part of the fresh ciphertext, and the addition of the two parts will naturally get the extended ciphertext we want. We call this method ''the decomposition method''.

Instance. To be specific, the fresh ciphertext of user i, i=$\{1,2\}$ is $\boldsymbol{C}_i = \boldsymbol{A}_i \boldsymbol{R}_i + \mu_i \boldsymbol{G}_n \in \mathbb{Z}_q^{n \times m}$. Assume $\hat{\boldsymbol{t}}_{1,2} = (\boldsymbol{t}_1, \boldsymbol{t}_2)$ is the concatenation of two secret keys. When constructing the MKFHE scheme using ''the integral method'', the fresh ciphertext needs to be extended and the extended ciphertext can be expressed as $\hat{\boldsymbol{C}}_i = \begin{pmatrix} \boldsymbol{C}_i & \boldsymbol{D}_i \\ \boldsymbol{0}^{n \times m} & \boldsymbol{F}_i \end{pmatrix} \in \mathbb{Z}_q^{2n \times 2m}$. Using ''the decomposition method'', the extended ciphertext can also be expressed as $\hat{\boldsymbol{C}}_i = \boldsymbol{X}_i + \boldsymbol{Y}_i \in \mathbb{Z}_q^{2n \times 2m}$ in two steps.

- Constructing $\boldsymbol{X}_i$, we get $\boldsymbol{X}_i$ satisfying $\hat{\boldsymbol{t}}_{1,2} \cdot \boldsymbol{X}_i = \hat{\boldsymbol{e}}_{1,2} \in \mathbb{Z}_q^{2m}$;
- Constructing $\boldsymbol{Y}_i$, we get $\boldsymbol{Y}_i = \mu_i \boldsymbol{G}_{2n}$ satisfying $\hat{\boldsymbol{t}}_{1,2} \cdot \boldsymbol{Y}_i = \mu_i \hat{\boldsymbol{t}}_{1,2} \boldsymbol{G}_{2n}$.

So we get the extended ciphertext

$$\begin{aligned} \hat{\boldsymbol{C}}_i &= \boldsymbol{X}_i + \boldsymbol{Y}_i \\ &= \boldsymbol{X}_i + \mu_i \boldsymbol{G}_{2n} \in \mathbb{Z}_q^{2n \times 2m} \end{aligned} \tag{3}$$

satisfying $\hat{\boldsymbol{t}}_{1,2} \hat{\boldsymbol{C}}_i = \mu_i \hat{\boldsymbol{t}}_{1,2} \boldsymbol{G}_{2n} + \hat{\boldsymbol{e}}_{1,2} \in \mathbb{Z}_q^{2m}$.

Comparison. When comparing ''the integral method'' with ''the decomposition method'', the simplicity of our scheme is demonstrated. In ''the integral method'' process, to ensure the correctness of decryption, it needs to be satisfied

$$\begin{aligned} \hat{\boldsymbol{t}}_{1,2} \cdot \hat{\boldsymbol{C}} &= (\boldsymbol{t}_1, \boldsymbol{t}_2) \begin{pmatrix} \boldsymbol{C} & \boldsymbol{D} \\ \boldsymbol{0} & \boldsymbol{F} \end{pmatrix} \\ &= (\boldsymbol{t}_1 \boldsymbol{C}, \boldsymbol{t}_1 \boldsymbol{D} + \boldsymbol{t}_2 \boldsymbol{F}) \\ &= (\mu \boldsymbol{t}_1 \boldsymbol{G}_n + \boldsymbol{e}_1, \boldsymbol{t}_1 \boldsymbol{D} + \boldsymbol{t}_2 \boldsymbol{F}) \\ &= \mu \hat{\boldsymbol{t}}_{1,2} \boldsymbol{G}_{2n} + (\boldsymbol{e}_1, \boldsymbol{e}_2) \end{aligned} \tag{4}$$

We get

$$\boldsymbol{t}_1 \boldsymbol{D} + \boldsymbol{t}_2 \boldsymbol{F} = \mu \boldsymbol{t}_2 \boldsymbol{G}_n + \boldsymbol{e}_2 \tag{5}$$

As mentioned above, using ''the decomposition method'', we only need to satisfy

$$\begin{aligned} \hat{\boldsymbol{t}}_{1,2} \cdot \hat{\boldsymbol{C}}_i &= \hat{\boldsymbol{t}}_{1,2} (\boldsymbol{X}_i + \boldsymbol{Y}_i) \\ &= \hat{\boldsymbol{t}}_{1,2} \boldsymbol{X}_i + \mu_i \hat{\boldsymbol{t}}_{1,2} \boldsymbol{G}_{2n} \\ &= \mu_i \hat{\boldsymbol{t}}_{1,2} \boldsymbol{G}_{2n} + \hat{\boldsymbol{e}}_{1,2} \end{aligned} \tag{6}$$

We get

$$\hat{\boldsymbol{t}}_{1,2} \cdot \boldsymbol{X}_i = \hat{\boldsymbol{e}}_{1,2} \tag{7}$$

Compared with (5) and (7), it can be seen that it is easier to construct auxiliary ciphertexts by using our method.

Interestingly, using ''the decomposition method'' to construct our MKFHE scheme, the fresh ciphertext is no longer a part of the extended ciphertext, and we can directly generate the extended ciphertext that can perform homomorphic operations without generating the fresh ciphertext. This makes our scheme more concise and allows users to perform fewer operations.

C. ORGANIZATION

The remainder of this paper is organized as follows. In Section II, we introduce the preliminaries of MKFHE and MIBFHE. In Section III, we proposed our MKFHE scheme using ''the decomposition method''. In Section IV, we described the IBE scheme after deformation. In Section V, we proposed our MIBFHE scheme. We conclude the paper in Section VI.

II. PRELIMINARIES

A. NOTION

We define some notations that will be used throughout this paper in Table 1.

TABLE 1. Symbol specification.

Symbol	Meaning
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{P}, \boldsymbol{R}, \boldsymbol{X}, \boldsymbol{Y}$	matrix
$\boldsymbol{b}, \boldsymbol{e}, \boldsymbol{g}, \boldsymbol{p}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{u}, \boldsymbol{x}$	vector
$m, n, r, q, B, K, L, N, \alpha, \beta, \bar{m}, \ell, \sigma$,	variable
$negl, poly, Pr, Adv, \boldsymbol{g}^{-1}, \boldsymbol{G}^{-1}, H$	function
$\{D_n\}_{n\in\mathbb{N}}, \chi$	probability distribution
$\mathbb{N}, \mathbb{R}, \mathbb{Z}$	set of numbers
$\mathbb{Z}_q$	a finite field of module q
$\mathbb{Z}_q^n$	n-dimensional vector over $\mathbb{Z}_q$
L, Λ	lattice
$\Lambda(\boldsymbol{B})$	lattice based on $\mathbf{B}$
$[\boldsymbol{A} \Vert \boldsymbol{B}], (\boldsymbol{a}, \boldsymbol{b})$	horizontal connection of matrices or vectors
$\Theta(\cdot), \tilde{O}(\cdot), \omega(\cdot)$	polynomial coefficient
$\boldsymbol{A}^{-1}$	inverse of matrix
$\boldsymbol{A}^{\mathrm{T}}$	transpose of the matrix
$\langle \boldsymbol{a}, \boldsymbol{s} \rangle$	dot product of vectors
FHE, IBE, LWE, MKFHE, IBFHE, MIBFHE, PP, *MPK*, *MSK*, IND-CPA, IND-sID-CPA, PPT, FRD, PKE, PKG, VR	the abbreviation that appears in the text

We also gave definitions for some basic notions that will be used throughout this paper.

Definition 1 ([3] Negligible Function): Suppose n represents the input size of the algorithm and $negl(n)$ is a constant with a variable value. $Negl(n)$ is said to be negligible if for any polynomial $poly(n)$ there exists an integer n such that, for $n \geq N$, the formula $negl(n) \leq 1/poly(n)$ always holds.

Definition 2 ([3] B-Bounded Distributions): Suppose $\{D_n\}_{n\in\mathbb{N}}$ denotes a distribution over the integers, the distribution is called B-bounded if $Pr_{e\leftarrow D_n}[|e| > B] = negl(n)$.

Definition 3 (([7]β-Noisy Ciphertext)): There is a β-noisy ciphertext $\boldsymbol{C}$ encrypting μ under secret key $\boldsymbol{t}$ such that $\boldsymbol{tC} = \mu \boldsymbol{tG}_n + \boldsymbol{e}$ for $\|\boldsymbol{e}\|_\infty \leq \beta$.

Definition 4 ([24] Gadget Matrix): The gadget notion is used for decomposing $\mathbb{Z}_q$-elements into short vectors over $\mathbb{Z}$. For simplicity, throughout this work, we use the standard form as $\boldsymbol{g} = (2^0, 2^1, \ldots, 2^{\ell-1}) \in \mathbb{Z}_q^\ell$, where $\ell = \lceil \log q \rceil$. We define a computable randomized function $\boldsymbol{g}^{-1} : \mathbb{Z}_q \to \mathbb{Z}^\ell$ such that for any $\boldsymbol{x} \leftarrow \boldsymbol{g}^{-1}(a)$, $\langle \boldsymbol{g}, \boldsymbol{x} \rangle = a$ holds. Extending the above properties from vectors to matrices, we define $\boldsymbol{G}_n = \boldsymbol{I}_n \otimes \boldsymbol{g}$ where $\boldsymbol{I}_n$ is the identity matrix of dimension n. We also define a computable randomized function $\boldsymbol{G}_n^{-1} : \mathbb{Z}_q^{n\times m} \to \mathbb{Z}_q^{n\ell\times m}$ such that for any $\boldsymbol{X} = \boldsymbol{G}_n^{-1}(\boldsymbol{A})$, $\boldsymbol{G} \cdot \boldsymbol{X} = \boldsymbol{A}$ holds.

B. BACKGROUND ON LATTICES AND TRAPDOORS

Definition 5 ([27] Lattice): Assume $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m \in \mathbb{R}^n (m \geq n)$ is a set of linearly independent vectors, the set of points formed by integer linear combinations of these vectors is called a lattice, denoted by $\Lambda(\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m)$, which can be defined as

$$\Lambda(\boldsymbol{B}) = \left\{ \boldsymbol{y} \in \mathbb{R}^n : \exists \boldsymbol{s} \in \mathbb{Z}^m, \boldsymbol{y} = \boldsymbol{Bs} = \sum\nolimits_{i=1}^{m} \boldsymbol{s}_i \boldsymbol{b}_i \right\} \quad (8)$$

where $\boldsymbol{B} = [\boldsymbol{b}_1 \| \boldsymbol{b}_2 \| \cdots \| \boldsymbol{b}_m]$ is called a basis of the lattice $\Lambda(\boldsymbol{B})$.

Definition 6 ([27], [28] q-Module Lattice): Let $n, m \in \mathbb{Z}$, q be a prime, $m = O(n \log q)$. For matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, we define

$$\Lambda_q(\boldsymbol{A}) = \left\{ \boldsymbol{y} \in \mathbb{Z}_q^m : \exists \boldsymbol{s} \in \mathbb{Z}_q^n, \boldsymbol{y} = \boldsymbol{A}^T \boldsymbol{s} (\mathrm{mod} q) \right\} \quad (9)$$

$$\Lambda_q^\perp(\boldsymbol{A}) = \left\{ \boldsymbol{y} \in \mathbb{Z}_q^m : \boldsymbol{Ay} = \boldsymbol{0} (\mathrm{mod} q) \right\} \quad (10)$$

$$\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \left\{ \boldsymbol{y} \in \mathbb{Z}_q^m : \boldsymbol{Ay} = \boldsymbol{u} (\mathrm{mod} q) \right\} \quad (11)$$

where $\Lambda_q^\perp(\boldsymbol{A})$ is the q-module integer lattice and $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$ is the coset of $\Lambda_q^\perp(\boldsymbol{A})$.

Definition 7 ([24] Trapdoor Function): Let $n \geq 2$ be an integer, $q \geq 2$ be an integer, assume parameters $m = O(n \log q)$, $\ell = \lceil \log q \rceil$, $\bar{m} = m - n\ell$. For matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, $\boldsymbol{G}_n = (\boldsymbol{I}_n \otimes \boldsymbol{g})$ and invertible matrix $\boldsymbol{H} \in \mathbb{Z}_q^{n\times n}$, the corresponding $\boldsymbol{G}$-trapdoor matrix of $\boldsymbol{A}$ is $\boldsymbol{R} \in \mathbb{Z}^{\bar{m}\times n\ell}$ which satisfies $\boldsymbol{A} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{I}_{n\ell} \end{bmatrix} = \boldsymbol{HG}$. We use the maximum singular value of the matrix $\boldsymbol{R}$ which is represented by $s(\boldsymbol{R})$ to measure the quality of a trapdoor.

Lemma 1 ([24] Trapdoor Generation Algorithm): Let parameters $n, m, q, \bar{m}, \ell, \boldsymbol{H}, \boldsymbol{G}$ be chosen as Definition 7. Choose a randomly uniform matrix $\bar{\boldsymbol{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$. Then there is a probabilistic polynomial time (PPT) algorithm $TrapGen(1^n, 1^m, q)$, outputs matrix $\boldsymbol{A} = \left[\bar{\boldsymbol{A}} \middle\| (\boldsymbol{HG} - \bar{\boldsymbol{A}}\bar{\boldsymbol{R}})\right] \in \mathbb{Z}_q^{n\times m}$ and its trapdoor matrix $\boldsymbol{R} \in \mathbb{Z}^{\bar{m}\times n\ell}$, where matrix $\boldsymbol{A}$ is statistically close to the uniform matrix in $\mathbb{Z}_q^{n\times m}$ and $s(\boldsymbol{R}) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$.

Lemma 2 ([24] Sampling Algorithm): Let parameters $n, m, q, \bar{m}, \ell, \boldsymbol{A}, \boldsymbol{R}$ be chosen as Lemma 1. Let χ be a discrete Gaussian distribution with parameters $\sigma \geq s(\boldsymbol{R}) \cdot \omega(\sqrt{\log n})$, choose a random vector $\boldsymbol{u} \in \mathbb{Z}_q^n$. There exists a PPT algorithm $SampleD(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{u}, \sigma)$, outputs $\boldsymbol{t} \in \mathbb{Z}_q^m$ satisfying $\boldsymbol{A} \cdot \boldsymbol{t} = \boldsymbol{u} \bmod q$. And the vector $\boldsymbol{t}$ is close

to the discrete Gaussian distribution $\chi_{\Lambda_q^u(\boldsymbol{A}),\sigma\cdot\omega(\sqrt{\log n})}$ as $Pr\left[\boldsymbol{t} \leftarrow \chi_{\Lambda_q^u(\boldsymbol{A}),\sigma\cdot\omega(\sqrt{\log n})} : \boldsymbol{t} \geq \sigma\sqrt{m}\right] \leq negl(n)$.

C. LEARNING WITH ERRORS

Definition 8 ([28] LWE): For security parameter $\lambda \in \mathbb{N}$, let $q = q(\lambda) \geq 2$ be the module, let $n \geq 1$ be an integer dimension, and let $\chi = \chi(\lambda)$ be the Gaussian distribution over $\mathbb{Z}$. Randomly selected $\boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e_i \leftarrow \chi$, output $(\boldsymbol{a}, b_i = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e_i) \in \mathbb{Z}_q^{n+1}$. The LWE problem is to solve $\boldsymbol{s}$ from a given set of instances of $(\boldsymbol{a}, b_i)$, also known as the computational LWE problem.

Definition 9 ([28] Decisional LWE Problem): For security parameter $\lambda \in \mathbb{N}$, let $q = q(\lambda) \geq 2$ be the module, let $n \geq 1$ be an integer dimension, and let $\chi = \chi(\lambda)$ be the Gaussian distribution over $\mathbb{Z}$. The decisional LWE problem is to distinguish the following two distributions with a non-negligible advantage.

- Distribution 0. Select several instances at random $(\boldsymbol{a}, b) \xleftarrow{\$} \mathbb{Z}_q^{n+1}$, where $\boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $b \xleftarrow{\$} \mathbb{Z}_q$;
- Distribution 1. Select several instances at random $\boldsymbol{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $e \leftarrow \chi$, output $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e) \in \mathbb{Z}_q^{n+1}$.

The difficulty of the LWE problem is demonstrated by the following theorem.

Theorem 1 [28]: Suppose that the integer $n \geq 1, q = q(n)$ is an integer that can be decomposed into the product of several prime numbers, and $\chi = \chi(n)$ is a sampleable B-bounded Gaussian distribution with $B \geq \omega(\log n)\cdot\sqrt{n}$. If there exists an efficient algorithm that can solve the average-case $\mathrm{LWE}_{n,q,\chi}$ problem, then

- For any n-dimensional lattice, there exists an efficient quantum algorithm that can solve the GapSVP problem with an approximate factor of $\tilde{O}(nq/B)$;
- For any n-dimensional lattice, if $q = q(n) \geq \tilde{O}(2^{n/2})$, there exists an efficient classical algorithm that can solve the GapSVP problem with an approximate factor of $\tilde{O}(nq/B)$.

Lemma 3 [28]: The decisional LWE problem assumption holds if

$$|Pr[\mathcal{A} \leftarrow Distr0]| - |Pr[\mathcal{A} \leftarrow Distr1]| = negl(n) \quad (12)$$

for any PPT adversary.

D. MULTI-IDENTITY BASED FULLY HOMOMORPHIC ENCRYPTION

We give the definition of MIBFHE, which is mainly adapted and summarized from the definition of IBFHE in [3], and the definition of MIBFHE in [6].

Definition 10 ([3], [6] MIBFHE): An MIBFHE scheme consists of six PPT algorithms (Setup, Extract, Enc, Extend, Eval and Dec) defined as follows.

- Setup$(1^\lambda, 1^K, 1^L)$: Input the security parameter λ, the maximum circuit depth L for homomorphic operations and the maximum number of different identities K. Output MPK and MSK.
- Extract(MPK, MSK, id) : Input MPK, MSK and the identity vector id, output the public key $\boldsymbol{A}_{id}$ and the corresponding secret key $\boldsymbol{s}_{id}$.
- Enc $(MPK, id, \mu \in \{0,1\})$: Input MPK, id and a bit message $\mu \in \{0,1\}$, output the fresh ciphertext $\boldsymbol{C}_{id}$ using the extracted public key.
- Extend $(MPK, (id_1, id_2, \cdots, id_k), \boldsymbol{C}_{id})$: Input MPK, identities involved in the computation $(id_1, id_2, \cdots, id_k)$ and the fresh ciphertext $\boldsymbol{C}_{id}$. Output the extended ciphertext $\hat{\boldsymbol{C}}_{id}$ under the concatenation of k identities $(id_1, id_2, \cdots, id_k)$.
- Eval $\left(MPK, (\hat{\boldsymbol{C}}_{id_1}, \hat{\boldsymbol{C}}_{id_2}, \cdots, \hat{\boldsymbol{C}}_{id_N}), f\right)$: Input MPK and a Boolean circuit f with N ciphertexts wires, and output the evaluated ciphertext $\hat{\boldsymbol{C}}_{eval}$.
- Dec $\left(MPK, (\boldsymbol{s}_{id_1}, \boldsymbol{s}_{id_2}, \cdots, \boldsymbol{s}_{id_k}), \boldsymbol{C}_{id}\right)$: Input MPK, the concatenation of k secret keys and $\boldsymbol{C}_{id}$ which may be the extended or evaluated ciphertext. Output a bit $\mu \in \{0,1\}$.

For the MIBFHE scheme in Definition 10, we define the security model according to IBE security model.

Definition 11 ([26] Indistinguishable from Random, Select-Identity, Chosen-Plaintext Attachment (IND-sID-CPA)): For an MIBFHE scheme consisting of six PPT algorithms (Setup, Extract, Enc, Extend, Eval, and Dec), the security mode between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$ can be defined as follows:

- Initial: The adversary is given the maximum depth of the computing circuits L and the maximum number of identities K involved and outputs a target identity id^*.
- Setup: For a security parameter λ, the challenger runs Setup$(1^\lambda, 1^K, 1^L)$ to generate (MPK, MSK), sends MPK to the adversary, and keeps MSK to itself.
- Phase 1: The adversary can issue polynomial time queries on identities $\{id_i\}$ where $id^* \notin \{id_i\}$ and $\{id_i\}$ are not a prefix of id^*. Then, the challenger runs Extract(MPK, MSK, id) to generate a secret key $\boldsymbol{s}_{id}$ corresponding to a public key $\boldsymbol{A}_{id}$ and sends $\boldsymbol{s}_{id}$ to the adversary.
- Challenge: Once the adversary determines the query in phase 1 is over, it selects two bits $(\mu_0, \mu_1) \in \{0,1\}$ and sends them to the challenger. The challenger chooses a random bit $\alpha \in \{0,1\}$ and runs Enc (MPK, id^*, μ_α) to obtain a ciphertext $\boldsymbol{C}_\alpha \leftarrow$ Enc (MPK, id^*, μ_α). Then the challenger sends $\boldsymbol{C}_\alpha$ as the challenge to the adversary.
- Phase 2: The adversary issues additional adaptive queries as in Phase 1.
- Guess: Finally, the adversary outputs a guess α' in polynomial time.

We define the advantage of adversary in breaking the above IND-sID-CPA security game is

$$Adv_{\mathcal{A},\varepsilon}^{\text{IND-sID-CPA}} = \left|Pr[\mathcal{A}(1^\lambda, 0) = 1] - Pr[\mathcal{A}(1^\lambda, 1) = 1]\right| \quad (13)$$

So, the above MIBFHE scheme is IND-sID-CPA secure if $Adv_{\mathcal{A},\varepsilon}^{\text{IND-sID-CPA}} \leq negl(\lambda)$ for any $\lambda \in \mathbb{N}$ and polynomial time adversary $\mathcal{A}$.

III. OUR MKFHE SCHEME

This section mainly introduces how we use "the decomposition method" to construct a concise and efficient MKFHE scheme from LWE.

A. CONSTRUCTION

Unlike previous MKFHE schemes, our MKFHE scheme consists of five PPT algorithms (Setup, KeyGen, Enc, Eval, Dec) which do not need the fresh ciphertext.

- MKFHE.Setup$(1^\lambda, 1^L, 1^K)$: Take the security parameter $\lambda \in \mathbb{N}$, the maximum depth L of the Boolean circuit and the upper bound K of the number of users as input. Choose $n = n(\lambda), m = O(n\ell) = 2n\ell$ where $\ell = \lceil \log q \rceil$, a large prime q and an error distribution $\chi = \chi(\lambda, L)$ with parameter $2\sqrt{n}$ for upper bound $B_\chi = \Theta(n)$ standard discrete Gaussian distribution appropriately for $\text{LWE}_{n,q,\chi}$ that achieves at least 2^λ security against known attacks. Choose a random matrix $\boldsymbol{B} \in \mathbb{Z}_q^{(n-1)\times m}$ and output public parameter $PP = (n, m, q, \chi, B_\chi, \boldsymbol{B})$.
- MKFHE.KeyGen(PP) : Sample $\bar{\boldsymbol{t}} \leftarrow \chi^{n-1}$ and $\boldsymbol{e} \leftarrow \chi^m$ from the standard discrete Gaussian distribution. Set $\boldsymbol{b} = \bar{\boldsymbol{t}}\boldsymbol{B} + \boldsymbol{e} \in \mathbb{Z}_q^m$, output public key $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{B} \end{pmatrix} \in \mathbb{Z}_q^{n\times m}$ and secret key $\boldsymbol{t} = (1, -\bar{\boldsymbol{t}}) \in \mathbb{Z}_q^n$. Clearly $\boldsymbol{t}\boldsymbol{A} = \boldsymbol{e} \in \mathbb{Z}_q^m$.
- MKFHE.Enc$(\mu, (pk_1, pk_2, \cdots, pk_K))$: Assume $\hat{\boldsymbol{t}} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_K) \in \mathbb{Z}_q^{Kn}$ is the concatenation of K secret keys. For the *i-th* user, choose a series of matrices $\left(\boldsymbol{M}_i^1, \boldsymbol{M}_i^2, \cdots, \boldsymbol{M}_i^K\right)$ where $\boldsymbol{M}_i^j \in \{0,1\}^{m\times n\ell}$, $i, j \in [K]$ as randomness and define the extended ciphertext

$$\hat{\boldsymbol{C}}_i = \boldsymbol{X}_i + \boldsymbol{Y}_i \in \mathbb{Z}_q^{Kn\times Kn\ell} \tag{14}$$

as the encryption of message $\mu \in \{0,1\}$.

(1). Constructing $\boldsymbol{X}$. As described in the technical overview above, the first part $\boldsymbol{X}$ consists of public keys and randomness matrices, which can be described as

$$\boldsymbol{X}_i = \begin{pmatrix} \boldsymbol{A}_i\boldsymbol{M}_i^1 & \cdots & \boldsymbol{0}^{n\times n\ell} \\ \vdots & \ddots & \\ \boldsymbol{A}_1\boldsymbol{M}_i^1 & \cdots \boldsymbol{A}_i\boldsymbol{M}_i^i & \cdots \boldsymbol{A}_K\boldsymbol{M}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0}^{n\times n\ell} & \boldsymbol{0}^{n\times n\ell} & \boldsymbol{A}_i\boldsymbol{M}_i^K \end{pmatrix} \in \mathbb{Z}_q^{Kn\times Kn\ell} \tag{15}$$

where the diagonal elements are $\left(\boldsymbol{A}_i\boldsymbol{M}_i^1, \boldsymbol{A}_i\boldsymbol{M}_i^2, \cdots, \boldsymbol{A}_i\boldsymbol{M}_i^K\right)$, the elements $\boldsymbol{X}[i,\cdot]$ in row i are $\left(\boldsymbol{A}_1\boldsymbol{M}_i^1, \boldsymbol{A}_2\boldsymbol{M}_i^2, \cdots, \boldsymbol{A}_K\boldsymbol{M}_i^K\right)$ and the rest elements are 0. We define and compute

$$\begin{aligned} \boldsymbol{p}_{i,j} &= \boldsymbol{t}_i\boldsymbol{A}_j + \boldsymbol{t}_j\boldsymbol{A}_i \\ &= (1, -\bar{\boldsymbol{t}}_i)\begin{pmatrix} \boldsymbol{b}_j \\ \boldsymbol{B} \end{pmatrix} + (1, -\bar{\boldsymbol{t}}_j)\begin{pmatrix} \boldsymbol{b}_i \\ \boldsymbol{B} \end{pmatrix} \\ &= \boldsymbol{b}_j - \bar{\boldsymbol{t}}_i\boldsymbol{B} + \boldsymbol{b}_i - \bar{\boldsymbol{t}}_j\boldsymbol{B} \\ &= \boldsymbol{e}_i + \boldsymbol{e}_j \end{aligned} \tag{16}$$

when $i \neq j \in [K]$. We also define $\boldsymbol{p}_i = \boldsymbol{t}_i\boldsymbol{A}_i = \boldsymbol{e}_i$ when $i = j$. Refer to (16), we can compute

$$\begin{aligned} \hat{\boldsymbol{t}}\boldsymbol{X}_i &= (\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_K)\begin{pmatrix} \boldsymbol{A}_i\boldsymbol{M}_i^1 & & \boldsymbol{0}^{n\times n\ell} \\ \vdots & \ddots & \\ \boldsymbol{A}_1\boldsymbol{M}_i^1 & \cdots \boldsymbol{A}_i\boldsymbol{M}_i^i & \cdots \boldsymbol{A}_K\boldsymbol{M}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0}^{n\times n\ell} & \boldsymbol{0}^{n\times n\ell} & \boldsymbol{A}_i\boldsymbol{M}_i^K \end{pmatrix} \\ &= \left(\boldsymbol{p}_{i,1}, \boldsymbol{p}_{i,2}, \cdots, \boldsymbol{p}_i, \cdots, \boldsymbol{p}_{i,K}\right) \\ &= \left((\boldsymbol{e}_i + \boldsymbol{e}_1)\boldsymbol{M}_i^1, \cdots, \boldsymbol{e}_i\boldsymbol{M}_i^i, \cdots, (\boldsymbol{e}_i + \boldsymbol{e}_K)\boldsymbol{M}_i^K\right) \\ &= \hat{\boldsymbol{e}}_i \in \mathbb{Z}_q^{Kn\ell} \end{aligned} \tag{17}$$

(2). Constructing $\boldsymbol{Y}$. As described in the technical overview above, the second part $\boldsymbol{Y}$ consists of the plain message and the garget matrix, which can be described as

$$\begin{aligned} \boldsymbol{Y}_i &= \mu_i\boldsymbol{G}_{Kn} \\ &= \mu_i(\boldsymbol{I}_{Kn} \otimes \boldsymbol{g}) \\ &= \mu_i \begin{pmatrix} \boldsymbol{G}_n & \cdots \boldsymbol{0}^{n\times n\ell} \\ \vdots & \ddots \quad \vdots \\ \boldsymbol{0}^{n\times n\ell} & \cdots \quad \boldsymbol{G}_n \end{pmatrix}_{K\times K} \in \mathbb{Z}_q^{Kn\times Kn\ell} \end{aligned} \tag{18}$$

We can also compute

$$\hat{\boldsymbol{t}}\boldsymbol{Y}_i = \mu_i\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn} \tag{19}$$

Now, we have completed the ciphertext generation and extension. So, we can describe $\hat{\boldsymbol{C}}_i$ as

$$\begin{aligned} \hat{\boldsymbol{C}}_i &= \boldsymbol{X}_i + \boldsymbol{Y}_i \\ &= \begin{pmatrix} \boldsymbol{A}_i\boldsymbol{M}_i^1 & \cdots & \boldsymbol{0}^{n\times n\ell} \\ \vdots & \ddots & \\ \boldsymbol{A}_1\boldsymbol{M}_i^1 & \cdots \boldsymbol{A}_i\boldsymbol{M}_i^i & \cdots \boldsymbol{A}_K\boldsymbol{M}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0}^{n\times n\ell} & \boldsymbol{0}^{n\times n\ell} & \boldsymbol{A}_i\boldsymbol{M}_i^K \end{pmatrix} + \mu_i\boldsymbol{G}_{Kn} \end{aligned} \tag{20}$$

Referring to (17) and (19), the decryption process is

$$\begin{aligned} \hat{\boldsymbol{t}}\hat{\boldsymbol{C}}_i &= \hat{\boldsymbol{t}}\boldsymbol{X}_i + \hat{\boldsymbol{t}}\boldsymbol{Y}_i \\ &= \hat{\boldsymbol{e}} + \mu_i\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn} \end{aligned} \tag{21}$$

- MKFHE.Eval$((\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2, \cdots, \hat{\boldsymbol{C}}_N), f)$: Input ciphertext set $(\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2, \cdots, \hat{\boldsymbol{C}}_N)$ and effective Boolean circuit f, output the evaluated ciphertext. The definitions of homomorphic addition, multiplication and NAND operations are the same as [3] and [7]:

$$\begin{aligned} \text{GSW.Add}(\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2) &= \hat{\boldsymbol{C}}^{(+)} \\ &= \hat{\boldsymbol{C}}_1 + \hat{\boldsymbol{C}}_2 \\ &= (\boldsymbol{X}_1 + \boldsymbol{X}_2) + (\boldsymbol{Y}_1 + \boldsymbol{Y}_2) \\ &= (\boldsymbol{X}_1 + \boldsymbol{X}_2) + (\mu_1 + \mu_2) \\ &\quad \boldsymbol{G}_{Kn} \in \mathbb{Z}_q^{Kn\times Kn\ell} \end{aligned} \tag{22}$$

$$\begin{aligned}\text{GSW.Multi}(\hat{\boldsymbol{C}}_1,\hat{\boldsymbol{C}}_2) &= \hat{\boldsymbol{C}}^{(\times)}\\ &= \hat{\boldsymbol{C}}_1\cdot\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)\\ &= (\boldsymbol{X}_1+\mu_1\boldsymbol{G}_{Kn})\cdot\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)\\ &= \boldsymbol{X}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\left(\boldsymbol{X}_2+\mu_2\boldsymbol{G}_{Kn}\right)\\ &= \left(\boldsymbol{X}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\boldsymbol{X}_2\right)+\mu_1\mu_2\boldsymbol{G}_{Kn}\end{aligned} \tag{23}$$

$$\begin{aligned}\text{GSW.NAND}(\hat{\boldsymbol{C}}_1,\hat{\boldsymbol{C}}_2) &= \hat{\boldsymbol{C}}^{(\text{NAND})}\\ &= \boldsymbol{G}_{Kn}-\hat{\boldsymbol{C}}_1\cdot\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)\\ &= (1-\mu_1\mu_2)\,\boldsymbol{G}_{Kn}\\ &\quad-\left(\boldsymbol{X}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\boldsymbol{X}_2\right)\in\mathbb{Z}_q^{Kn\times Kn\ell}\end{aligned} \tag{24}$$

which satisfies

$$\begin{aligned}\hat{\boldsymbol{t}}\hat{\boldsymbol{C}}^{(+)} &= \hat{\boldsymbol{t}}\left(\hat{\boldsymbol{C}}_1+\hat{\boldsymbol{C}}_2\right)\\ &= \hat{\boldsymbol{t}}\,(\boldsymbol{X}_1+\boldsymbol{X}_2)+(\mu_1+\mu_2)\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}\\ &= (\mu_1+\mu_2)\,\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}+\left(\hat{\boldsymbol{e}}_1+\hat{\boldsymbol{e}}_2\right)\end{aligned} \tag{25}$$

$$\begin{aligned}\hat{\boldsymbol{t}}\hat{\boldsymbol{C}}^{(\times)} &= \hat{\boldsymbol{t}}\hat{\boldsymbol{C}}_1\cdot\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)\\ &= \hat{\boldsymbol{t}}\left(\boldsymbol{X}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\boldsymbol{X}_2\right)+\mu_1\mu_2\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}\\ &= \mu_1\mu_2\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}+\left(\hat{\boldsymbol{e}}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\hat{\boldsymbol{e}}_2\right)\end{aligned} \tag{26}$$

$$\begin{aligned}\hat{\boldsymbol{t}}\hat{\boldsymbol{C}}^{(NAND)} &= (1-\mu_1\mu_2)\,\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}-\hat{\boldsymbol{t}}\left(\boldsymbol{X}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\boldsymbol{X}_2\right)\\ &= (1-\mu_1\mu_2)\,\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}\\ &\quad-\left(\hat{\boldsymbol{e}}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\hat{\boldsymbol{e}}_2\right)\in\mathbb{Z}_q^{Kn\times Kn\ell}\end{aligned} \tag{27}$$

- MKFHE.Dec$(\hat{\boldsymbol{C}},\hat{\boldsymbol{t}})$: We only analyze binary cases where $\mu\in\{0,1\}$. Refer to (21), we define a vector $\boldsymbol{\eta}=(\lceil q/2\rceil,0,\cdots,0)^{\mathrm{T}}\in\mathbb{Z}_q^{Kn}$ and compute

$$\begin{aligned}\mu' &= \hat{\boldsymbol{t}}\hat{\boldsymbol{C}}\cdot\boldsymbol{G}_{Kn}^{-1}(\boldsymbol{\eta})\\ &= \left(\hat{\boldsymbol{e}}+\mu\hat{\boldsymbol{t}}\boldsymbol{G}_{Kn}\right)\boldsymbol{G}_{Kn}^{-1}(\boldsymbol{\eta})\\ &=<\hat{\boldsymbol{e}},\boldsymbol{G}_{Kn}^{-1}(\boldsymbol{\eta})>+\mu<\hat{\boldsymbol{t}},\boldsymbol{\eta}>\\ &=<\hat{\boldsymbol{e}},\boldsymbol{G}_{Kn}^{-1}(\boldsymbol{\eta})>+\mu\cdot\lceil q/2\rceil\end{aligned} \tag{28}$$

According to the analysis of parameters in [3] and [7], we know that $\left\langle\hat{\boldsymbol{e}},\boldsymbol{G}_{Kn}^{-1}(\boldsymbol{\eta})\right\rangle\le\frac{q}{8}<\frac{q}{4}$ should be satisfied to ensure the correctness of decryption. Finally, we output $u'=1$ when $\left|u'-\lceil q/2\rceil\right|<\frac{q}{4}$ and output $u'=0$ when $\left|u'\right|<\frac{q}{4}$.

The working model of our MKFHE scheme is shown in Fig. 1.

B. PARAMETERS SETTING

For the above algorithms (Setup, KeyGen, Enc, Eval, Dec), and selected parameters $(n,m,q,\chi,B_\chi,\boldsymbol{B})$, we analyse the worst-case noise growth using Definition 3.

In the Enc(•) phase, $\hat{\boldsymbol{C}}_i$ is the extended ciphertext of the *i-th* user under secret key $\hat{\boldsymbol{t}}$. Refer to (17), (19), (21), we know

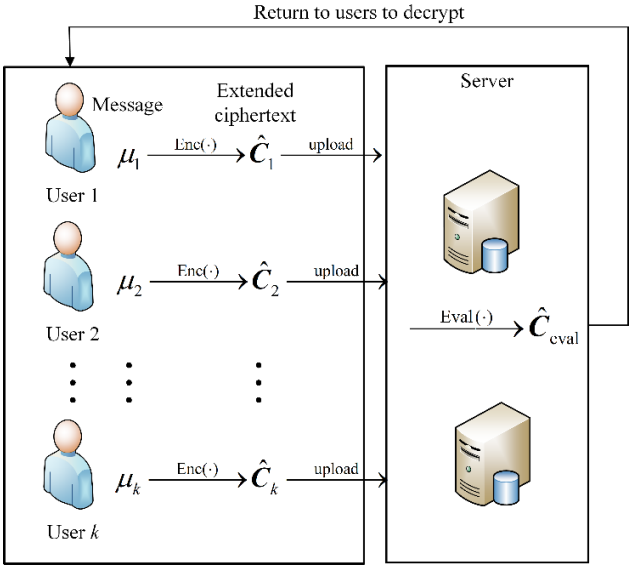

FIGURE 1. The working mode of our MKFHE scheme. Users directly encrypt private data into the extended ciphertexts that can perform homomorphic operations and upload them to the server. The cloud server performs homomorphic operations on the extended ciphertexts and returns the results to users for joint decryption.

that the noise is

$$\begin{aligned}\beta_{enc} &= \hat{\boldsymbol{e}}\\ &= \left((\boldsymbol{e}_i+\boldsymbol{e}_1)\,\boldsymbol{M}_i^1,\cdots,\boldsymbol{e}_i\boldsymbol{M}_i^i,\cdots,(\boldsymbol{e}_i+\boldsymbol{e}_K)\,\boldsymbol{M}_i^K\right)\end{aligned} \tag{29}$$

for $\|\boldsymbol{e}\|_\infty\le B_\chi$ and $\|\boldsymbol{M}\|_\infty\le 1$. Compute

$$\begin{aligned}\|\beta_{enc}\|_\infty &= \left\|\hat{\boldsymbol{e}}\right\|_\infty\\ &= \left\|\left((\boldsymbol{e}_i+\boldsymbol{e}_1)\,\boldsymbol{M}_i^1,\cdots,\boldsymbol{e}_i\boldsymbol{M}_i^i,\cdots,(\boldsymbol{e}_i+\boldsymbol{e}_K)\,\boldsymbol{M}_i^K\right)\right\|_\infty\\ &\le 2mB_\chi\end{aligned} \tag{30}$$

Hence, $\hat{\boldsymbol{C}}_i$ is the $2mB_\chi$-noisy ciphertext.

In the Eval(•) phase, we only need to consider the homomorphic multiplication operation, which has a larger noise increase. Referring to (26), we know that

$$\beta_{eval}=\left(\hat{\boldsymbol{e}}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\hat{\boldsymbol{e}}_2\right) \tag{31}$$

for one homomorphic operation. As known $\left\|\hat{\boldsymbol{e}}\right\|_\infty\le 2mB_\chi$, we can compute

$$\begin{aligned}\|\beta_{eval}\|_\infty &= \left\|\hat{\boldsymbol{e}}_1\boldsymbol{G}_{Kn}^{-1}(\hat{\boldsymbol{C}}_2)+\mu_1\hat{\boldsymbol{e}}_2\right\|_\infty\\ &\le 2mB_\chi(1+Kn\ell)\end{aligned} \tag{32}$$

According to the analysis of noise growth in computing circuits in [3] and [7], we could evaluate a circuit of depth L and a designed upper bound of users K while keeping the error magnitude at most

$$\left\|\beta_{final}\right\|_\infty\le 2mB_\chi(1+Kn\ell)^L \tag{33}$$

TABLE 2. Comparison of MKFHE schemes.

Scheme	Must fresh ciphertext	Auxiliary ciphertext [a]	Noise expansion rate [b]	q
[6]	YES	$\tilde{O}\left(n^4L^4\right)$	$1+n\ell$	$8\left(1+n\ell\right)\left(1+K\ell(1+n\ell)\right)^L B_\chi$
[7]	YES	$\tilde{O}\left(n^4L^4\right)$	$1+n^3\ell^3$	$4Kn^2\ell^2(1+n\ell)(1+Kn\ell)^L B_\chi$
Ours	NO	$\tilde{O}\left(n^2L^2\right)$	2	$16n\ell(1+Kn\ell)^L B_\chi$

We define that all sizes are in bits. Here K denotes the upper bound number of users; L represents the maximum depth of the Boolean circuits homomorphically evaluated (without bootstrapping); n is the dimension of the underlying LWE problem used for security; $\ell = \lceil \log q \rceil$. The $\tilde{O}(\cdot)$ notation hides the factors of the form $\log poly(n,k,\ell)$ for some polynomial functions.

[a]Auxiliary ciphertext indicates the ciphertext scale to be introduced when constructing the extended ciphertext.

[b]Noise expansion rate denotes the ratio between the noise of the extended ciphertext and the noise of the fresh ciphertext.

Our decryption algorithm works correctly as long as its error is smaller than $q/4$, hence it suffices to choose the modulus

$$q \geq 8mB_\chi(1+Kn\ell)^L \tag{34}$$

C. SECURITY

Theorem 2: Our MKFHE scheme described in this section is indistinguishable under chosen-plaintext attachment (IND-CPA) secure under the $LWE_{n,q,\chi}$ problem.

Proof of Theorem 2. The security can be proved by the training-challenge game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

Initialize. The challenger $\mathcal{C}$ runs MKFHE.Setup(1^λ, 1^L, 1^K) to generate public parameter $PP = (n, m, q, \chi, B_\chi, \boldsymbol{B})$ and then runs MKFHE.KeyGen(PP) to generate a pair of keys $pk_i = \boldsymbol{A}_i = \begin{pmatrix} \boldsymbol{b}_i \\ \boldsymbol{B} \end{pmatrix}$, $sk_i = \boldsymbol{t}_i$. Then it sends pk_i to adversary $\mathcal{A}$.

Training 1. This is the real IND-CPA game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The challenger $\mathcal{C}$ receives a pair of messages (μ_0, μ_1) from adversary $\mathcal{A}$, then chooses a random bit $\alpha \in \{0,1\}$ and a series of matrices $\left(\boldsymbol{M}_i^1, \boldsymbol{M}_i^2, \cdots, \boldsymbol{M}_i^K\right)$ where $\boldsymbol{M}_i^j \in \{0,1\}^{m\times n\ell}$, and runs MKFHE.Enc($\mu$, $(pk_1, pk_2, \cdots, pk_K)$) to generate a ciphertext $\hat{\boldsymbol{C}}_\alpha = \boldsymbol{X}_i + \mu_\alpha \boldsymbol{G}_{Kn} \in \mathbb{Z}_q^{Kn\times Kn\ell}$ of the message μ_α. The challenger $\mathcal{C}$ sends $\hat{\boldsymbol{C}}_\alpha$ to adversary $\mathcal{A}$.

Training 2. This is a hybrid experiment in the ideal world. The adversary $\mathcal{A}$ chooses a pair of messages (μ_0, μ_1) for challenger $\mathcal{C}$ again. The challenger $\mathcal{C}$ chooses a random bit $\alpha \in \{0,1\}$ and a uniformly random matrix $\boldsymbol{P} \in \mathbb{Z}_q^{Kn\times Kn\ell}$ to generate a ciphertext $\hat{\boldsymbol{C}}'_\alpha = \boldsymbol{P} + \mu_\alpha \boldsymbol{G}_n \in \mathbb{Z}_q^{Kn\times Kn\ell}$ of the message μ_α. The challenger $\mathcal{C}$ sends $\boldsymbol{C}'_\alpha$ to adversary $\mathcal{A}$.

Challenge. Finally, adversary $\mathcal{A}$ guesses the bit for $\alpha \in \{0,1\}$ as α'. The game outputs 1 if $\alpha' = \alpha$ and 0 otherwise.

Corollary 1: We define the probability that the adversary guesses bit $\alpha \in \{0,1\}$ in training 1 and training 2 as $\left|Pr\left[\alpha' = \alpha \middle| \mathrm{T}_1\right]\right|$ and $\left|Pr\left[\alpha' = \alpha \middle| \mathrm{T}_2\right]\right|$, respectively. We know that $\left|Pr\left[\alpha' = \alpha \middle| \mathrm{T}_2\right]\right| = \frac{1}{2}$ as the random matrix $\boldsymbol{P}$ is computationally independent of pk and μ. So, the adversary's advantage is

$$Adv[\mathcal{A}] = \left|Pr\left[\alpha' = \alpha \,|\mathrm{T}_1\right] - \frac{1}{2}\right| \tag{35}$$

Comparing the structure of $\hat{\boldsymbol{C}}_\alpha$ and $\hat{\boldsymbol{C}}'_\alpha$, we can translate the analysis of $\hat{\boldsymbol{C}}_\alpha$ and $\hat{\boldsymbol{C}}'_\alpha$ into the analysis of $\boldsymbol{X}_i$ and $\boldsymbol{P}$, as

$$\begin{cases} \hat{\boldsymbol{C}}_\alpha = \boldsymbol{X}_i + \mu_\alpha \boldsymbol{G}_{Kn} \in \mathbb{Z}_q^{Kn\times Kn\ell} \\ \text{comparison} \Updownarrow \\ \hat{\boldsymbol{C}}'_\alpha = \boldsymbol{P} + \mu_\alpha \boldsymbol{G}_n \in \mathbb{Z}_q^{Kn\times Kn\ell} \end{cases} \rightarrow \begin{cases} \boldsymbol{X}_i \\ \Updownarrow \\ \boldsymbol{P} \end{cases} \tag{36}$$

Lemma 4: Let $pp = (n, m, q, \chi, B_\chi, \boldsymbol{B})$ be such that the $LWE_{n,q,\chi}$ assumption holds. For $m = O(n\ell)$ and $\boldsymbol{A}$, $\boldsymbol{M}$ as generated above, the joint distribution $(\boldsymbol{A}, \boldsymbol{AM})$ is computationally indistinguishable from uniform over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times n\ell}$.

Analysing the structure of $\boldsymbol{X}_i$, we know that both the diagonal elements $\left(\boldsymbol{A}_i\boldsymbol{M}_i^1, \boldsymbol{A}_i\boldsymbol{M}_i^2, \cdots, \boldsymbol{A}_i\boldsymbol{M}_i^K\right)$ and the elements $\left(\boldsymbol{A}_1\boldsymbol{M}_i^1, \boldsymbol{A}_2\boldsymbol{M}_i^2, \cdots, \boldsymbol{A}_K\boldsymbol{M}_i^K\right)$ in row i apply to Lemma 4. Given that the rest elements of $\boldsymbol{X}_i$ is 0, we can conclude that $\boldsymbol{X}_i$ is computationally indistinguishable from $\boldsymbol{P}$, i.e.,

$$Adv[\mathcal{A}] = \left|Pr\left[\alpha' = \alpha \,|\mathrm{T}_1\right] - \frac{1}{2}\right| = negl\,(n) \tag{37}$$

Hence, any adversary cannot obtain useful information about the message μ in PPT under the $LWE_{n,q,\chi}$ problem. Therefore, Theorem 2 holds and our scheme is IND-CPA secure.

D. EFFICIENCY ANALYSIS OF MKFHE SCHEME

We compare our MKFHE scheme with MKFHE schemes in [6] and [7], which are commonly used to construct MIBFHE schemes. See Table 2 for comparison results. Through the analysis in Table 2, the following conclusions can be drawn.

- In our scheme, users do not need to generate fresh ciphertext in advance, but can directly generate the extended ciphertext that can perform homomorphic operations through the encryption algorithm, and the scale of auxiliary ciphertext introduced is smaller as the scale of the extended ciphertext is the same. This makes our scheme more concise.

- When performing homomorphic operations, our scheme significantly reduces the noise expansion rate and has a smaller modulus q, which makes our scheme more efficient.

IV. OUR MODIFIED IBE SCHEME

The IBE scheme in [22] has better efficiency parameters, but it cannot be directly combined with our MKFHE scheme. To this end, to ensure correctness and safety, we made an appropriate deformation for it.

A. CONSTRUCTION

- IBE.Setup$(1^\lambda, 1^L, 1^K)$: Choose parameters λ, K, L, q, B and error distribution χ as stated in the MKFHE.Setup$(1^\lambda, 1^L, 1^K)$. Let $m = 2n\ell$, $\ell = \lceil \log q \rceil$. Choose a uniformly random matrix $\bar{\boldsymbol{A}} \in \mathbb{Z}_q^{n\times n\ell}$, an invertible matrix $\boldsymbol{H} \in \mathbb{Z}_q^{n\times n}$ and a collision-resistant hash function $H : \mathbb{Z}_q^* \to \mathbb{Z}_q^n$. Use the trapdoor generation algorithm $TrapGen(1^n, 1^m, q)$ described in Lemma 1 to generate a matrix $\boldsymbol{A} = \left[\bar{\boldsymbol{A}} \,\middle\|\, \left(\boldsymbol{HG} - \bar{\boldsymbol{A}}\bar{\boldsymbol{R}}\right)\right] \in \mathbb{Z}_q^{n\times m}$ with its trapdoor matrix $\boldsymbol{R} \in \mathbb{Z}^{n\ell\times n\ell}$. Output $MPK = (\boldsymbol{A}, H)$ and $MSK = \boldsymbol{R}$. For different identities, the matrix $\boldsymbol{A}$ stays the same.
- IBE.Extract (MPK, MSK, id) : Input MPK, MSK and user's identity $id \in \mathbb{Z}_q^*$. Instead of using the full-rank differences (FRD) encoding function in [22], here we use the hash function to get an identity vector $\boldsymbol{u}_{id} \in \mathbb{Z}_q^n \leftarrow H(id)$ for each different id. Then we use the sampling algorithm SampleD$(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{u}_{id}, \sigma)$ described in Lemma 2 to generate the secret key $\boldsymbol{t}_{id} \in \mathbb{Z}_q^m$ for each different id, satisfying $\boldsymbol{A}\cdot\boldsymbol{t}_{id} = \boldsymbol{u}_{id} \bmod q$. Finally, we let $\boldsymbol{A}_{id} = [\boldsymbol{u}_{id} \| \boldsymbol{A}]^T \in \mathbb{Z}_q^{(m+1)\times n}$ and $\boldsymbol{s}_{id} = \left(1, -\boldsymbol{t}_{id}^T\right) \in \mathbb{Z}_q^{m+1}$. So, we can compute $\boldsymbol{s}_{id} \cdot \boldsymbol{A}_{id} = \boldsymbol{0} \bmod q$. Output the secret key $\boldsymbol{s}_{id}$.
- IBE.Enc (MPK, id, μ) : Input MPK, id and the plaintext message $\mu \in \{0, 1\}$. Choose a uniformly random vector $\boldsymbol{m} \in \{0, 1\}^n$ and the error vector $\boldsymbol{e} \in \chi^{m+1}$. Define $\boldsymbol{\mu} = \left(\mu \cdot \lceil q/2 \rceil, 0, \cdots, 0\right) \in \mathbb{Z}_q^{m+1}$, and output the fresh ciphertext matrix

$$\boldsymbol{c}_{id} = \boldsymbol{A}_{id}\boldsymbol{m} + \boldsymbol{\mu} + \boldsymbol{e} \in \mathbb{Z}_q^{m+1} \tag{38}$$

  encrypting μ under the secret key $\boldsymbol{s}_{id}$.
- IBE.Dec $(MPK, \boldsymbol{c}_{id}, \boldsymbol{s}_{id})$: Input MPK, $\boldsymbol{c}_{id}$ and the secret key $\boldsymbol{s}_{id}$, compute

$$\begin{aligned}\mu' &= <\boldsymbol{s}_{id}, \boldsymbol{c}_{id}> \\ &= \mu \cdot \lceil q/2 \rceil + <\boldsymbol{s}_{id}, \boldsymbol{e}> \in \mathbb{Z}_q\end{aligned} \tag{39}$$

  According to the analysis of decryption correctness in [14], we know that $<\boldsymbol{s}_{id}, \boldsymbol{e}> \le \frac{q}{5} < \frac{q}{4}$ with a great probability. So, we can conclude that $\left\| <\boldsymbol{s}_{id}, \boldsymbol{c}_{id}> - \lceil q/2 \rceil \right\| = \| <\boldsymbol{s}_{id}, \boldsymbol{e}> \| < \frac{q}{4}$ when $b = 1$, and $\| <\boldsymbol{s}_{id}, \boldsymbol{c}_{id}> \| = \| <\boldsymbol{s}_{id}, \boldsymbol{e}> \| < \frac{q}{4}$ when $b = 0$.

B. PARAMETERS, SECURITY, AND EFFICIENCY

We mainly used the IBE scheme proposed in [22]. The difference is that we added a collision-resistant hash function $H : \mathbb{Z}_q^* \to \mathbb{Z}_q^n$ and adjusted the construction of public key $\boldsymbol{A}_{id}$ to change the relationship between the identity vector and the public key. However, these do not change the choice of parameters and the dimensions of the matrix and vector. Compared with [22], the parameters, security, and efficiency of our modified IBE scheme have not changed.

V. OUR MIBFHE SCHEME

As described in Section IV, the fresh ciphertext encrypted under a certain identity can be decrypted correctly. However, homomorphic operations cannot be performed among fresh ciphertexts encrypted under different identities. It naturally occurs to us that we can use the extension algorithm of MKFHE in Section III to process fresh ciphertexts under different identities so that they meet the homomorphic properties. In this section, we describe how to combine the MKFHE scheme in Section III with the IBE scheme in Section IV to construct our MIBFHE scheme.

A. CONSTRUCTION

- MIBFHE.Setup$(1^\lambda, 1^L, 1^K)$: Choose parameters λ, K, L, q, B and error distribution χ as stated in the MKFHE.Setup$(1^\lambda, 1^L, 1^K)$. Let $m = 2n\ell$, $\ell = \lceil \log q \rceil$. Run IBE.Setup$(1^\lambda, 1^L, 1^K)$ and output $MPK = (\boldsymbol{A}, H)$ and $MSK = \boldsymbol{R}$.
- MIBFHE.Extract $\left(MPK, MSK, \{id_i\}_{i\in[K]}\right)$: Input MPK, MSK and users' identity vector set $\{id_i\}_{i\in[K]} \in \mathbb{Z}_q^n$. Run IBE.Extract (MPK, MSK, id) and output secret keys set $\{\boldsymbol{s}_{id_i}\}_{i\in[K]}$.
- MIBFHE.Enc $\left(MPK, \{id_i\}_{i\in[K]}, \mu\right)$: Input MPK, the identities set $\{id_i\}_{i\in[K]}$ and $\mu \in \{0, 1\}$. Assume $\hat{\boldsymbol{s}}_{id} = \left(\boldsymbol{s}_{id_1}, \boldsymbol{s}_{id_2}, \cdots, \boldsymbol{s}_{id_K}\right) \in \mathbb{Z}_q^{K(m+1)}$ is the concatenation of K secret keys corresponding to K identities. Run MKFHE.Enc$(\mu, (pk_1, pk_2, \cdots, pk_K))$ to generate the extended ciphertext $\hat{\boldsymbol{C}}_{id_i} = \boldsymbol{X}_{id_i} + \boldsymbol{Y}_{id_i} \in \mathbb{Z}_q^{K(m+1)\times K(m+1)\ell}$ as follows:
  (1). Constructing $\boldsymbol{X}_{id_i}$. Similar to (15), we can define

$$\boldsymbol{X}_{id_i} = \begin{pmatrix} \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \\ \boldsymbol{A}_{id_1}\tilde{\boldsymbol{M}}_i^1 & \cdots \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^i & \cdots \boldsymbol{A}_{id_K}\tilde{\boldsymbol{M}}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0} & \cdots \quad \boldsymbol{0} & \cdots \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^K \end{pmatrix} \tag{40}$$

  where the randomness matrices are $\left(\tilde{\boldsymbol{M}}_i^1, \tilde{\boldsymbol{M}}_i^2, \cdots, \tilde{\boldsymbol{M}}_i^K\right) \in \{0, 1\}^{n\times(m+1)\ell}$. Similar to (16), we can compute

$$\begin{aligned}&\boldsymbol{s}_{id_i}\boldsymbol{A}_{id_j} + \boldsymbol{s}_{id_j}\boldsymbol{A}_{id_i} \\ &= \left(1, -\boldsymbol{t}_{id_i}^T\right)\begin{pmatrix}\boldsymbol{u}_{id_j}^T \\ \boldsymbol{A}^T\end{pmatrix} + \left(1, -\boldsymbol{t}_{id_j}^T\right)\begin{pmatrix}\boldsymbol{u}_{id_i}^T \\ \boldsymbol{A}^T\end{pmatrix}\end{aligned}$$

$$= \boldsymbol{u}_{id_j}^T - \boldsymbol{t}_{id_i}^T \boldsymbol{A}^T + \boldsymbol{u}_{id_i}^T - \boldsymbol{t}_{id_j}^T \boldsymbol{A}^T$$
$$= \boldsymbol{0}^{(m+1)\ell} \tag{41}$$

Referring to (40) and (41), we know that

$$\hat{\boldsymbol{s}}_{id}\boldsymbol{X}_{id_i} = \left(\boldsymbol{s}_{id_1}, \boldsymbol{s}_{id_2}, \cdots, \boldsymbol{s}_{id_K}\right) \begin{pmatrix} \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \\ \boldsymbol{A}_{id_1}\tilde{\boldsymbol{M}}_i^1 & \cdots \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^i & \cdots \boldsymbol{A}_{id_K}\tilde{\boldsymbol{M}}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^K \end{pmatrix} = \boldsymbol{0}^{K(m+1)\ell} \tag{42}$$

(2). Constructing $\boldsymbol{Y}_{id_i}$. Similar to (18), we define

$$\begin{aligned} \boldsymbol{Y}_{id_i} &= \mu_i \boldsymbol{G}_{K(m+1)} + \boldsymbol{E}_{id_i} \\ &= \mu_i (\boldsymbol{I}_{K(m+1)} \otimes \boldsymbol{g}) + \boldsymbol{E}_{id_i} \\ &= \mu_i \begin{pmatrix} \boldsymbol{G}_{m+1} & \cdots \boldsymbol{0}^{(m+1)\times(m+1)\ell} \\ \vdots & \ddots \quad \vdots \\ \boldsymbol{0}^{(m+1)\times(m+1)\ell} & \cdots \quad \boldsymbol{G}_{m+1} \end{pmatrix}_{K\times K} + \boldsymbol{E}_{id_i} \end{aligned} \tag{43}$$

where the error matrix is $\boldsymbol{E}_{id_i} \leftarrow \chi^{K(m+1)\times K(m+1)\ell}$. The decryption process can be described as

$$\hat{\boldsymbol{s}}_{id}\boldsymbol{Y}_{id_i} = \mu_i \hat{\boldsymbol{s}}_{id}\boldsymbol{G}_{K(m+1)} + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i} \tag{44}$$

Now, we have completed the ciphertext generation and extension, and we can describe $\hat{\boldsymbol{C}}_{id_i}$ as

$$\begin{aligned} \hat{\boldsymbol{C}}_{id_i} &= \boldsymbol{X}_{id_i} + \boldsymbol{Y}_{id_i} \\ &= \begin{pmatrix} \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \\ \boldsymbol{A}_{id_1}\tilde{\boldsymbol{M}}_i^1 & \cdots \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^i & \cdots \boldsymbol{A}_{id_K}\tilde{\boldsymbol{M}}_i^K \\ \vdots & & \ddots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{id_i}\tilde{\boldsymbol{M}}_i^K \end{pmatrix} \\ &\quad + \mu_i \boldsymbol{G}_{K(m+1)} + \boldsymbol{E}_{id_i} \end{aligned} \tag{45}$$

Referring to (42) and (44), the decryption process is

$$\begin{aligned} \hat{\boldsymbol{s}}_{id}\hat{\boldsymbol{C}}_{id_i} &= \hat{\boldsymbol{s}}_{id}\boldsymbol{X}_{id_i} + \hat{\boldsymbol{s}}_{id}\boldsymbol{Y}_{id_i} \\ &= \mu_i \hat{\boldsymbol{s}}_{id}\boldsymbol{G}_{K(m+1)} + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i} \end{aligned} \tag{46}$$

- MIBFHE.Eval $\left(MPK, (\hat{\boldsymbol{C}}_{id_1}, \hat{\boldsymbol{C}}_{id_2}, \cdots, \hat{\boldsymbol{C}}_{id_N}), f\right)$: Input MPK and a Boolean circuit f with N ciphertexts wires. Run MKFHE.Eval$(\cdot)$ to generate the evaluated ciphertext $\hat{\boldsymbol{C}}_{eval}$.
- MIBFHE.Dec $\left(MPK, \hat{\boldsymbol{s}}_{id}, \hat{\boldsymbol{C}}_{id_i}\right)$: Input MPK, the concatenation of K secret keys $\hat{\boldsymbol{s}}_{id}$, and the extended ciphertext $\hat{\boldsymbol{C}}_{id_i}$. Referring to (28), we also set a vector $\tilde{\boldsymbol{\eta}} = \left(\lceil q/2 \rceil, 0, \cdots, 0\right)^T \in \mathbb{Z}_q^{K(m+1)}$ and compute

$$\begin{aligned} \mu' &= \hat{\boldsymbol{s}}_{id}\hat{\boldsymbol{C}}_{id_i} \cdot \boldsymbol{G}_{K(m+1)}^{-1}(\tilde{\boldsymbol{\eta}}) \\ &= \left(\mu \hat{\boldsymbol{s}}_{id}\boldsymbol{G}_{K(m+1)} + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i}\right) \cdot \boldsymbol{G}_{K(m+1)}^{-1}(\tilde{\boldsymbol{\eta}}) \\ &= \mu \left\langle \hat{\boldsymbol{s}}_{id}, \tilde{\boldsymbol{\eta}} \right\rangle + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i}\boldsymbol{G}_{K(m+1)}^{-1}(\tilde{\boldsymbol{\eta}}) \\ &= \mu \cdot \lceil q/2 \rceil + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i}\boldsymbol{G}_{K(m+1)}^{-1}(\tilde{\boldsymbol{\eta}}) \end{aligned} \tag{47}$$

Output $\mu' = 1$ when $\left\|\mu' - \lceil q/2 \rceil\right\| \leq \frac{q}{4}$, and output $\mu' = 0$ when $\left\|\mu'\right\| \leq \frac{q}{4}$.

B. PARAMETERS SETTING

Using the same method stated in Section III, we analyse the worst-case noise growth of our MIBFHE scheme.

In the Enc$(\bullet)$ phase, $\hat{\boldsymbol{C}}_{id_i}$ is the extended ciphertext. Referring to (42), (44), (46), we know that the noise is $\tilde{\beta}_{enc} = \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i}$ for $\left\|\boldsymbol{E}_{id_i}\right\|_\infty \leq B_\chi$. Compute

$$\left\|\tilde{\beta}_{enc}\right\|_\infty = \left\|\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_i}\right\|_\infty \leq KB_\chi(m+1) \tag{48}$$

So, $\hat{\boldsymbol{C}}_{id_i}$ is the $KB_\chi(m+1)$-noisy ciphertext.

In the Eval$(\bullet)$ phase, the homomorphic multiplication operation can be described as

$$\begin{aligned} \hat{\boldsymbol{C}}_{id}^{(\times)} &= \hat{\boldsymbol{C}}_{id_1} \cdot \boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &= \left(\boldsymbol{X}_{id_1} + \mu_1 \boldsymbol{G}_{K(m+1)} + \boldsymbol{E}_{id_1}\right) \cdot \boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &= (\boldsymbol{X}_{id_1} + \boldsymbol{E}_{id_1})\boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &\quad + \mu_1(\boldsymbol{X}_{id_2} + \mu_2 \boldsymbol{G}_{K(m+1)} + \boldsymbol{E}_{id_2}) \\ &= \mu_1\mu_2\boldsymbol{G}_{K(m+1)} + (\boldsymbol{X}_{id_1} + \boldsymbol{E}_{id_1})\boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &\quad + \mu_1(\boldsymbol{X}_{id_2} + \boldsymbol{E}_{id_2}) \end{aligned} \tag{49}$$

which satisfies

$$\begin{aligned} \hat{\boldsymbol{s}}_{id}\hat{\boldsymbol{C}}_{id}^{(\times)} &= \hat{\boldsymbol{s}}_{id}\hat{\boldsymbol{C}}_{id_1} \cdot \boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &= \mu_1\mu_2\hat{\boldsymbol{s}}_{id}\boldsymbol{G}_{K(m+1)} + \hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_1}\boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) \\ &\quad + \mu_1\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_2} \end{aligned} \tag{50}$$

So, we know that

$$\tilde{\beta}_{eval} = \left(\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_1}\boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) + \mu_1\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_2}\right) \tag{51}$$

for one homomorphic operation. Referring to (48), we can compute

$$\begin{aligned} \left\|\tilde{\beta}_{eval}\right\|_\infty &= \left\|\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_1}\boldsymbol{G}_{K(m+1)}^{-1}(\hat{\boldsymbol{C}}_{id_2}) + \mu_1\hat{\boldsymbol{s}}_{id}\boldsymbol{E}_{id_2}\right\|_\infty \\ &\leq K^2 B_\chi \ell(m+1) \end{aligned} \tag{52}$$

According to the analysis of noise growth in computing circuits in [3] and [7], we could evaluate a circuit of depth L and a designed upper bound of users K while keeping the error magnitude at most

$$\left\|\tilde{\beta}_{final}\right\|_\infty \leq K^{2^L} B_\chi \ell(m+1) \tag{53}$$

Our decryption algorithm works correctly as long as its error is smaller than $q/4$, hence it suffices to choose the modulus

$$q \geq 4K^{2^L} B_\chi \ell(m+1) \tag{54}$$

TABLE 3. Comparison of MIBFHE schemes.

Scheme	Dimension	Secret key	Auxilary ciphertext	Noise expansion	Must fresh ciphertext	Multi-hop
[6]	$6n\log q$	$n\ell^2$	$\tilde{O}(n^4L^4)$	$1+n\ell$	YES	NO
[21]	$6n\log q$	$n\ell^2$	$\tilde{O}\left(kn^3(K+L)^5\right)$	$1+7n\ell$	YES	YES
[22]	$2n\log q$	$n\ell$	$\tilde{O}\left(n^4L^6\right)$	$(1+2n)+3n(1+2n\ell)^3$	YES	NO
Ours	$2n\log q$	$n\ell$	$\tilde{O}\left(n^2L^4\right)$	K	NO	NO

We define that all sizes are in bits. Here k denotes the actual number of operators involved. The meanings of other symbols are the same as in Table II.

C. SECURITY

Theorem 3: When $m \geq 2n\log q$, our MIBFHE scheme in this section is IND-sID-CPA secure in the random oracle model if the $\mathrm{LWE}_{n,q,\chi}$ assumption holds.

Proof of Theorem 3. We prove the security of our MIBFHE scheme using a sequence of hybrid games between challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

Assume that the identity id^* is the target identity that adversary $\mathcal{A}$ plans to attack and $Adv[i]$ represents adversary's advantage in Game i. The steps are as follows:

Game 0. This is the standard original IND-sID-CPA game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

Game 1. Compared with Game 0, challenger $\mathcal{C}$ changes the way to generate the MPK matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and chooses a uniformly random matrix $\boldsymbol{A''} \in \mathbb{Z}_q^{n\times m}$. Then output $MPK = (\boldsymbol{A''}, H)$.

According to Lemma 1, we know that matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ in Game 0 is statistically indistinguishable from $\boldsymbol{A''} \in \mathbb{Z}_q^{n\times m}$ in Game 1. Thus, adversary $\mathcal{A}$ in polynomial time cannot distinguish between Game 1 and Game 0 with non-negligible advantages, that is,

$$|Adv\,[1] - Adv\,[0]| = negl(\lambda) \tag{55}$$

Game 2. Compared with Game 1, challenger $\mathcal{C}$ changes the way to generate the public key and secret key.

(1). The challenger $\mathcal{C}$ runs $TrapGen(1^n, 1^m, q)$ to generate the gadget matrix $\boldsymbol{G}_n$ and the trapdoor matrix $\boldsymbol{R_G}$ for lattice $\Lambda_q^{\perp}(\boldsymbol{G}) = \left\{\boldsymbol{y} \in \mathbb{Z}_q^{n\ell} : \boldsymbol{Gy} = \boldsymbol{0} \bmod q\right\}$.

(2). The adversary $\mathcal{A}$ sends identity set $\{id_\alpha\}_{\alpha\in poly(\lambda)}$ to challenger $\mathcal{C}$ for hash queries.

(3). The challenger $\mathcal{C}$ chooses a uniformly random vector $\boldsymbol{u}''_{id_\alpha} \in \mathbb{Z}_q^n$ and lets $\boldsymbol{A}''_{id_\alpha} = \left[\boldsymbol{u}''_{id_\alpha} \middle\| \boldsymbol{A''}\right]^{\mathrm{T}} \in \mathbb{Z}_q^{(m+1)\times n}$. If $id^* \in \{id_\alpha\}_{\alpha\in poly(\lambda)}$, the game ends. Otherwise, challenger $\mathcal{C}$ runs the $\mathrm{SampleD}(\boldsymbol{A}''_{id_\alpha}, \boldsymbol{R_G}, \boldsymbol{u}''_{id_\alpha}, \sigma)$ to generate $\boldsymbol{t}''_{id_\alpha}$ which satisfies $\boldsymbol{A}''_{id_\alpha} \cdot \boldsymbol{t}''_{id_\alpha} = \boldsymbol{u}''_{id_\alpha} \bmod q$. Let $\boldsymbol{s}''_{id_\alpha} = \left(1, -\boldsymbol{t}''^{\mathrm{T}}_{id_\alpha}\right)$ and return it to adversary $\mathcal{A}$.

The challenger $\mathcal{C}$ makes $(id_\alpha, \boldsymbol{u}''_{id_\alpha}, \boldsymbol{t}''_{id_\alpha}) \in$ store. Therefore, for the same id, adversary $\mathcal{A}$ gets the same result on each query.

Lemma 5 [14]: Let $n = poly(\lambda) \geq 1$, q be a prime and $m = O(n\log q) \geq 2n\log q$. Then, for all but a $2q^{-n}$ fraction of all $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and any $r \geq \omega(\sqrt{\log m})$, the distribution of the syndrome $\boldsymbol{u} = \boldsymbol{At} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$, where $\boldsymbol{t} \leftarrow D_{\mathbb{Z}^m,r}$.

According to Lemma 5, vectors $\boldsymbol{u}''_{id_\alpha} \in \mathbb{Z}_q^n$ in Game 2 and $\boldsymbol{u}_{id} \in \mathbb{Z}_q^n \leftarrow H(id)$ in Game 1 are statistically indistinguishable. Hence, the public key $\boldsymbol{A}''_{id} = \left[\boldsymbol{u}''_{id} \middle\| \boldsymbol{A''}\right]^{\mathrm{T}} \in \mathbb{Z}_q^{(m+1)\times n}$ in Game 2 and $\boldsymbol{A}_{id} = \left[\boldsymbol{u}_{id} \middle\| \boldsymbol{A''}\right]^{\mathrm{T}} \in \mathbb{Z}_q^{(m+1)\times n}$ in Game 1 are also statistically indistinguishable.

According to Lemma 2, the secret key $\boldsymbol{t}''_{id}$ in Game 2 and $\boldsymbol{t}''_{id}$ in Game 1 are all generated by the $\mathrm{SampleD}(\cdot)$ algorithm. And both of them are close to the discrete Gaussian distribution with the same Gaussian parameter. Hence, the secret key $\boldsymbol{t}''_{id}$ in Game 2 and $\boldsymbol{t}''_{id}$ in Game 1 are also statistically indistinguishable.

Therefore, the public key and secret key in Game 2 and Game 1 are all statistically indistinguishable. The adversary $\mathcal{A}$ in polynomial time cannot distinguish between Game 2 and Game 1 with non-negligible advantages, that is,

$$|Adv\,[2] - Adv\,[1]| = negl(\lambda) \tag{56}$$

Game 3. The adversary $\mathcal{A}$ chooses a pair of messages (μ_0, μ_1) for challenger $\mathcal{C}$. Compared with Game 2, challenger $\mathcal{C}$ changes the way to generate the extended ciphertext. The challenger $\mathcal{C}$ chooses a uniformly random matrix $\hat{\boldsymbol{P}} \in \mathbb{Z}_q^{K(m+1)\times K(m+1)\ell}$ and $\hat{\boldsymbol{E}} \in \chi^{K(m+1)\times K(m+1)\ell}$ to generate a ciphertext $\hat{\boldsymbol{C}}_{id^*} = \hat{\boldsymbol{P}} + \mu_b \boldsymbol{G}_{K(m+1)} + \hat{\boldsymbol{E}}$ encrypting μ_b where $b \in \{0, 1\}$ and sends it to adversary $\mathcal{A}$.

According to the security analysis of our MKFEH scheme in Section III, it can be seen that the extended ciphertext in Game 3 and Game 2 are computationally indistinguishable based on the $\mathrm{LWE}_{n,q,\chi}$ problem. In the same way, we can get

$$|Adv\,[3] - Adv\,[2]| = negl(\lambda) \tag{57}$$

According to (55), (56), (57), the advantage of adversary $\mathcal{A}$ in our MIBFHE scheme can be negligible in polynomial time, and the security reduction is based on the $\mathrm{LWE}_{n,q,\chi}$ assumption. Thus, this scheme is IND-sID-CPA secure in the random oracle, and the proof is over.

D. EFFICIENCY ANALYSIS OF MIBFHE SCHEME

We compare our MIBFHE scheme with current MIBFHE schemes. The comparison results are shown in Table 3.

Through the analysis in Table 3, the following conclusions can be drawn.

- Compared with MIBFHE schemes in [6], [21], and [22], our scheme inherits the advantages of our MKFHE scheme in Section III. Users do not need to generate the fresh ciphertext in advance, but can directly generate the extended ciphertext that can perform homomorphic operations through the encryption algorithm, and the scale of auxiliary ciphertexts introduced is smaller. This makes our scheme more concise. In addition, the noise expansion of our scheme is significantly reduced, and it also has certain advantages in lattice dimension and secret key size, so we can also choose a smaller modulus q. These parameters make our scheme more efficient.
- Compared with the MIBFHE scheme in [21], our scheme has obvious advantages in each parameter index. However, compared with the multi-hop scenario in [21], our scheme is only single-hop. How to extend the single-hop scenario to the multi-hop scenario under the premise of maintaining the technical advantages of our scheme is also worthy of research and attention.

VI. CONCLUSION

We construct an MKFHE scheme using a new method (the decomposition method) and then construct an MIBFHE scheme, which provides a new idea and reference for cryptographers to study FHE. We tried to extend our MIBFHE scheme to multi-hop scenarios, but if the number of users is not fixed, how to use ''the decomposition method'' in this paper to extend the evaluated ciphertexts still needs a better solution. Although the existing multi-hop MKFHE scheme [8] can achieve this goal, the scheme construction is too complicated. How to construct a more concise and efficient multi-hop MIBFHE scheme still needs more research.

In addition to its combination with other cryptographic primitives, FHE has also been widely studied in practical applications, such as biometric authentication, medical data acquisition and protection, etc. With the rapid development of virtual reality (VR), the security of large amounts of private data stored internally is a growing concern. Implementing user authentication mechanisms in VR is a crucial step in resisting unauthorized access [29], [30], [31], [32], [33], [34]. We notice that Zhu et al. [35] proposed a novel user authentication scheme using auditory-pupillary response and further reduced the authentication time. In the authentication on VR devices, we may use FHE to get more secure and faster authentication in the biometric field. In the field of medical data, FHE also plays an important role not only in preventing attackers from obtaining user identity tags but also in providing high efficiency in data integration between sensors and server [36]. Gu et al. [37] proposed an aggregate signature scheme based on a linearly homomorphic signature for electronic healthcare systems which realizes double data compression. The application of the homomorphic signature mechanism in medical data deserves the attention of cryptographers.

REFERENCES

[1] R. L. Rivest, L. Adleman, and M. L. Dertouzos, ''On data banks and privacy homomorphisms,'' *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–180, 1978.

[2] C. Gentry, ''Fully homomorphic encryption using ideal lattices,'' in *Proc. 41st Annu. ACM Symp. Theory Comput. (STOC)*, Bethesda, MD, USA, 2009, pp. 169–178.

[3] C. Gentry, A. Sahai, and B. Waters, ''Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based,'' in *Advances in Cryptology–(CRYPTO)*. Berlin, Germany: Springer, 2013, pp. 75–92.

[4] J. Alperin-Sheriff and C. Peikert, ''Faster bootstrapping with polynomial error,'' in *Advance in Cryptology–(CRYPTO)*. Santa Barbara, CA, USA: Springer, 2014, pp. 297–314.

[5] A. López-Alt, E. Tromer, and V. Vaikuntanathan, ''On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption,'' in *Proc. 44th Annu. ACM Symp. Theory Comput. (STOC)*, May 2012, pp. 1219–1234.

[6] M. Clear and C. McGoldrick, ''Multi-identity and multi-key leveled FHE from learning with errors,'' in *Advances in Cryptology–(CRYPTO)*, vol. 9216. Berlin, Germany: Springer, 2015, pp. 630–656.

[7] P. Mukherjee and D. Wichs, ''Two round multiparty computation via multi-key FHE,'' in *Advances in Cryptology–(EUROCRYPT)*, vol. 9666. Berlin, Germany: Springer, 2016, pp. 735–763.

[8] C. Peikert and S. Shiehian, ''Multi-key FHE from LWE, revisited,'' in *Theory of Cryptography–(TCC)*, vol. 9986. Berlin, Germany: Springer, 2016, pp. 217–238.

[9] Y. Huang, K. Wu, and M. Chen, ''Fully dynamic multi-key FHE without Gaussian noise,'' *IEEE Access*, vol. 9, pp. 50639–50645, 2021.

[10] P. Ananth, A. Jain, Z. Jin, and G. Malavolta, ''Multi-key fully-homomorphic encryption in the plain model,'' in *Theory of Cryptology–(TCC)*, vol. 12550. Raleigh, NC, USA: Springer, 2020, pp. 28–57.

[11] C. Biswas and R. Dutta, ''Dynamic multi-key FHE in symmetric key setting from LWE without using common reference matrix,'' *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 2, pp. 1241–1254, Mar. 2021, doi: 10.1007/s12652-021-02980-w.

[12] C. Biswas and R. Dutta, ''Secure and efficient multi-key FHE scheme supporting multi-bit messages from LWE preserving non-interactive decryption,'' *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 12, pp. 16451–16464, 2023, doi: 10.1007/s12652-022-03864-3.

[13] A. Shamir, ''Identity-based cryptosystems and signature schemes,'' in *Advances in Cryptology–(CRYPTO)*. Tbilisi, GA, USA: Springer, 1984, pp. 47–53.

[14] C. Gentry, C. Peikert, and V. Vaikuntanathan, ''Trapdoors for hard lattices and new cryptographic constructions,'' in *Proc. 40th Annu. ACM Symp. Theory Comput. (STOC)*, Victoria, QC, Canada, 2008, pp. 197–206.

[15] M. Clear and C. McGoldrick, ''Bootstrappable identity-based fully homomorphic encryption,'' in *Cryptology and Network Security–(CANS)*. Heraklion, Greece: Springer, 2014, pp. 1–19.

[16] R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan, ''Chosen-ciphertext secure fully homomorphic encryption,'' in *Public-Key Cryptography–(PKC)*. Amsterdam, The Netherlands: Springer, 2017, pp. 213–240.

[17] X. Wang, B. Wang, B. Liang, and R. Xue, ''A more compact multi-id identity-based FHE scheme in the standard model and its applications,'' *Sci. China Inf. Sci.*, vol. 62, no. 3, pp. 186–188, 2018, doi: 10.1007/s11432-017-9412-3.

[18] T. Shen, F. Wang, K. Chen, K. Wang, and B. Li, ''Efficient leveled (Multi) identity-based fully homomorphic encryption schemes,'' *IEEE Access*, vol. 7, pp. 79299–79310, 2019.

[19] T. Pal and R. Dutta, ''Chosen-ciphertext secure multi-identity and multi-attribute pure FHE,'' in *Cryptology and Network Security–(CANS)*. Vienna, Austria: Springer, 2020, pp. 387–408.

[20] T. Shen, F. Wang, K. Chen, Z. Shen, and R. Zhang, ''Compressible multikey and multi-identity fully homomorphic encryption,'' *Secur. Commun. Netw.*, vol. 2021, Mar. 2021, Art. no. 6619476, doi: 10.1155/2021/6619476.

[21] W. Liu, F. Wang, J. Jin, K. Chen, and Z. Shen, "Leveled multi-hop multi-identity fully homomorphic encryption," *Secur. Commun. Netw.*, vol. 2022, Mar. 2022, Art. no. 1023439, doi: 10.1155/2022/1023439.
[22] H. Fan, R. Huang, and F. Luo, "Efficient multi-identity full homomorphic encryption scheme on lattice," *Appl. Sci.*, vol. 13, no. 10, p. 6343, May 2023, doi: 10.3390/app13106343.
[23] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," *SIAM J. Comput.*, vol. 45, no. 3, pp. 882–929, Jan. 2016, doi: 10.1109/FOCS.2013.13.
[24] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Advances in Cryptology–(EUROCRYPT)*. Cambridge, U.K.: Springer, 2012, pp. 700–718.
[25] F. Wang and K. Wang, "Fully homomorphic encryption with auxiliary inputs," in *Proc. INSCRYPT*. Beijing, China: Springer, 2015, pp. 220–238.
[26] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Advances in Cryptology–(EUROCRYPT)*. Paris, France: Springer, 2010, pp. 553–572.
[27] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, Harrisburg, PA, USA, 1996, pp. 99–108.
[28] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, Feb. 2009, doi: 10.1145/1568318.1568324.
[29] Z. Zhang, Y. Wang, and K. Yang, "Strong authentication without temper-resistant hardware and application to federated identities," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Sacramento, CA, USA, 2020, pp. 23–26.
[30] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 193–208, Jan. 2023, doi: 10.1109/TDSC.2021.3129512.
[31] Y. C. Feng and P. C. Yuen, "Binary discriminant analysis for generating binary face template," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 613–624, Apr. 2012, doi: 10.1109/TIFS.2011.2170422.
[32] D. Wang, Q. Gu, X. Huang, and P. Wang, "Understanding human-chosen PINs: Characteristics, distribution and security," in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, Abu Dhabi, United Arab Emirates, 2017, pp. 372–385.
[33] K. Sadeghi, A. Banerjee, J. Sohankar, and S. K. S. Gupta, "Geometrical analysis of machine learning security in biometric authentication systems," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Cancun, Mexico, Dec. 2017, pp. 309–314.
[34] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1338–1351, Mar. 2022, doi: 10.1109/TDSC.2020.3022797.
[35] H. Zhu, M. Xiao, D. Sherman, and M. Li, "SoundLock: A novel user authentication scheme for VR devices using auditory-pupillary response," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Sacramento, CA, USA, 2023, pp. 1–18.
[36] M. Al-Zubaidie, Z. Zhang, and J. Zhang, "REISCH: Incorporating lightweight and reliable algorithms into healthcare applications of WSNs," *Appl. Sci.*, vol. 10, no. 6, p. 2007, Mar. 2020, doi: 10.3390/app10062007.
[37] Y. Gu, L. Shen, F. Zhang, and J. Xiong, "Provably secure linearly homomorphic aggregate signature scheme for electronic healthcare system," *Mathematics*, vol. 10, no. 15, p. 2588, Jul. 2022, doi: 10.3390/math10152588.

GUANGSHENG TU was born in Zhengyang, Henan, China, in 1992. He received the B.S. and M.S. degrees in military cryptography from the Engineering University of People's Armed Police, Shaanxi, China, in 2017 and 2019, respectively. He is currently an Assistant Engineer of confidential cryptography with Non-Commissioned Officer Academy, People's Armed Police, China. His research interests include fully homomorphic encryption, identity based encryption, and cryptographic network system construction.

WENCHAO LIU was born in Wuwei, China, in 1994. He received the M.S. degree in computer science and technology from the Engineering University of People's Armed Police, Shaanxi, China, in 2019, where he is currently pursuing the Ph.D. degree. His main research interests include fully homomorphic encryption and information security.

TANPING ZHOU was born in Yingtan, China, in 1989. He received the B.S., M.S., and Ph.D. degrees in military cryptography from the Engineering University of People's Armed Police, Shaanxi, China, in 2012, 2014, and 2018, respectively. He is currently an Associate Professor with the Engineering University of People's Armed Police. His research interests include fully homomorphic encryption and encryption scheme based on lattice.

XIAOYUAN YANG was born in Xiangtan, China, in 1959. He is currently the Ph.D. Supervisor of the Engineering University of People's Armed Police, Shanxi, China. His research interests include information security and cryptology.

FAN ZHANG was born in Jiaxing, China, in 1981. She received the M.S. degree from Zhejiang University, Zhejiang, China, in 2009. She is currently a Professor with the Non-Commissioned Officer Academy, People's Armed Police, China. Her research interests include computer technology and new communication equipment.