Hindawi
Security and Communication Networks
Volume 2021, Article ID 6619476, 14 pages
https://doi.org/10.1155/2021/6619476

Research Article

Compressible Multikey and Multi-Identity Fully Homomorphic Encryption

Tongchen Shen,[1,2] **Fuqun Wang**,[1,2,3] **Kefei Chen**,[1,2,3] **Zhonghua Shen**,[1] **and Renjun Zhang**[1]

[1]*Department of Mathematics, Hangzhou Normal University, Hangzhou, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin, China*
[3]*Westone Cryptologic Research Center, Beijing, China*

Correspondence should be addressed to Fuqun Wang; fqwang@hznu.edu.cn

Received 3 October 2020; Revised 4 February 2021; Accepted 11 February 2021; Published 4 March 2021

Academic Editor: Neetesh Saxena

With the development of new computing models such as cloud computing, user's data are at the risk of being leaked. Fully homomorphic encryption (FHE) provides a possible way to fundamentally solve the problem. It enables a third party who does not know anything about the secret key and plaintexts to homomorphically perform any computable functions on the corresponding ciphertexts. In 2009, Gentry proposed the first FHE scheme. After that, its inefficiency has always been a bottleneck of the development of practical schemes and applications. At TCC 2019, Gentry and Halevi proposed the first compressible FHE scheme that enables the ratio of plaintext size to the ciphertext size (i.e., the compression rate) to reach $1 - \varepsilon$ for any small $\varepsilon > 0$ under the standard learning with errors (LWE) assumption. However, it is only a single-key one, where the homomorphic evaluation can only be performed over ciphertexts encrypted under the same key. Compared with single-key FHE, multikey FHE is more practical. Multikey FHE enables ciphertexts encrypted under different public keys to be homomorphically computed without having to decrypt these ciphertexts using their own private keys. In addition, in a multi-identity FHE scheme, only identity information and public parameters are required when encrypting, which simplifies certificate-based key management in public key infrastructure. In this paper, a new compressible ciphertext expansion technique is proposed. Then, we use this technique to construct a compressible multikey FHE scheme and a compressible multi-identity FHE scheme to overcome the bottleneck of bandwidth inefficiency in the multikey and multi-identity settings. The two schemes proposed in this paper make it possible that the objects of homomorphic operation can be the ciphertexts encrypted under different keys or different identities before compression, thus solving the single-key defect of the work of Gentry and Halevi.

1. Introduction

We are quickly entering a new digital era where huge amounts of data will be stored and operated remotely in powerful cloud servers. Cloud computing is a popular option for people and businesses for a number of reasons including cost savings, increased productivity, and performance. With all of these advantages, however, it also raises grave security issues and challenges. User's sensitive information, such as medical data and organizational secrets, is vulnerable to be leaked to cloud providers and even clients. Thus, it is prudent for users to encrypt their data before transmitting to the cloud.

Fully homomorphic encryption (FHE) provides a fundamental method to solve the issue, making it possible to perform arbitrary operations over encrypted data. In particular, FHE enables the third party who does not know anything about the secret key and plaintexts to homomorphically evaluate any computable functions on the corresponding ciphertexts. Since 2009, when Gentry proposed the first FHE scheme [1], FHE has been a research hot spot. Various FHE schemes were then proposed [2–4].

However, conventional FHE is only a single-key one, where the homomorphic evaluation can only be performed over ciphertexts encrypted under the same key. Compared with single-

key FHE, multikey FHE is more practical. In many scenarios, several users (often mutually distrusting) want to compute a joint function on their collective data, which have been encrypted and stored in the cloud. For example, two medical companies want the cloud to calculate some statistical information on their medical databases that are encrypted using their own public keys and stored on the cloud. In the meantime, they do not want to leak anything except the final result to others. Single-key FHE cannot deal with these situations, while multikey FHE enables ciphertexts encrypted under different public keys to be homomorphically computed without having to decrypt these ciphertexts using their own private keys.

At STOC 2012, López-Alt, Tromer and Vaikuntanathan [5] proposed the notion of multikey FHE (MKFHE) and they provided the first MKFHE candidate from NTRU-based cryptography. At CRYPTO 2015, Clear and McGoldrick [6] constructed an MKFHE scheme based on the learning with errors (LWE) assumption, which was simplified by Mukherjee and Wichs [7] and then followed by [8–11]. Very recently, Chen, Chillotti, and Song proposed an MKFHE scheme [12] based on TFHE [13], and they were the first to implement an MKFHE scheme.

Identity-based FHE (IBFHE) is an identity-based version of FHE. The first IBFHE scheme was achieved by Gentry, Sahai, and Waters [4] using a special compiler, which allows all lattice-based IBE schemes [14, 15] to be compiled into IBFHE schemes. Multi-identity FHE (MIFHE) is an identity-based version of MKFHE. In an MIFHE scheme, only identity information and public parameters are required when encrypting, which simplifies certificate-based key management in public key infrastructure. Clear and McGoldrick [6] also constructed the first MIFHE scheme in the random oracle model based on the GPV-IBE scheme [14].

However, there is always a question plaguing FHE researchers. How bandwidth-efficient can FHE be? Although it is quite easy to achieve public key encryption (PKE) with nearly no loss in bandwidth, the same is nontrivial in the FHE case because there should be much more redundancy in an FHE ciphertext to support homomorphism, which means much more bandwidth waste.

Some works managed to partly improve the inefficiency by using the "dimension-modulus reduction" technique [16, 17] to make ciphertexts have smaller dimensions and coefficients, and some by using the "ciphertext packing" technique [18–20] managed to encrypt an array of plaintexts in a single ciphertext and even encrypt a matrix of plaintexts in a single ciphertext [21, 22]. But none of them achieved a "sufficiently high" rate; that is, none of them break the rate-1/2 bottleneck. Very recently, Gentry and Halevi [23] proposed the first compressible FHE scheme, allowing compressing many ciphertexts into a compressed one, so that the compression rate can reach $1 - \varepsilon$ for any small $\varepsilon > 0$. A concurrent work of Brakerski et al. [24] also achieved compressible FHE, but their work is more general and is unlikely to yield practical schemes for applications.

A natural question is the following: how can we extend their idea of compressible single-key FHE into compressible MKFHE or compressible MIFHE to achieve an optimal compression rate even in the multikey and multi-identity cases? In other words, we want to construct a compressible MKFHE scheme and a compressible MIFHE scheme that enable ciphertexts encrypted under different public keys and different identities to perform homomorphic operations without having to decrypt these ciphertexts using their own private keys. At the same time, expanded ciphertexts (under the combined key) are supposed to be compressed to achieve an optimal compression rate. In this work, we focus on this problem.

1.1. Technical Overview. We now give a technical overview of our new compressible expansion algorithm, which is the key step towards our constructions of compressible MKFHE and compressible MIFHE. We start with Gentry and Halevi's basic ideas of constructing compressible single-key FHE [23]. Then, we give a discussion about the difficulties encountered when trying to construct compressible multikey (multi-identity) FHE, which can be solved using our new compressible expansion algorithm.

1.1.1. Compressible Single-Key FHE. We begin with a brief description of the approach of Gentry and Halevi. At a very high level, their approach of building a compressible single-key FHE is based on two ideas:

Using matrix versions of LWE and GSW-FHE [4]. Firstly, they noticed that the conventional GSW-FHE uses only one "slot" to place a message bit. To achieve an optimal compression rate, more message "slots" must be utilized. Their solution is to use LWE with matrix secrets, which yields a matrix version of GSW-FHE. Then, a compression algorithm is added after evaluation, so that multiple ciphertexts can be compressed into one. Besides, the secret key of matrix version is used in the decryption algorithm. The compression rate is now bounded by $\alpha = ((n \times n)/((n + \text{small}) \times (n + \text{small})\ell)) \approx (1/\ell) = (1/\lceil \log q \rceil) \leq (1/2)$.

Using a "nearly square" gadget matrix. The second idea is to transform a "fat" compressed ciphertext matrix into a "nearly square" compressed ciphertext matrix using a "nearly square" gadget matrix $\mathbf{H}$ with some special properties, which comes with almost no dimension expansion. On the other hand, attributed to $\mathbf{H}$, the redundancy in ciphertexts enables us to erase the noise and finally recover the message matrix. Using $\mathbf{H}$ instead of the original gadget matrix $\mathbf{G}$, the compression rate is now $\alpha = ((n \times n)/((n + \text{small}) \times (n + \text{small}))) \approx 1$, an optimal compression rate!

More information can be found later in Section 2.4.

1.1.2. Compressible Multikey (Multi-Identity) FHE. We propose the first compressible multikey (multi-identity) FHE that achieves an optimal compression rate via a new compressible expansion algorithm. At a high level, the key observation of our compressible expansion algorithm is that we can transform a secret key vector and a "fat" ciphertext matrix into a "nearly square" secret key matrix and a "nearly square"

ciphertext matrix, even in the multikey and multi-identity FHE cases.

We note that it is nontrivial to utilize the powerful functionality of the nearly square gadget matrix $\mathbf{H}$ described above in the multikey (or multi-identity) setting. The idea of using the matrix version of secret key does not directly work because all of the secret matrices for different users, say 2 users totally, are required to decrypt the expanded ciphertext. Hence, the combined key matrix $\widehat{S} = [\mathbf{S}_1 \| \mathbf{S}_2]$ is a "fat" one, rather than a "nearly square" one. To the best of our knowledge, all of the existing GSW-based multikey (or multi-identity) FHE schemes combine secret keys like this. To make the combined secret key matrix more "square," we must decrease the number of columns of the combined secret key matrix.

Our solution is to delete the identity matrix in every single secret key matrix because it seems like the most insignificant part compared with the LWE secrets. After all, everybody knows there is an identity matrix in the secret key matrix but no one can recover the LWE secrets after discarding them. The combined secret key matrix $\widehat{S} = [\overline{S}_1 \| \overline{S}_2] = [\mathbf{I}_n \| \mathbf{S}_1 \| \mathbf{I}_n \| \mathbf{S}_2] \in \mathbb{Z}_q^{n \times 2(n+r)}$ now becomes $\widehat{S} = [\mathbf{I}_n \| \mathbf{S}_1 \| \mathbf{S}_2] \in \mathbb{Z}_q^{n \times} (n + 2r)$, a nearly square one again.

Meanwhile, the structure of corresponding expanded ciphertexts must be tailored to fit the combined secret key. Traditionally, the expanded ciphertexts are supposed to be partitioned into 2^2 blocks, each with the same size. Since we have discarded some columns of the combined secret key, it is fair to also dump some rows of expanded ciphertexts. In particular, the expanded ciphertexts are now partitioned into $(2 + 1) \times 2$ blocks, each with different size, i.e.,

$$\begin{bmatrix} \mathbf{C}^{1,1} & \mathbf{X}^1 \\ 0 & \mathbf{C}^{2,2} \end{bmatrix} \longrightarrow \begin{bmatrix} \mathbf{C}^{1,1} & \mathbf{X}^1 \\ \mathbf{C}^{2,1} & \mathbf{X}^2 \\ 0 & \mathbf{C}^{3,3} \end{bmatrix}. \tag{1}$$

However, abandoning some rows disrupts the structure of the G-trapdoor, which bears great responsibility on evaluation and decryption. Here comes our second idea: restoring the disrupted structure of the G-trapdoor. In particular, the expanded ciphertexts should be partitioned into $(2 + 1) \times (2 + 1)$ blocks rather than $(2 + 1) \times 2$ blocks as above. Our construction allows user 1 to generate some additional helper information $\mathbf{X}^1$ and $\mathbf{X}^2$ associated with the ciphertext $\mathbf{C}^{i,j}$ when encrypting, and releasing all of them does not compromise semantic security. We define the expanded ciphertext as

$$\widehat{\mathbf{C}} = \begin{bmatrix} \mathbf{C}^{1,1} & \mathbf{C}^{1,2} & \mathbf{X}^1 \\ \mathbf{C}^{2,1} & \mathbf{C}^{2,2} & \mathbf{X}^2 \\ 0 & 0 & \mathbf{C}^{3,3} \end{bmatrix} \in \mathbb{Z}_q^{(n+2r) \times (n+2r)\ell}, \tag{2}$$

so that

$$\begin{aligned} \widehat{S}\widehat{\mathbf{C}} &= \left[\mathbf{C}^{1,1} + \mathbf{S}_1\mathbf{C}^{2,1} \middle\| \mathbf{C}^{1,2} + \mathbf{S}_1\mathbf{C}^{2,2} \middle\| \mathbf{X}^1 + \mathbf{S}_1\mathbf{X}^2 + \mathbf{S}_2\mathbf{C}^{3,3}\right] \\ &\overset{\text{want}}{\approx} \left[\mu_1\mathbf{I}_n\mathbf{G}_n \middle\| \mu_1\mathbf{S}_1\mathbf{G}_r \middle\| \mu_1\mathbf{S}_2\mathbf{G}_r\right] \\ &\triangleq [\text{Part 1} \| \text{Part 2} \| \text{Part 3}]. \end{aligned} \tag{3}$$

Part 1 and Part 2 are nothing special and there are many ways to achieve them [6, 7, 11]. The major difficulty lies in Part 3, i.e., $\mathbf{X}^1 + \mathbf{S}_1\mathbf{X}^2 + \mathbf{S}_2\mathbf{C}^{3,3} \overset{\text{want}}{\approx} \mu_1\mathbf{S}_2\mathbf{G}_r$. The right-hand side must come from $\mathbf{S}_2\mathbf{C}^{3,3}$, since the randomness $\mathbf{S}_2$ can only come from this. In terms of normal routine, $\mathbf{S}_2\mathbf{C}^{3,3} - \mu_1\mathbf{S}_2\mathbf{G}_r \approx \mathbf{A}_2\mathbf{R}$, where $\mathbf{A}_2$ is the public key of user 2 and $\mathbf{R}$ is the encryption randomness. Thus, the only thing left is to construct $\mathbf{X}^1 + \mathbf{S}_1\mathbf{X}^2 \overset{\text{want}}{\approx} -\mathbf{A}_2\mathbf{R}$.

Similarly, the expanded ciphertext for user 2 can be created, and then we can perform regular GSW-like homomorphic evaluations. Our new expansion technique leads to the first compressible MKFHE and the first compressible MIFHE. The relationship between them is just like the relationship between Regev-PKE [25] and GPV-IBE [14]. A more formal and detailed description can be found later in Sections 3 and 4.

1.2. Contribution. In this paper, a new compressible expansion algorithm is proposed, which makes it possible to construct the first compressible MKFHE scheme, as well as the first compressible MIFHE scheme, while the construction of Gentry and Halevi only works for compressible single-key FHE. Informally, the following theorem outlines our main result. A formal version will be presented later in Sections 3 and 4.

Theorem 1. *For any $\varepsilon = \varepsilon(\lambda) > 0$, there exist a rate-$(1 - \varepsilon)$ compressible multikey fully homomorphic encryption scheme that is semantically secure and a rate-$(1 - \varepsilon)$ compressible multi-identity fully homomorphic encryption scheme that is selectively secure, both under the decisional learning with error assumption.*

Besides, as Gentry and Halevi noted, a compressible MKFHE scheme or a compressible MIFHE scheme enables messages always in a compressed state except when operating homomorphic evaluation for a short time. In particular, freshly encrypted ciphertexts are compressed after encryption. The compressed ciphertexts are homomorphically decompressed before performing required homomorphic operations. Finally, these evaluated ciphertexts are compressed again and stored or transmitted to corresponding users.

1.3. Paper Organization. Firstly, we recall some notations, definitions, and facts in Section 2. Then, we present our compressible MKFHE scheme in Section 3 and our compressible MIFHE scheme in Section 4, respectively. Finally, we conclude the paper in Section 5.

2. Preliminaries

There are some notations that we will use throughout this paper. Let λ denote the security parameter throughout the paper. Matrices are represented by bold uppercase letters (e.g., $\mathbf{A}, \mathbf{B}$), vectors are represented by bold lowercase letters (e.g., $\mathbf{a}, \mathbf{b}$), the i^{th}-entry of $\mathbf{a}$ is represented by the notation of $\mathbf{a}[i]$, the n-dimensional identity matrix is represented by $\mathbf{I}_n$, the concatenation of two matrices is represented by the notation of $[\mathbf{A} \| \mathbf{B}]$, and the concatenation of two vectors is represented by the notation of $[\mathbf{a}, \mathbf{b}]$.

2.1. Lattice Trapdoor. Our constructions will make use of the following results including the **G**-trapdoor generation algorithm, sub-Gaussian sampling algorithm, and Gaussian sampling algorithm [26]. We only summarize the following lemmas here while neglecting the details of implementation because they are not strictly required.

Let $\mathcal{D}$ be a distribution over $\{-1, 0, 1\}$ which outputs 0 with a probability of 0.5, -1 with a probability of 0.25, and 1 with a probability of 0.25 (the notation of $\mathcal{D}$ will be used throughout this paper to denote this distribution).

Given an integer $q > 2$, for any $n \in \mathbb{Z}^+$, define $\mathbf{G}_n \stackrel{\Delta}{=} \mathbf{I}_n \otimes \mathbf{g}^T \in \mathbb{Z}_q^{n \times n\lceil \log q \rceil}$, where $\mathbf{g}^T = (1, 2, 2^2, \ldots, 2^{\lceil \log q \rceil - 1})$. The notation of $\mathbf{G}_n$ will be used throughout this paper.

Lemma 1 (see [26]). *Let n, m_0, m_1, m, q, ℓ be positive integers such that $q = q(n), \ell = \lceil \log q \rceil, m_0 = n\ell + O(n)$, $m_1 = n\ell$, and $m = m_0 + m_1$. For $\mathbf{A}_0 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n \times m_0}$, invertible $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, and $\mathbf{R} \leftarrow \mathcal{D}^{m_0 \times m_1}$, there exists an efficiently randomized algorithm GenTrap $(\mathbf{A}_0, \mathbf{H})$ to generate a matrix $\mathbf{A} \stackrel{\Delta}{=} [\mathbf{A}_0 \mathbf{H} \mathbf{G}_n - \mathbf{A}_0 \mathbf{R}] \in \mathbb{Z}_q^{n \times m}$ with trapdoor $\mathbf{R}$ and tag $\mathbf{H}$ such that $\mathbf{A}$ is $negl(n)$-far from uniform. The matrix $\mathbf{R}$ is called the **G**-trapdoor of $\mathbf{A}$ with tag $\mathbf{H}$.*

Lemma 2 (see [26, 27]). *Given any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times n'}$, there exists an efficiently randomized algorithm that samples a sub-Gaussian matrix $\mathbf{X}$ with some constant parameter $O(1)$ over $\mathbb{Z}_q^{n\ell \times n'}$ such that $\mathbf{X} = \mathbf{G}_n^{-1}(\mathbf{A})$, where the gadget matrix $\mathbf{G}_n$ is specified as above.*

Lemma 3 (see [26]). *Using the parameters described in Lemma 1, given a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$, there exists an efficient algorithm SampleD$(\mathbf{R}, \mathbf{A}_0, \mathbf{H}, \mathbf{u}, s)$ that samples a vector $\mathbf{t}$ over $\mathcal{D}_{\mathbb{Z}, s \cdot \omega(\sqrt{\log n})}^m$ for some $s \in \mathbb{R}$ and $\omega(\sqrt{\log n})$ satisfying $\mathbf{A} \cdot \mathbf{t} = \mathbf{u}$.*

2.2. LWE. The learning with errors (LWE) problem plays an important role in lattice-based cryptography. Although a ring-based version is more efficient, this work is confined to LWE. We define the decisional LWE (DLWE$_{n,m,q,\chi}$) problem as follows.

Definition 1. (DLWE) For positive integers n, q and an error distribution χ over $\mathbb{Z}$, let $A_{\mathbf{s},\chi}$ be the distribution of $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{s} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^n$, $e \leftarrow \chi$, and $\mathbf{a} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^n$; and additions are performed in $\mathbb{Z}_q$. Given $m = \text{poly}(n)$ independent instances sampled either from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ or from $A_{\mathbf{s},\chi}$, the decisional learning with errors (DLWE$_{n,m,q,\chi}$) problem is to determine which distribution these samples come from. The DLWE assumption says that these two distributions are computationally indistinguishable.

For simplicity, DLWE$_{n,m,q,\alpha}$ is frequently used to denote DLWE$_{n,m,q,\chi}$ and $A_{\mathbf{s},\alpha}$ to denote $A_{\mathbf{s},\chi}$ for $\chi = \mathcal{D}_{\mathbb{Z},\alpha q}$. It is generally known that the DLWE$_{n,m,q,\alpha}$ problem in the average case is as hard as approximation lattices problems with approximation factors of $\widetilde{O}(n/\alpha)$ in the worst case by quantum or classical reductions, when $\alpha q \geq 2\sqrt{n}$ [25, 26, 28, 29].

In this work, we rely on the matrix form of LWE denoted by MLWE.

Definition 2. (MLWE) For positive integers n, m, r, q and an error distribution χ over $\mathbb{Z}$, the MLWE$_{n,r,m,q,\chi}$ problem is to distinguish between $(\mathbf{B}, \mathbf{A} = \mathbf{S} \cdot \mathbf{B} + \mathbf{E})$, where $\mathbf{S} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n \times r}, \mathbf{B} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{r \times m}, \mathbf{E} \leftarrow \chi^{n \times m}$; and additions are performed in $\mathbb{Z}_q$, and $(\mathbf{B}, \mathbf{A})$ sampled uniformly at random from $\mathbb{Z}_q^{r \times m} \times \mathbb{Z}_q^{n \times m}$.

Standard hybrid arguments show that MLWE is equivalent to DLWE with at most an n factor loss in the distinguishing advantage.

2.3. GSW-FHE. In this subsection, a matrix version of GSW-FHE [4] is given.

GSW.Setup (1^λ): Given a security parameter λ, choose a lattice dimension $r = r(\lambda)$, a sufficiently large m modulus $q = q(\lambda)$, a number of LWE instances $n = n(\lambda)$, and a β_χ-bounded error distribution $\chi = \chi(\lambda)$. Set $\ell \stackrel{\Delta}{=} \lceil \log q \rceil$, $\overline{n} \stackrel{\Delta}{=} n + r$, and $m \stackrel{\Delta}{=} \overline{n}\ell$. Output public parameters params $= (r, n, \overline{n}, q, \ell, \beta_\chi, \chi, m)$.

GSW.KeyGen(params): Randomly choose $\mathbf{B} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{r \times m}$, $\mathbf{S} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n \times r}$, and $\mathbf{E} \leftarrow \chi^{n \times m}$. Compute $\mathbf{A} = \mathbf{SB} + \mathbf{E}$. Let the secret key and public key be $\text{SK} \stackrel{\Delta}{=} \overline{S} \stackrel{\Delta}{=} [\mathbf{I}_n \| \mathbf{S}] \in \mathbb{Z}_q^{n \times \overline{n}}$ and $\text{PK} \stackrel{\Delta}{=} \overline{A} \stackrel{\Delta}{=} \begin{bmatrix} -\mathbf{A} \\ \mathbf{B} \end{bmatrix} \in \mathbb{Z}_q^{\overline{n} \times m}$. Note that $\overline{SA} = \mathbf{E}$. Output (PK, SK).

GSW.Enc(params, PK, $\mu \in \{0, 1\}$): Choose random matrices $\overline{R} \stackrel{\$}{\leftarrow} \{0, 1\}^{m \times m}$. Set $\mathbf{C} \stackrel{\Delta}{=} \mu \mathbf{G}_{\overline{n}} + \overline{AR} \in \mathbb{Z}_q^{\overline{n} \times m}$. Output the ciphertext $\mathbf{C}$.

GSW.Dec(params, $\mathbf{C}$, SK): Let $\mathbf{c}$ be the penultimate column vector of $\mathbf{C}$ and let $\mathbf{s}$ be any column of $\overline{S}$. Output $\lfloor \langle \mathbf{s}, \mathbf{c} \rangle \rceil_2$, where $\lfloor \cdot \rceil_2 \colon \mathbb{Z}_q \longrightarrow \{0, 1\}$ maps a number $x \pmod q$ to 0 if it is closer to 0 and to 1 if it is closer to $2^{\ell-2}$.

GSW.Eval(params, $f, \widehat{\mathbf{C}}_1, \widehat{\mathbf{C}}_2, \ldots, \widehat{\mathbf{C}}_s$): To homomorphically evaluate a function f, perform homomorphic addition and multiplication in sequence.

Addition: $\mathbf{C}_{\text{add}} \stackrel{\Delta}{=} \mathbf{C}_1 + \mathbf{C}_2$.

Multiplication: $\mathbf{C}_{\text{mult}} \stackrel{\Delta}{=} \mathbf{C}_1 \times \mathbf{G}_{\overline{n}}^{-1}(\mathbf{C}_2)$.

This also allows us to compute a homomorphic NAND gate by outputting $\mathbf{G}_{\overline{n}} - \mathbf{C}_1 \times \mathbf{G}_{\overline{n}}^{-1}(\mathbf{C}_2)$.

The correctness of decryption and homomorphic evaluation is obvious due to the correctness of the conventional GSW-FHE, and the semantic security is based on the MLWE assumption, which is in turn based on the DLWE assumption.

2.4. Compressible Single-Key FHE. Very recently, Gentry and Halevi proposed the first compressible single-key FHE. At a high level, their approach is based on two ideas:

Using matrix versions of LWE and GSW-FHE [4]. Firstly, they noticed that the conventional GSW-FHE uses only one "slot" when decrypting, i.e.,

$$[1, -\mathbf{s}]\mathbf{C} = \mu[1, -\mathbf{s}]\mathbf{G} + [1, -\mathbf{s}]\begin{bmatrix}\mathbf{b}\\ \mathbf{A}\end{bmatrix}\mathbf{R} \approx [\mu, -\mu\mathbf{s}]\mathbf{G}. \quad (4)$$

Their solution to utilize more "slots" is to use LWE with matrix secrets, which yields the matrix version of GSW-FHE described above. They added a compression algorithm after evaluation:

$$\mathbf{C}^* \triangleq \sum_i \sum_j \sum_k \mathbf{C}_{i,j,k} \cdot \mathbf{G}^{-1}\left(2^k \cdot \begin{bmatrix}\mathbf{E}_n^{i,j}\\ 0\end{bmatrix}\mathbf{G}\right) \in \mathbf{Z}_q^{\overline{n}\times\overline{n}\ell}, \quad (5)$$

where $\mathbf{C}^*$ is the compressed ciphertext and $\mathbf{E}_n^{i,j}$ is an $n \times n$ matrix that has 1 in the i^{th} row and j^{th} column and 0's in all other entries. Then the matrix secrets are used in the decryption algorithm:

$$\begin{aligned}[\mathbf{I}_n\| - \mathbf{S}]C^* &\approx \sum_i \sum_j \sum_k 2^k \mu_{i,j,k}[\mathbf{I}_n\|\mathbf{S}]\begin{bmatrix}\mathbf{E}_n^{i,j}\\ 0\end{bmatrix}\mathbf{G}\\ &= \sum_i \sum_j \sum_k \left(2^k \mu_{i,j,k}\mathbf{I}_n\mathbf{E}_n^{i,j}\right)\mathbf{G}\\ &\triangleq \mathbf{MG},\end{aligned} \quad (6)$$

where $\mathbf{M} \in \mathbf{Z}_q^{n\times n}$. The compression rate is now bounded by $\alpha = ((n \times n)/(\overline{n} \times \overline{n}\ell)) = ((n \times n)/((n + \text{small}) \times (n + \text{small})\ell)) \approx (1/\ell) = (1/\lfloor \log q \rfloor)$.

Using a "nearly square" gadget matrix. The second idea is to transform a "fat" compressed ciphertext matrix into a "nearly square" compressed ciphertext matrix using a "nearly square" gadget matrix $\mathbf{H}$ with some special properties, which comes with almost no dimension expansion. On the other hand, attributed to $\mathbf{H}$, the redundancy in ciphertexts enables us to erase the noise and finally recover the message matrix. We omit the details of constructing the "nearly square" gadget matrix $\mathbf{H}$ because it is not strictly required in this paper. The essential characteristic of $\mathbf{H}$ that is required in our construction is that it has a "public trapdoor" matrix $\mathbf{H}^{-1}(0) = \mathbf{F}$ satisfying the following:

(1) $\mathbf{F}$ has small entries ($\ll q$);
(2) $\mathbf{H} \times \mathbf{F} = 0 \pmod q$, i.e., all rows of $\mathbf{H}$ span the kernel space of $\mathbf{F}$ modulo q;
(3) $\mathbf{F}$ is full rank over R;
(4) $\mathbf{F}$ is efficiently computable.

Then, the compression algorithms can produce an optimal rate compressed ciphertext. In particular, let $\{\mathbf{C}_{i,j,k}\}_{i,j\in[n],\ k\in[\ell]}$ be GSW ciphertexts encrypting $\{\mu_{i,j,k} \in \{0,1\}\}_{i,j\in[n],k\in[\ell]}$. Set the compressed ciphertext

$$\mathbf{C}^* \triangleq \sum_i \sum_j \sum_k \mathbf{C}_{i,j,k} \times \mathbf{G}^{-1}\left(2^k \cdot \begin{bmatrix}\mathbf{E}_n^{i,j}\\ 0\end{bmatrix}\mathbf{H}\right) \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}, \quad (7)$$

where $\mathbf{E}_n^{i,j}$ is an $n \times n$ matrix that has 1 in the i^{th} row and j^{th} column and 0's in all other entries.

Roughly speaking, they have now achieved an optimal compression rate because after "routine decryption" we get

$$\begin{aligned}[\mathbf{I}_n\|\mathbf{S}]\mathbf{C}^* &\approx \sum_i \sum_j \sum_k \left(2^k \mu^{i,j,k}[\mathbf{I}_n\|\mathbf{S}]\begin{bmatrix}\mathbf{E}_n^{i,j}\\ 0\end{bmatrix}\mathbf{H}\right)\\ &= \sum_i \sum_j \sum_k \left(2^k \mu^{i,j,k}\mathbf{I}_n\mathbf{E}_n^{i,j}\right)\mathbf{H}\\ &\triangleq \mathbf{MH},\end{aligned} \quad (8)$$

where $\mathbf{M} \in \mathbf{Z}_q^{n\times n}$. The compression rate is now $\alpha = ((n \times n)/(\overline{n} \times \overline{n})) = ((n \times n)/((n + \text{small}) \times (n + \text{small}))) \approx 1$, an optimal compression rate!

In particular, they upgrade the "routine decryption" into "compressed decryption" which is composed of four steps:

(1) $\mathbf{Z} \triangleq \overline{S} \times C^* = \mathbf{MH} + \mathbf{E} \pmod q$, where $\mathbf{M} \in \mathbb{Z}_q^{n\times n}$ is a message matrix; $\mathbf{H} \in \mathbb{Z}_q^{n\times\overline{n}}$ is the nearly square gadget matrix; $\mathbf{E} \in \mathbb{Z}_q^{n\times\overline{n}}$ is a small error matrix;
(2) $\mathbf{Y} \triangleq \mathbf{Z} \times \mathbf{F} = \mathbf{EF} \pmod q$, where $\mathbf{F} \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}$ is the public trapdoor matrix;
(3) $\mathbf{X} \triangleq \mathbf{Y} \times \mathbf{F}^{-1} = \mathbf{E}$ (note that $\mathbf{F}$ is full rank over R and it is required that every entry of $\mathbf{EF}$ does not wrap around $\mathbb{Z}_q$);
(4) $M' \triangleq (\mathbf{Z} - \mathbf{X}) \times \mathbf{H}^{-1} = \mathbf{M} \pmod q$ (note that $\mathbf{H}$ is a row full-rank matrix modulo q).

Recall that the compression rate is the ratio of plaintext size to the ciphertext size. To achieve an optimal compression rate of $1 - \varepsilon$ for some small $\varepsilon > 0$, set $t \triangleq (\varepsilon/2)$, where t is the compression parameter defined as a function of the desired compression rate. Then the compression rate $\gamma = (n \cdot n/\overline{n} \cdot \overline{n}) = ((t-1)/t)^2 \geq 1 - \varepsilon$.

The additional compression algorithm does not compromise security and the key-generation, encryption, and evaluation algorithms are basically the same as the ones of matrix version of GSW-FHE. Thus, the semantic security of their compressible FHE is based solely on the DLWE assumption. However, the compressed ciphertexts are not GSW ciphertexts any longer, which means they can only support homomorphic addition and multiplication on the left by some GSW ciphertexts encrypting a small scalar. For more information, please refer to [23].

3. Compressible Multikey Fully Homomorphic Encryption

In this section, utilizing the idea of our new compressible expansion algorithm described in Section 1.1, we propose the first compressible MKFHE scheme and analyze the correctness and security level of the construction.

3.1. Compressible Multikey FHE. We begin with the definition of compressible MKFHE, which is adapted from the definitions of compressible FHE in [23] and MKFHE in [7].

Definition 3. A compressible multikey fully homomorphic encryption scheme consists of seven PPT algorithms (Setup, KeyGenerate, Encrypt, Expand, Evaluate, Compress and Decrypt) defined as follows:

Setup $(1^\lambda, 1^N)$: Take a security parameter λ and a bound N on the number of users involved as inputs and output parameters params. params is taken in as an input in all of the following algorithms; thus, we just omit it.

KeyGen (params, $i \in [N]$): Output a secret key SK_i and a public key PK_i for the i^{th} user.

Enc $(\mathrm{PK}_i, \mu \in \{0,1\})$: Take a public key PK_i and a bit $\mu \in \{0,1\}$ as inputs, and output a fresh low-rate ciphertext c_i .

$\mathrm{Exp}\left(\left\{\mathrm{PK}_j\right\}_{j\in[N]}, c_i\right)$: Take public keys $\left\{\mathrm{PK}_j\right\}_{j\in[N]}$ and a fresh ciphertext c_i under the i^{th} public key as inputs; and compute and output an expanded low-rate ciphertext $\widehat{c}$ under N's public keys $(\mathrm{PK}_1, \mathrm{PK}_2, \ldots, \mathrm{PK}_N)$.

Eval $(f, \widehat{c} = (\widehat{c}_i)_{i\in[t]})$: Take a circuit f and a vector of expanded ciphertexts $\widehat{c} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_t)$ as inputs, corresponding to each input bit of f, and output another vector of evaluated low-rate ciphertext $\widehat{c}' = (\widehat{c}_1', \widehat{c}_2', \ldots, \widehat{c}_t')$, corresponding to each output bit of f.

Comp $(\widehat{c} = (\widehat{c}_i)_{i\in[t]})$: Take a vector of expanded or evaluated low-rate ciphertexts $\widehat{c} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_t)$ as inputs, and output one or more compressed high-rate ciphertext(s) $c^* = \left\{c_i^*\right\}_{i\in[s]}$.

$\mathrm{Dec}\left(\left\{\mathrm{SK}_i\right\}_{i\in[N]}, c^*\right)$: Take N secret keys $(\mathrm{SK}_1, \mathrm{SK}_2, \ldots, \mathrm{SK}_N)$, corresponding to identities $(\mathrm{id}_1, \mathrm{id}_2, \ldots, \mathrm{id}_N)$, and the compressed ciphertext(s) c^* as inputs, and output a vector of message bits $\mathbf{m} = \{\mu_i\}_{i\in[t]}$ corresponding to $\{\widehat{c}_i\}_{i\in[t]}$.

The scheme is said to be correct if, for every allowed circuit f and a vector of message bits $\mathbf{m} = (\mu_1, \ldots, \mu_t) \in \{0,1\}^t$ corresponding to each input bit of f, it holds that

$$\mathrm{Dec}\left(\left\{\mathrm{SK}_i\right\}_{i\in[N]}, \mathrm{Comp}\left(\mathrm{Eval}\left(f, \widehat{c} = (\widehat{c}_i)_{i\in[t]}\right)\right)\right) = \mathbf{m}, \quad (9)$$

where each $\widehat{c}_i = \mathrm{Exp}\left(\left\{\mathrm{PK}_j\right\}_{j\in[N]}, \mathrm{Enc}\left(\mathrm{PK}_k, \mu_i \in \{0,1\}\right)\right)$.

Using the parameters described above, the scheme is said to have a compression rate of $1-\varepsilon$ for any $\varepsilon > 0$ if, for every allowed circuit f that has sufficiently long outputs, the compression rate

$$\gamma = \frac{|\mathbf{m}|}{\left|\mathrm{Comp}\left(\mathrm{Eval}\left(f, \widehat{c} = (\widehat{c}_i)_{i\in[t]}\right)\right)\right|} \geq 1 - \varepsilon. \quad (10)$$

The semantic security definition for compressible multikey fully homomorphic encryption is the same as that for multikey fully homomorphic encryption, which is the same as that for single-key fully homomorphic encryption because all of parameters used in the expansion and compression algorithms are public and thus do not compromise security. In particular, given a security parameter λ and a bound N on the number of users involved, the following distributions are computationally indistinguishable:

$$\left(\mathrm{params}, \mathrm{PK}_i, \mathrm{Enc}\left(\mathrm{PK}_i, 0\right)\right) \overset{\mathrm{comp}}{\approx} \left(\mathrm{params}, \mathrm{PK}_i, \mathrm{Enc}\left(\mathrm{PK}_i, 1\right)\right), \quad (11)$$

where $\mathrm{params} \leftarrow \mathrm{Setup}\left(1^\lambda, 1^N\right)$ and $(\mathrm{SK}_i, \mathrm{PK}_i) \leftarrow \mathrm{KeyGen}\left(\mathrm{params}, i \in [N]\right)$. We do not consider the distributed decryption for our MKFHE scheme, which can be used to construct round-efficient secure multiparty computation protocols, and instead put the main focus on the compressible expansion algorithm.

3.2. Our Construction

C-MKFHE.Setup $(1^\lambda, 1^N)$: Choose a lattice dimension parameter $r = \mathrm{poly}(\lambda, N)$, a sufficiently large modulus q, a β_χ-bounded error distribution $\chi = \chi(\lambda, N)$, and a compression parameter $t \overset{\Delta}{=} (t'-1)N$. The parameter t' will be specified in Subsection 3.3.2. All of the rows of the nearly square gadget matrix $\mathrm{H}' \in \mathbb{Z}_q^{(t'-1)\times t'}$ span the kernel space of its public trapdoor matrix $\mathrm{F}' \in \mathbb{Z}_q^{t'\times t'}$ having the properties stated in 2.4. Let $\ell \overset{\Delta}{=} \lceil \log q \rceil$, $n \overset{\Delta}{=} tr$, $\overline{n} \overset{\Delta}{=} (t+N)r$, $m \overset{\Delta}{=} (t+1)r\ell$, and $\overline{m} \overset{\Delta}{=} (t+N)r\ell$. Then, set $\mathbf{H} \overset{\Delta}{=} \mathrm{H}' \otimes \mathbf{I}_{rN} \in \mathbb{Z}_q^{n\times\overline{n}}$ and $\mathbf{F} \overset{\Delta}{=} \mathrm{F}' \otimes \mathbf{I}_{rN} \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}$. Choose a random matrix $\mathbf{B} \overset{\$}{\leftarrow} \mathbb{Z}_q^{r\times m}$. Output params $= (r, N, q, \ell, \beta_\chi, \chi, t', t, n, \overline{n}, m, \overline{m}, \mathbf{H}, \mathrm{H}', \mathbf{F}, \mathrm{F}', \mathbf{B})$.

C-MKFHE.KeyGen(params): Choose random matrices $\mathbf{S} \overset{\$}{\leftarrow} \mathbb{Z}_q^{tr\times r}$ and $\mathbf{E} \leftarrow \chi^{tr\times m}$. Compute $\mathbf{A} = \mathbf{SB} + \mathbf{E}$. Let the secret key and public key be $\mathrm{SK} \overset{\Delta}{=} \overline{\mathrm{S}} \overset{\Delta}{=} [\mathbf{I}_{\mathrm{tr}}\mathbf{S}] \in \mathbb{Z}_q^{tr\times(t+1)r}$ and $\mathrm{PK} \overset{\Delta}{=} \overline{A} \overset{\Delta}{=} \begin{bmatrix} -\mathbf{A} \\ \mathbf{B} \end{bmatrix} \in \mathbb{Z}_q^{(t+1)r\times m}$. Note that $\overline{SA} = \mathbf{E}$. Output (PK, SK).

C-MKFHE.Enc(params, $\mathrm{PK}_i, \mu \in \{0,1\}$): Given a message bit μ, randomly select a matrix $\overline{R}_i \overset{\$}{\leftarrow} \{0,1\}^{m\times m}$. Set

$$\overline{\mathrm{C}}_i \overset{\Delta}{=} \mu\mathbf{G}_{(t+1)r} + \overline{\mathrm{A}}_i\overline{\mathrm{R}}_i \in \mathbb{Z}_q^{(t+1)r\times m}. \quad (12)$$

Then, partition $\overline{\mathrm{C}}_i$ as follows:

$$\overline{\mathrm{C}}_i \overset{\Delta}{=} \begin{bmatrix} \overline{\mathrm{C}}_i^{(1,1)} & \overline{\mathrm{C}}_i^{(1,2)} \\ \overline{\mathrm{C}}_i^{(2,1)} & \overline{\mathrm{C}}_i^{(2,2)} \end{bmatrix}, \quad (13)$$

where $\overline{\mathrm{C}}_i^{(1,1)} \in \mathbb{Z}_q^{tr\times tr\ell}, \overline{\mathrm{C}}_i^{(1,2)} \in \mathbb{Z}_q^{tr\times r\ell}, \overline{\mathrm{C}}_i^{(2,1)} \in \mathbb{Z}_q^{r\times tr\ell}, \overline{\mathrm{C}}_i^{(2,2)} \in \mathbb{Z}_q^{r\times r\ell}$. Let $\mathbf{R}_i \in \{0,1\}^{m\times r\ell}$ be the last $r\ell$ columns of $\overline{R}_i$. Note that

$$\overline{\mathrm{C}}_i^{(2,2)} = \mu\mathbf{G}_r + \mathbf{BR}_i \in \mathbb{Z}_q^{r\times r\ell}. \quad (14)$$

For $x \in [t], y \in [m], b \in [rl]$, compute

$$\begin{aligned} \mathbf{V}_{i,1}^{(x,y,b)} &= -\mathbf{A}_i\widetilde{R}_i^{(x,y,b)} + \mathbf{R}_i[y,b]\mathbf{G}_m^{(x)} \in \mathbb{Z}_q^{tr\times r\ell}; \\ \mathbf{V}_{i,2}^{(x,y,b)} &= \mathbf{B}\widetilde{R}_i^{(x,y,b)} \in \mathbb{Z}_q^{r\times r\ell}, \end{aligned} \quad (15)$$

where $\widetilde{R}_i^{(x,y,b)} \xleftarrow{\$} \{0,1\}^{m\times r\ell}$ and $\mathbf{G}_r^{(x)} = \begin{bmatrix} 0^{(x-1)r\times r\ell} \\ \mathbf{G}_r \\ 0^{(t-x)r\times r\ell} \end{bmatrix} \in \mathbb{Z}^{tr\times r\ell}$.

Set

$$\mathbf{U}_i \triangleq \left(\mathbf{V}_{i,1}^{(x,y,b)}, \mathbf{V}_{i,2}^{(x,y,b)}\right)_{x\in[t],y\in[m],b\in[rl]}. \tag{16}$$

Output a tuple of encryption $\mathrm{CIPHER}_i \triangleq (\overline{\mathrm{C}}_i, \mathbf{U}_i)$.

C-MKFHE.Exp(params, $\{\mathrm{PK}_i\}_{i\in[N]}$, CIPHER_i): For every $j \in [N]\backslash\{i\}$, $x \in [t]$, $y \in [n], b \in [ml]$, compute

$$\mathbf{Z}_{i,j}^{(x,y,b)}[z_1, z_2] \triangleq \begin{cases} -\mathbf{A}_j[z_1 + xr, y], & \text{if } z_2 = b, \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

that is, $\mathbf{Z}_{i,j}^{(x,y,b)} \triangleq \sum_{z_1=1}^{r} -\mathbf{A}_j[z_1 + xr, y] \cdot \mathbf{E}'_{z_1,b} \in \mathbb{Z}_q^{r\times r\ell}$, where $\mathbf{E}'_{z_1,b}$ is an $r \times r\ell$ matrix that has 1 in the z_1^{th} row and the b^{th} column and 0's in all other entries. Set

$$\overline{Z}_{i,j}^{(x,y,b)} \triangleq \begin{bmatrix} 0^{(x-1)r\times r\ell} \\ \mathbf{Z}_{i,j}^{(x,y,b)} \\ 0^{(t-x)r\times r\ell} \end{bmatrix} \in \mathbb{Z}_q^{tr\times r\ell}. \tag{18}$$

Compute

$$\begin{aligned} \mathbf{X}_{i,j}^{(1)} &\triangleq \sum_{x=1}^{t}\sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{V}_{i,1}^{(x,y,b)} \cdot \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) \in \mathbb{Z}_q^{tr\times r\ell}; \\ \mathbf{X}_{i,j}^{(2)} &\triangleq \sum_{x=1}^{t}\sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{V}_{i,2}^{(x,y,b)} \cdot \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) \in \mathbb{Z}_q^{r\times r\ell}. \end{aligned} \tag{19}$$

Finally, expand a fresh ciphertext to an expanded one.

$$\widehat{\mathrm{C}}_i \triangleq \begin{bmatrix} \overline{\mathrm{C}}_i^{(1,1)} & \mathbf{X}_{i,1}^{(1)} & \cdots & \overline{\mathrm{C}}_i^{(1,2)} & \cdots & \mathbf{X}_{i,N}^{(1)} \\ & \overline{\mathrm{C}}_i^{(2,2)} & & & & \\ & & \ddots & & & \\ \overline{\mathrm{C}}_i^{(2,1)} & \mathbf{X}_{i,1}^{(2)} & \cdots & \overline{\mathrm{C}}_i^{(2,2)} & \cdots & \mathbf{X}_{i,N}^{(2)} \\ & & & & \ddots & \\ & & & & & \overline{\mathrm{C}}_i^{(2,2)} \end{bmatrix} \in \mathbb{Z}_q^{\overline{n}\times\overline{m}}. \tag{20}$$

Output $\widehat{\mathrm{C}}_i$.

C-MKFHE.Comp$\left(\text{params}, \left\{\widehat{\mathrm{C}}_{u,v,\omega}\right\}_{u,v\in[n],\omega\in[\ell]}\right)$: In this algorithm, a series of low-rate expanded (or evaluated) ciphertexts are compressed into a high-rate ciphertext. Let $\mathbf{T}_{u,v} \triangleq \begin{bmatrix} \mathbf{E}'_{u,v} \\ 0^{Nr\times n} \end{bmatrix} \in \mathbb{Z}^{\overline{n}\times n}$, where $\mathbf{E}'_{u,v}$ is an $n \times n$ matrix that has 1 in the u^{th} row and the v^{th} column and 0's in all other entries. Set the compressed ciphertext as

$$\mathbf{C}^* \triangleq \sum_{u\in[n]}\sum_{v\in[n]}\sum_{\omega\in[\ell]} \widehat{\mathrm{C}}_{u,v,\omega} \cdot \mathbf{G}_{\overline{n}}^{-1}\left(2^{\omega}\mathbf{T}_{u,v} \times \mathbf{H}\right) \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}. \tag{21}$$

Note that the nearly square gadget matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times\overline{n}}$ and its public trapdoor $\mathbf{F} \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}$ play a significant role in the compression and compressed decryption algorithms.

C-MKFHE.CompDec(params, $\mathbf{C}^*$, $(\mathrm{SK}_i)_{i\in[N]}$): Given a compressed ciphertext $\mathbf{C}^*$, set the combined key $\widehat{\mathbf{S}} \triangleq [\mathbf{I}_{\mathrm{tr}}, \mathbf{S}_1, \ldots, \mathbf{S}_N] \in \mathbb{Z}^{n\times\overline{n}}$. The compressed decryption is composed of four steps:

(1) $\mathbf{Z} \triangleq \mathbf{S} \times \mathbf{C}^* \pmod q$;
(2) $\mathbf{Y} \triangleq \mathbf{Z} \times \mathbf{F} \pmod q$, where $\mathbf{F} \in \mathbb{Z}_q^{\overline{n}\times\overline{n}}$ is the public trapdoor matrix;
(3) $\mathbf{X} \triangleq \mathbf{Y} \times \mathbf{F}^{-1}$ (note that $\mathbf{F}$ is full rank over R);
(4) $M' \triangleq (\mathbf{Z} - \mathbf{X}) \times \mathbf{H}^{-1} \pmod q$ (note that $\mathbf{H}$ is a row full-rank matrix modulo q, so that there exists a matrix $\mathbf{H}^{-1} \in \mathbb{Z}_q^{\overline{n}\times n}$ such that $\mathbf{H} \times \mathbf{H}^{-1} = \mathbf{I}_n$).

C-MKFHE.NAND(params, $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2$): Given two expanded ciphertext matrices $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2$ for two plaintexts μ_1, μ_2, homomorphic NAND operation is defined as

$$\widehat{\mathrm{C}}_{\mathrm{NAND}} \triangleq \mathbf{G}_{\overline{n}} - \widehat{\mathrm{C}}_1 \cdot \mathbf{G}_{\overline{n}}^{-1}\left(\widehat{\mathrm{C}}_2\right). \tag{22}$$

Output $\widehat{\mathrm{C}}_{\mathrm{NAND}}$. Observe that $\mathbf{G}_x^{-1}(\cdot)$ is randomized, and so is this algorithm.

C-MKFHE.Eval(params, $f, \widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2, \ldots, \widehat{\mathrm{C}}_s$): An NAND-circuit $f\colon \{0,1\}^t \longrightarrow \{0,1\}$ is applied to a set of ciphertexts $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2, \ldots, \widehat{\mathrm{C}}_s$, which leads to a ciphertext $\widehat{\mathrm{C}}_f$.

3.3. Setting the Parameters. In this subsection, the correctness, homomorphic properties, security, and compression rate of the proposed compressible MKFHE are analyzed. If we use a somewhat (or leveled) version without bootstrapping, then the error bound is dependent on the maximal multiplication depth of allowed evaluated functions. In this paper, we would like to give an asymptotic analysis of various parameters of a fully homomorphic one with bootstrapping so that all ciphertexts before being packed have an error bound similar to that of freshly expanded ciphertexts.

3.3.1. Correctness and Homomorphism. The correctness of compressed decryption may be less obvious, since the encryption and expansion algorithms are quite different from previous ones. During encryption, it holds that

$$\overline{S}_i \times \overline{\mathrm{C}}_i = \mu\overline{S}_i\mathbf{G}_{(t+1)r} + \mathbf{E}_i\overline{R}_i \in \mathbb{Z}_q^{n\times m}, \tag{23}$$

because $\overline{S}_i\overline{A}_i = \mathbf{E}_i$.

Besides, for the additional helper information matrices, we have

$$\begin{aligned} \mathbf{S}_j \times \overline{\mathrm{C}}_i^{(2,2)} &= \mu\mathbf{S}_j\mathbf{G}_r + \mathbf{A}_j\mathbf{R}_i - \mathbf{E}_j\mathbf{R}_i \in \mathbb{Z}_q^{n\times r\ell}; \\ \mathbf{I}_{tr} \times \mathbf{V}_{i,1}^{(x,y,b)} &= -\mathbf{A}_i\widetilde{R}_i^{(x,y,b)} + \mathbf{R}_i[y,b]\mathbf{G}_r^{(x)} \in \mathbb{Z}_q^{n\times r\ell}; \\ \mathbf{S}_i \times \mathbf{V}_{i,2}^{(x,y,b)} &= (\mathbf{A}_i - \mathbf{E}_i)\widetilde{R}_i^{(x,y,b)} \in \mathbb{Z}_q^{n\times r\ell}, \end{aligned} \tag{24}$$

because $\mathbf{A}_i - \mathbf{E}_i = \mathbf{S}_i\mathbf{B}$.

During the expansion algorithm, those weird matrices $\mathbf{X}_{i,j}^{(1)}$ and $\mathbf{X}_{i,j}^{(2)}$ are customized to eliminate the undesired $\mathbf{A}_j\mathbf{R}_i$ in $\mathbf{S}_j\overline{\mathrm{C}}_i^{(2,2)}$. In particular, we have

$$\begin{aligned}\mathbf{I}_{\mathrm{tr}} \times \mathbf{X}_{i,j}^{(1)} &= \sum_{x=1}^{t}\sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{V}_{i,1}^{(x,y,b)} \cdot \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) \\ &= \sum_{x=1}^{t}\sum_{y=1}^{m}\sum_{b=1}^{rl}\left(\mathbf{A}_i \widetilde{R}_i^{(x,y,b)} \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) + \underline{\mathbf{R}_i[y,b]\overline{Z}_{i,j}^{(x,y,b)}}\right) \\ &= \sum_{x=1}^{t}\sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{A}_i \widetilde{R}_i^{(x,y,b)} \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) - \underline{\mathbf{A}_j \mathbf{R}_i},\end{aligned} \tag{25}$$

$$\begin{aligned}\mathbf{S}_i \times \mathbf{X}_{i,j}^{(2)} &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml} \mathbf{S}_i \mathbf{V}_{i,2}^{(x,y,b)} \cdot \mathbf{G}_r^{-1}\left(\mathbf{Z}_i^{(x,y,b)}\right) \\ &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml} (\mathbf{A}_i - \mathbf{E}_i)\widetilde{R}_i^{(x,y,b)} \cdot \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right).\end{aligned} \tag{26}$$

The underlined part can be a tricky step. We now analyze it as follows:

$$\begin{aligned}&\sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{R}_i[y,b] \sum_{x=1}^{t}\left(\overline{Z}_{i,j}^{(x,y,b)}\right) \\ &= \sum_{y=1}^{m}\sum_{b=1}^{rl} \mathbf{R}_i[y,b] \sum_{a=1}^{tr}\left(-\mathbf{A}_j[a,y] \times \mathbf{E}'_{a,b}\right) \\ &= \sum_{a=1}^{tr}\sum_{y=1}^{m}\sum_{b=1}^{rl}\left(-\mathbf{A}_j[a,y] \cdot \mathbf{R}_i[y,b] \times \mathbf{E}'_{a,b}\right) \\ &= -\mathbf{A}_j \mathbf{R}_i,\end{aligned} \tag{27}$$

where $\mathbf{E}_{a,b}{}'$ is an $tr \times r\ell$ matrix that has 1 in the a^{th} row and the b^{th} column and 0's in all other entries.

Thus, for any expanded ciphertext, we have

$$\begin{aligned}\widehat{S} \times \widehat{\mathrm{C}}_i = \Big[&\overline{\mathrm{C}}_i^{(1,1)} + \mathbf{S}_i \overline{\mathrm{C}}_i^{(2,1)} \Big\| \mathbf{X}_{i,1}^{(1)} + \mathbf{S}_1 \overline{\mathrm{C}}_i^{(2,2)} + \mathbf{S}_i \mathbf{X}_{i,j}^{(2)} \Big\| \cdots \\ &\cdot \Big\| \overline{\mathrm{C}}_i^{(1,2)} + \mathbf{S}_i \overline{\mathrm{C}}_i^{(2,2)} \Big\| \cdots \mathbf{X}_{i,N}^{(1)} + \mathbf{S}_N \overline{\mathrm{C}}_i^{(2,2)} + \mathbf{S}_i \mathbf{X}_{i,N}^{(2)} \Big] \\ &\in \mathbb{Z}^{n \times \overline{m}}.\end{aligned} \tag{28}$$

There are only two types of all blocks in the above seemingly cumbersome matrix. The first comes from equation (23) by partitioning

$$\begin{aligned}&[\mathbf{I}_{tr} \| \mathbf{S}_i] \times \begin{bmatrix} \overline{\mathrm{C}}_i^{(1,1)} & \overline{\mathrm{C}}_i^{(1,2)} \\ \overline{\mathrm{C}}_i^{(2,1)} & \overline{\mathrm{C}}_i^{(2,2)} \end{bmatrix} \\ &= \mu[\mathbf{I}_{tr}\mathbf{G}_{tr} \| \mathbf{S}_i \mathbf{G}_r] + \mathbf{E}_i \overline{R}_i \\ &\triangleq [\mu \mathbf{I}_{tr}\mathbf{G}_{tr} + \widetilde{E}_0 \mu \mathbf{S}_i \mathbf{G}_r + \widetilde{E}_i] \in \mathbb{Z}_q^{n \times m}.\end{aligned} \tag{29}$$

The second comes from equations (24)–(26):

$$\begin{aligned}&\mathbf{S}_j \overline{\mathrm{C}}_i^{(2,2)} + \mathbf{X}_{i,j}^{(1)} + \mathbf{S}_i \mathbf{X}_{i,j}^{(2)} \\ &= \mu \mathbf{S}_j \mathbf{G}_r - \mathbf{E}_j \mathbf{R}_i - \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml} \mathbf{E}_i \widetilde{R}_i^{(x,y,b)} \mathbf{G}_r^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b)}\right) \\ &\triangleq \mu \mathbf{S}_j \mathbf{G}_r + \widetilde{E}_j \in \mathbb{Z}_q^{n \times r\ell}.\end{aligned} \tag{30}$$

Finally, we get the most important decryption invariant in all GSW-like encryption from equations (28)–(30):

$$\begin{aligned}\widehat{S} \times \widehat{\mathrm{C}}_i &= \left[\mu \mathbf{I}_{tr}\mathbf{G}_{tr} + \widetilde{E}_0 \Big\| \mu \mathbf{S}_1 \mathbf{G}_r + \widetilde{E}_1 \Big\| \cdots \Big\| \mu \mathbf{S}_N \mathbf{G}_r + \widetilde{E}_N\right] \\ &\triangleq \mu \widehat{S} \mathbf{G}_{\overline{n}} + \widehat{E} \in \mathbb{Z}_q^{n \times \overline{m}}.\end{aligned} \tag{31}$$

In terms of the error bound, we have $\beta_{\mathrm{Exp}} \triangleq m^4 \beta_\chi$; that is, $\max_{a,b}|\widehat{E}[a,b]| < \beta_{\mathrm{Exp}}$ with overwhelming probability.

With equation (31) in mind, everything turns familiar again. Homomorphic properties are obvious, and one can find them in most GSW-like FHE (or MKFHE) schemes. Thus, we just omit them here.

On the other hand, the correctness of compressed decryption also turns obvious. According to equation (31), we have

$$\begin{aligned}\widehat{S} \times \mathbf{C}^* &= \sum_{u \in [n]}\sum_{v \in [n]}\sum_{\omega \in [\ell]} \widehat{S}\widehat{\mathrm{C}}_{u,v,\omega} \times \mathbf{G}_{\overline{n}}^{-1}\left(2^\omega \cdot \mathbf{T}_{u,v}\mathbf{H}\right) \\ &= \sum_{u,v,\omega}\left(\mu_{u,v,\omega}\widehat{S}\mathbf{G}_{\overline{n}} + \widehat{E}\right)\mathbf{G}_{\overline{n}}^{-1}\left(2^\omega \cdot \mathbf{T}_{u,v}\mathbf{H}\right) \\ &\triangleq \sum_{u,v,\omega}\left(2^\omega \mu_{u,v,\omega}\widehat{S}\mathbf{T}_{u,v}\mathbf{H}\right) + \mathbf{E}^* \\ &= \sum_{u,v,\omega}\left(2^\omega \mu_{u,v,\omega}\mathbf{E}_{u,v}{}'\right)\mathbf{H} + \mathbf{E}^* \\ &\triangleq \mathbf{M}\mathbf{H} + \mathbf{E}^* \in \mathbb{Z}_q^{n \times \overline{n}},\end{aligned} \tag{32}$$

where $\mathbf{E}'_{u,v}$ is an $n \times n$ matrix that has 1 in the u^{th} row and the v^{th} column and 0's in all other entries and $\mathbf{M} \in \mathbb{Z}_q^{n \times n}$ is a compressed plaintext matrix that can be obtained from $\mathbf{Z} \triangleq \widehat{S} \times \mathbf{C}^* = \mathbf{M}\mathbf{H} + \mathbf{E}^*$, by using Gentry and Halevi's idea of the nearly square gadget matrix, which was briefly described in 2.4. That is,

(1) $\mathbf{Y} \triangleq \mathbf{Z} \times \mathbf{F} = \mathbf{E}^*\mathbf{F} \pmod q$, where $\mathbf{F} \in \mathbb{Z}_q^{\overline{n} \times \overline{n}}$ is the public trapdoor matrix;

(2) $\mathbf{X} \triangleq \mathbf{Y} \times \mathbf{F}^{-1} = \mathbf{E}^*$ (note that $\mathbf{F}$ is full rank over R);

(3) $M' \triangleq (\mathbf{Z} - \mathbf{X}) \times \mathbf{H}^{-1} = \mathbf{M} \pmod q$ (note that $\mathbf{H}$ is a row full-rank matrix modulo q, so that there exists a matrix $\mathbf{H}^{-1} \in \mathbb{Z}_q^{\overline{n} \times n}$ such that $\mathbf{H} \times \mathbf{H}^{-1} = \mathbf{I}_n$).

The matrix M' is the desired one if every entry of $\mathbf{E}^*\mathbf{F}$ does not wrap around $\mathbb{Z}_q$; that is, $(\mathbf{E}^*\mathbf{F} \bmod q) = \mathbf{E}^*\mathbf{F}$ over the integers. As for the error bound, we have $\beta_{\mathrm{Comp}} \triangleq \overline{m} m^2 \beta_{\mathrm{Exp}}$; that is, $\max_{a,b}|\mathbf{E}^*[a,b]| < \beta_{\mathrm{Comp}}$ with overwhelming probability. On the other hand, by the construction of the public trapdoor matrix $\mathbf{F}$, it is required that $\beta_{\mathrm{Comp}} \le \lfloor q^{1/t'} \rfloor$.

3.3.2. Security and Compression Rate. It remains to show the security level and compression rate of our construction.

The additional expansion and compression algorithms are public and do not compromise security. The key-generation algorithm is basically the same as that of the underlying GSW-FHE scheme. In addition, during the encryption procedure, a fresh ciphertext for the i^{th} user CIPHER_i consists of two parts $\overline{\text{C}}_i$ and $(\mathbf{V}_{i,1}^{(x,y,b)}, \mathbf{V}_{i,2}^{(x,y,b)})$, of which randomness matrices are mutually independent, all in a similar encryption form. Exploiting a standard hybrid analysis, one can easily verify that the semantic security of our compressible MKFHE is based solely on the DLWE assumption.

The view of the attacker is the following distribution:

$$\left(\text{params}, \text{PK}_i, \overline{\text{C}}_i, \mathbf{U}_i = \left(\mathbf{V}_{i,1}^{(x,y,b)}, \mathbf{V}_{i,2}^{(x,y,b)}\right)_{x\in[t],y\in[m],b\in[rl]}\right), \tag{33}$$

generated via $\text{params} \leftarrow C - \text{MKFHE} \cdot \text{Setup}(1^\lambda, 1^N)$, $(\text{PK}_i, \text{SK}_i) \leftarrow C - \text{MKFHE} \cdot \text{KeyGen}(\text{params})$, and $(\overline{\text{C}}_i, \mathbf{U}_i) \leftarrow C - \text{MKFHE} \cdot \text{Enc}(\text{params}, \text{PK}_i, \mu)$, where $\mu \in \{0, 1\}$.

We prove the semantic security of our construction by relying on the semantic security of the underlying matrix version of GSW-FHE scheme. The proof consists of the following hybrids:

(1) Firstly, we alter each of the ciphertexts $(\mathbf{V}_{i,1}^{(x,y,b)}, \mathbf{V}_{i,2}^{(x,y,b)})$, where $x \in [t]$, $y \in [m]$, $b \in [rl]$ so that, instead of being GSW ciphertexts of $\mathbf{R}_i[y, b]$, we change them to GSW ciphertexts of 0. It follows from the semantic security of GSW encryption which is based solely on the DLWE assumption.

(2) After the first step, no information about the encryption randomness $\mathbf{R}_i$ is given out. Then, we choose $\overline{\text{C}}_i$ as a GSW ciphertext of 0. It also follows from the semantic security of GSW encryption.

(3) Finally, the distribution is totally irrelevant to the plaintext bit μ, which completes the proof.

Then, let us consider the compression rate of our construction. Recall that $t = (t' - 1)N$ is the compression parameter. In order to achieve a compression rate of $1 - \varepsilon$ for some small $\varepsilon > 0$, it is sufficient to set $t' \overset{\Delta}{=} (2/\varepsilon)$, so that the compression rate

$$\gamma = \frac{n \cdot n}{\overline{n} \cdot \overline{n}} = \left(\frac{t' - 1}{t'}\right)^2 \geq 1 - \varepsilon. \tag{34}$$

Thus, we need

$$\beta_\chi \cdot \text{poly}\left(\frac{rN}{\varepsilon}\right) = \beta_{\text{Comp}} \leq \frac{\lfloor q^{1/t'} \rfloor}{2} = \frac{\lfloor q^{\varepsilon/2} \rfloor}{2}. \tag{35}$$

Setting $\beta_\chi \leq \text{poly}(rN/\varepsilon)$, we need $q = \text{poly}(rN/\varepsilon)^{\Theta(1/\varepsilon)}$. This means that the semantic security of our construction is based on approximation lattices problems with gap $\text{poly}(rN/\varepsilon)^{\Theta(1/\varepsilon)}$. In particular, if we view the compression rate γ and maximal number of parties involved N as constants, then the hardness is only based on ones with polynomial gap. Formally, we have the following theorem.

Theorem 2. *For any $\varepsilon = \varepsilon(\lambda) > 0$, there exists a rate-$(1 - \varepsilon)$ compressible multikey fully homomorphic scheme that is semantically secure assuming the hardness of approximate lattices problems with gap $\text{poly}(\lambda N/\varepsilon)^{(1/\varepsilon)}$.*

4. Compressible Multi-Identity Fully Homomorphic Encryption

In this section, we propose the first compressible MIFHE scheme. All of the new techniques and analyses are similar to what we have done when constructing compressible MKFHE scheme.

4.1. Compressible Multi-Identity FHE. Similarly, we begin with the definition of the compressible MIFHE.

Definition 4. A compressible multi-identity fully homomorphic encryption scheme consists of seven PPT algorithms (Setup, Extract, Encrypt, Expand, Evaluate , Compress and Decrypt) defined as follows:

$\text{Setup}(1^\lambda, 1^N)$: Take a security parameter λ and a bound N on the number of identities involved as inputs, generate a master public key MPK and a master secret key MSK, and output (MPK, MSK). MPK is taken in as an input in all of the following algorithms; thus, we just omit it. The security parameter λ also defines an identity space $\mathcal{I}$.

$\text{Ext}(\text{MSK}, \text{id})$: Take the master secret key MSK and an identity $\text{id} \in \mathcal{I}$ as inputs; and extract and output a user-specific secret key SK_{id} for id.

$\text{Enc}(\text{id}, \mu \in \{0, 1\})$: Take an identity $\text{id} \in \mathcal{I}$ and a bit $\mu \in \{0, 1\}$ as inputs, and output a fresh low-rate ciphertext c_{id} under the identity id.

$\text{Exp}\left(\left\{\text{id}_j\right\}_{j\in[N]}, c_i\right)$: Take identities $\left\{\text{id}_j\right\}_{j\in[N]}$ and a fresh ciphertext c_i under the i^{th} identity id_i as inputs; and compute and output an expanded low-rate ciphertext $\widehat{c}_i$ under N's identities $(\text{id}_1, \text{id}_2, \ldots, \text{id}_N)$.

$\text{Eval}(f, \widehat{c} = (\widehat{c}_i)_{i\in[t]})$: Take a circuit f and a vector of expanded ciphertexts $\widehat{c} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_t)$ as inputs, corresponding to each input bit of f, and output another vector of evaluated low-rate ciphertext $\widehat{c} = (\widehat{c}'_1, \widehat{c}'_2, \ldots, \widehat{c}'_t)$, corresponding to each output bit of f.

$\text{Comp}(\widehat{c} = (\widehat{c}_i)_{i\in[t]})$: Take a vector of expanded or evaluated low-rate ciphertexts $\widehat{c} = (\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_t)$ as inputs, and output one or more compressed high-rate ciphertext(s) $c^* = \{c_i^*\}_{i\in[s]}$.

$\text{Dec}(\left\{\text{SK}_{\text{id}_i}\right\}_{i\in[N]}, c^*)$: Take N secret keys $(\text{SK}_{\text{id}_1}, \text{SK}_{\text{id}_2}, \ldots, \text{SK}_{\text{id}_N})$, corresponding to identities $(\text{id}_1, \text{id}_2, \ldots, \text{id}_N)$ as inputs, as well as the compressed ciphertext(s) c^*, and output a vector of message bits $\mathbf{m} = \{\mu_i\}_{i\in[t]}$ corresponding to $\{\widehat{c}_i\}_{i\in[t]}$.

The definitions of correctness and compression rate are exactly the same as those of the compressible MKFHE in Section 3.1. The selective security definition for compressible multi-identity fully homomorphic encryption is

the same as that for multi-identity fully homomorphic encryption, which is the same as that for identity-based encryption. We do not consider the distributed decryption for our MIFHE scheme, which can be used to construct round-efficient secure multiparty computation protocols, and instead put the main focus on the compressible expansion algorithm.

4.2. Public Parameters. Before giving our construction of the first compressible MIFHE scheme, we first describe some public parameters that will be used throughout the rest of this section.

(i) Let N be the maximum number of identities the scheme can support. The integer $n = \mathrm{poly}(\lambda, N)$ is a lattice dimension parameter. The modulus q is a sufficiently large number. Let $\ell = \lceil \log q \rceil$, $m_0 = n(\ell + O(1))$, $m_1 = n\ell$, and $m = m_0 + m_1$.

(ii) Given any matrix $\mathbf{A} \in \mathbb{Z}_q^{x \times y}$, by Lemma 2, there exists an efficiently randomized algorithm that samples a sub-Gaussian matrix $\mathbf{X}$ with some constant parameter $O(1)$ over $\mathbb{Z}^{x\ell \times y}$ such that $\mathbf{X} = \mathbf{G}_x^{-1}(\mathbf{A})$.

(iii) Let $\mathcal{D}$ be the distribution over $\{-1, 0, 1\}$ as defined in Subsection 2.1, such that $(\mathbf{A}_0, \mathbf{A}_0\mathbf{R})$ is $\mathrm{negl}(n)$-far from $(\mathbf{U}_0, \mathbf{U}_1) \xleftarrow{\$} \mathbb{Z}_q^{n \times m_0} \times \mathbb{Z}_q^{n \times m_1}$ for $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{n \times m_0}$ and $\mathbf{R} \leftarrow \mathcal{D}^{m_0 \times m_1}$. For more information, please refer to [26].

(iv) Every identity is assumed to have a counterpart element in $\mathrm{GF}(q^n)$. $\mathbf{H}$: $\mathrm{GF}(q^n) \longrightarrow \mathbb{Z}_q^{n \times n}$ is said to be an invertible difference, if $\mathbf{H}$ is computable in polynomial time in $n\ell$ and $\mathbf{H}(\mathrm{id}_1) - \mathbf{H}(\mathrm{id}_2)$ is invertible for any two different identities $\mathrm{id}_1, \mathrm{id}_2$. For more information, please refer to [15].

(v) The LWE error rate α should be sufficiently large satisfying $\alpha q \geq 2\sqrt{n}$ for the sake of security.

(vi) Let $t \triangleq (t' - 1)N$ be the compression parameter. The parameter t' will be specified in Subsection 4.4.2, All of the rows of the nearly square gadget matrix $\mathrm{H}' \in \mathbb{Z}_q^{(t'-1) \times t'}$ span the kernel space of its public trapdoor matrix $\mathrm{F}' \in \mathbb{Z}_q^{t' \times t'}$ having the properties stated in 2.4. Then, set $\mathbf{H} \triangleq \mathrm{H}' \otimes \mathbf{I}_{mN} \in \mathbb{Z}_q^{tm \times (t+N)m}$ and $\mathbf{F} \triangleq \mathrm{F}' \otimes \mathbf{I}_{mN} \in \mathbb{Z}_q^{(t+N)m \times (t+N)m}$.

4.3. Our Construction

C-MIFHE.Setup$(1^\lambda, 1^N)$: Choose $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{n \times m_0}$, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{n \times tm}$, and $\mathbf{R} \leftarrow \mathcal{D}^{m_0 \times m_1}$. Let $\mathbf{A} = [\mathbf{A}_0 \; -\mathbf{A}_0\mathbf{R}] \in \mathbb{Z}_q^{n \times m}$. Let the master secret key and the master public key be MSK = $\mathbf{R}$ and MPK = $[\mathbf{UA}]$. Output params = $(n, N, q, \ell, t', t, m_0, m_1, m, \mathbf{H}, \mathrm{H}', \mathbf{F}, \mathrm{F}')$, mpk, msk.

C-MIFHE.Ext(params, mpk, msk, $\{\mathrm{id}_i | i \in [N]\}$): Given $\mathrm{id}_i \in \mathbb{Z}_q^n$, compute $\mathbf{H}(\mathrm{id}_i)$ and set $\mathbf{A}_i = [\mathbf{A}_0 \; -\mathbf{A}_0\mathbf{R} + \mathbf{H}(\mathrm{id}_i) \cdot \mathbf{G}_n] \in \mathbb{Z}_q^{n \times m}$. Sample vectors $\mathbf{s}_i^{(k)} \in \mathbb{Z}^m$ for $k \in [tm]$ with small entries satisfying $\mathbf{A}_i \cdot \mathbf{s}_i^{(k)} = \mathbf{u}^{(k)}$ by running the algorithm Sample$D(\mathbf{R}, \mathbf{A}_0, \mathbf{H}(\mathrm{id}_i), \mathbf{u}^{(k)}, \|\mathbf{R}\|_2)$. Set $\mathbf{S}_i^T = [\mathbf{s}_i^{(1)}, \ldots, \mathbf{s}_i^{(tm)}] \in \mathbb{Z}_q^{m \times tm}$. Let the secret key and public key for i be $\mathrm{SK}_i = \overline{S}_i = [\mathbf{I}_{tm} \mathbf{S}_i] \in \mathbb{Z}_q^{tm \times (t+1)m}$ and $\mathrm{PK}_i = \overline{P}_i = [\mathbf{U} - \mathbf{A}_i] \in \mathbb{Z}_q^{n \times (t+1)m}$. Note that $\overline{S}_i \cdot \overline{P}_i^T = 0$. Output $(\mathrm{PK}_i, \mathrm{SK}_i)_{i \in [N]}$.

C-MIFHE.Enc(params, $\mathrm{PK}_{\mathrm{id}_i}$, id_i, $\mu \in \{0, 1\}$): Given a message bit μ, randomly select two matrices $\overline{Y}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times (t+1)m\ell}$ and $\mathbf{E}_i = \begin{bmatrix} -\mathbf{E}_i^{(0)} \\ -\mathbf{E}_i^{(1)} \\ \mathbf{E}_i^{(2)} \end{bmatrix} \in \mathbb{Z}^{(t+1)m \times (t+1)m\ell}$, where $\mathbf{E}_i^{(0)} \leftarrow \mathcal{D}_{\mathbb{Z}, \alpha q}^{tm \times (t+1)m\ell}$, $\mathbf{E}_i^{(1)} = [\mathbf{e}_i^{(1,1)}, \mathbf{e}_i^{(1,2)}, \ldots, \mathbf{e}_i^{(1,(t+1)m\ell)}] \leftarrow \mathcal{D}_{\mathbb{Z}, \alpha q}^{m_0 \times (t+1)m\ell}$, and $\mathbf{E}_i^{(2)} = [\mathbf{e}_i^{(2,1)}, \mathbf{e}_i^{(2,2)}, \ldots, \mathbf{e}_i^{(2,(t+1)m\ell)}]$, where $\mathbf{e}_i^{(2,j)} \leftarrow \mathcal{D}_{\mathbb{Z}, s_j}^{m_1}$ for $s_j^2 = (\|\mathbf{e}_i^{(1,j)}\|_2^2 + m_0(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$. Set

$$\overline{\mathrm{C}}_i = \mu \mathbf{G}_{(t+1)m} + \overline{P}_i^T \overline{Y}_i + \mathbf{E}_i \in \mathbb{Z}_q^{(t+1)m \times (t+1)m\ell}. \tag{36}$$

Remark 1. The setting of error parameters, especially $\mathbf{E}_i^{(2)}$, plays a critical role in the security proof. For more information, please refer to [30, 31].

Then, partition $\overline{\mathrm{C}}_i$ as follows:

$$\overline{\mathrm{C}}_i \triangleq \begin{bmatrix} \overline{\mathrm{C}}_i^{(1,1)} & \overline{\mathrm{C}}_i^{(1,2)} \\ \overline{\mathrm{C}}_i^{(2,1)} & \overline{\mathrm{C}}_i^{(2,2)} \end{bmatrix}, \tag{37}$$

where $\overline{\mathrm{C}}_i^{(1,1)} \in \mathbb{Z}_q^{tm \times tm\ell}$, $\overline{\mathrm{C}}_i^{(1,2)} \in \mathbb{Z}_q^{tm \times m\ell}$, $\overline{\mathrm{C}}_i^{(2,1)} \in \mathbb{Z}_q^{m \times tm\ell}$, $\overline{\mathrm{C}}_i^{(2,2)} \in \mathbb{Z}_q^{m \times m\ell}$.

For every $j \in [N] \backslash \{i\}$, randomly select two matrices $\mathbf{Y}_{i,j} \xleftarrow{\$} \mathbb{Z}_q^{n \times m\ell}$ and $\mathbf{E}_{i,j} = \begin{bmatrix} -\mathbf{E}_{i,j}^{(1)} \\ \mathbf{E}_{i,j}^{(2)} \end{bmatrix} \in \mathbb{Z}^{m \times m\ell}$, where $\mathbf{E}_{i,j}^{(1)} = [\mathbf{e}_{i,j}^{(1,1)}, \mathbf{e}_{i,j}^{(1,2)}, \ldots, \mathbf{e}_{(i,j)}^{(1,m\ell)}] \leftarrow \mathcal{D}_{\mathbb{Z}, \alpha q}^{m_0 \times m\ell}$, and $\mathbf{E}_{i,j}^{(2)} = [\mathbf{e}_{i,j}^{(2,1)}, \mathbf{e}_{i,j}^{(2,2)}, \ldots, \mathbf{e}_{i,j}^{(2,m\ell)}]$, where $\mathbf{e}_{i,j}^{(2,k)} \leftarrow \mathcal{D}_{\mathbb{Z}, s_k}^{m_1}$ for $s_k^2 = (\|\mathbf{e}_{i,j}^{(1,k)}\|_2^2 + m_0(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$. Set

$$\mathbf{C}_{i,j} = \mu \mathbf{G}_m + \mathbf{A}_j^T \mathbf{Y}_{i,j} + \mathbf{E}_{i,j} \in \mathbb{Z}_q^{m \times m\ell}. \tag{38}$$

Then, decompose the matrix $\mathbf{Y}_{i,j}$ that has every entry in $\mathbb{Z}_q$ to matrices in binary representation $\mathbf{Y}_{i,j} \triangleq \sum_{k=0}^{\lfloor \log q \rfloor} 2^k \cdot \mathbf{Y}_{i,j}^{(k)}$, where $\mathbf{Y}_{i,j}^{(k)} \in \{0, 1\}^{n \times m\ell}$.

For $x \in [t]$, $y \in [n]$, $b \in [ml]$, $k + 1 \in [\ell]$, compute

$$\begin{aligned} \mathbf{V}_{i,j,1}^{(x,y,b,k)} &= \mathbf{U}^T \mathbf{Y}_{i,j}^{(x,y,b,k)} + \mathbf{E}_{i,j,1}^{(x,y,b,k)} + \mathbf{Y}_{i,j}^{(k)}[y, b] \mathbf{G}_m^{(x)} \in \mathbb{Z}_q^{tm \times m\ell}; \\ \mathbf{V}_{i,j,2}^{(x,y,b,k)} &= \mathbf{A}_i^T \mathbf{Y}_{i,j}^{(x,y,b,k)} + \mathbf{E}_{i,j,2}^{(x,y,b,k)} \in \mathbb{Z}_q^{m \times m\ell}, \end{aligned} \tag{39}$$

where $\mathbf{Y}_{i,j}^{(x,y,b,k)} \xleftarrow{\$} \mathbb{Z}_q^{n \times m\ell}$, $\mathbf{E}_{i,j,2}^{(x,y,b,k)} \in \mathbb{Z}^{m \times m\ell}$ is extracted from the same distribution of $\mathbf{E}_{i,j}$, and $\mathbf{E}_{i,j,1}^{(x,y,b,k)} =$

$\begin{bmatrix} -\mathbf{E}_{i,j,1}^{(x,y,b,k),0} \\ \mathbf{E}_{i,j,1}^{(x,y,b,k),1} \end{bmatrix} \in \mathbb{Z}^{tm\times m\ell}$, where $\mathbf{E}_{i,j,1}^{(x,y,b,k),0} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}^{(t-1)m\times m\ell}$ and $\mathbf{E}_{i,j,1}^{(x,y,b,k),1} \in \mathbb{Z}^{m\times m\ell}$ is extracted from the same distribution of $\mathbf{E}_{i,j,2}^{(x,y,b,k)}$, and $\mathbf{G}_m^{(x)} = \begin{bmatrix} 0^{(x-1)m\times m\ell} \\ \mathbf{G}_m \\ 0^{(t-x)m\times m\ell} \end{bmatrix} \in \mathbb{Z}^{tm\times m\ell}$.

Set $\mathbf{U}_i \triangleq (\mathbf{C}_{i,j}, \mathbf{V}_{i,j,1}^{(x,y,b,k)}, \mathbf{V}_{i,j,2}^{(x,y,b,k)})$, where $j \in [N]\backslash\{i\}, x \in [t], y \in [n], b \in [ml], k+1 \in [\ell]$. Output a tuple of encryption $\mathrm{CIPHER}_i \triangleq (\overline{\mathrm{C}}_i, \mathbf{U}_i)$.

C-MIFHE.Exp(params, $\{\mathrm{id}_i\}_{i\in[N]}$, CIPHER_i): For every $j \in [N]\backslash\{i\}, x \in [t], y \in [n], b \in [ml], k+1 \in [\ell]$, compute

$$\mathbf{Z}_{i,j}^{(x,y,b)}[z_1, z_2] \triangleq \begin{cases} -\mathbf{A}_j[z_1 + xr, y], & \text{if } z_2 = b, \\ 0, & \text{otherwise,} \end{cases} \tag{40}$$

that is, $\mathbf{Z}_{i,j}^{(x,y,b,k)} \triangleq \sum_{z_1=1}^{m} 2^k \mathbf{U}^T[z_1 + xm, y] \cdot \mathbf{E}_{z_1,b}' \in \mathbb{Z}_q^{m\times m\ell}$, where $\mathbf{E}_{z_1,b}'$ is an $m \times m\ell$ matrix that has 1 in the z_1^{th} row and the b^{th} column and 0's in all other entries. Set

$$\overline{Z}_{i,j}^{(x,y,b)} \triangleq \begin{bmatrix} 0^{(x-1)r\times r\ell} \\ \mathbf{Z}_{i,j}^{(x,y,b)} \\ 0^{(t-x)r\times r\ell} \end{bmatrix} \in \mathbb{Z}_q^{tr\times r\ell}. \tag{41}$$

Compute

$$\mathbf{X}_{i,j}^{(1)} \triangleq \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1} \mathbf{V}_{i,j,1}^{(x,y,b,k)} \cdot \mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) \in \mathbb{Z}_q^{tm\times m\ell};$$

$$\mathbf{X}_{i,j}^{(2)} \triangleq \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1} \mathbf{V}_{i,j,2}^{(x,y,b,k)} \cdot \mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) \in \mathbb{Z}_q^{m\times m\ell}. \tag{42}$$

Finally, expand a fresh ciphertext to an expanded one.

$$\widehat{\mathrm{C}}_i \triangleq \begin{bmatrix} \overline{\mathrm{C}}_i^{(1,1)} & -\mathbf{X}_{i,1}^{(1)} & \cdots & \overline{\mathrm{C}}_i^{(1,2)} & \cdots & -\mathbf{X}_{i,N}^{(1)} \\ & \mathbf{C}_{i,1} & & & & \\ & & \ddots & & & \\ \overline{\mathrm{C}}_i^{(2,1)} & \mathbf{X}_{i,1}^{(2)} & \cdots & \overline{\mathrm{C}}_i^{(2,2)} & \cdots & \mathbf{X}_{i,N}^{(2)} \\ & & & & \ddots & \\ & & & & & \mathbf{C}_{i,N} \end{bmatrix} \in \mathbb{Z}_q^{(t+N)m\times(t+N)m\ell}. \tag{43}$$

Output $\widehat{\mathrm{C}}_i$.

C-MIFHE.Comp$\left(\text{params}, \left\{\widehat{\mathrm{C}}_{u,v,\omega}\right\}_{u,v\in[tm],\omega\in[\ell]}\right)$: In this algorithm, a series of low-rate expanded (or evaluated) ciphertexts are compressed into a high-rate ciphertext. Let $\mathbf{T}_{u,v} \triangleq \begin{bmatrix} \mathbf{E}_{u,v}' \\ 0^{Nm\times tm} \end{bmatrix} \in \mathbb{Z}^{(t+N)m\times tm}$, where $\mathbf{E}_{u,v}'$ is a $tm \times tm$ matrix that has 1 in the u^{th} row and the v^{th} column and 0's in all other entries. Set the compressed ciphertext as

$$\mathbf{C}^* \triangleq \sum_{u\in[tm]}\sum_{v\in[tm]}\sum_{\omega\in[\ell]} \widehat{\mathrm{C}}_{u,v,\omega} \times \mathbf{G}_{(t+N)m}^{-1}\left(2^\omega \cdot \mathbf{T}_{u,v} \times \mathbf{H}\right) \in \mathbb{Z}_q^{(t+N)m\times(t+N)m}. \tag{44}$$

Note that the nearly square gadget matrix $\mathbf{H} \in \mathbb{Z}_q^{tm\times(t+N)m}$ and its public trapdoor $\mathbf{F} \in \mathbb{Z}_q^{(t+N)m\times(t+N)m}$ play a significant role in the compression and compressed decryption algorithms.

C-MIFHE.CompDec(params, $\mathbf{C}^*$, $(\mathrm{SK}_i)_{i\in[N]}$): Given a compressed ciphertext $\mathbf{C}^*$, set the combined key $\widehat{\mathrm{S}} \triangleq [\mathbf{I}_{tm}, \mathbf{S}_1, \ldots, \mathbf{S}_N] \in \mathbb{Z}^{tm\times(t+N)m}$. The compressed decryption is composed of four steps:

(1) $\mathbf{Z} \triangleq \mathbf{S} \times \mathbf{C}^* \pmod q$;
(2) $\mathbf{Y} \triangleq \mathbf{Z} \times \mathbf{F} \pmod q$, where $\mathbf{F} \in \mathbb{Z}_q^{(t+N)m\times(t+N)m}$ is the public trapdoor matrix;
(3) $\mathbf{X} \triangleq \mathbf{Y} \times \mathbf{F}^{-1}$ (note that $\mathbf{F}$ is full rank over R);
(4) $M' \triangleq (\mathbf{Z} - \mathbf{X}) \times \mathbf{H}^{-1} \pmod q$ (note that $\mathbf{H}$ is a row full-rank matrix modulo q, so that there exists a matrix $\mathbf{H}^{-1} \in \mathbb{Z}_q^{(t+N)m\times tm}$ such that $\mathbf{H} \times \mathbf{H}^{-1} = \mathbf{I}_{tm}$).

C-MIFHE.NAND(params, $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2$): Given two expanded ciphertext matrices $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2$ for two plaintexts μ_1, μ_2, homomorphic NAND operation is defined as

$$\widehat{\mathrm{C}}_{\mathrm{NAND}} \triangleq \mathbf{G}_{(t+N)m} - \widehat{\mathrm{C}}_1 \cdot \mathbf{G}_{(t+N)m}^{-1}\left(\widehat{\mathrm{C}}_2\right). \tag{45}$$

Output $\widehat{\mathrm{C}}_{\mathrm{NAND}}$. Observe that $\mathbf{G}_x^{-1}(\cdot)$ is randomized, and so is this algorithm.

C-MIFHE.Eval(params, $f, \widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2, \ldots, \widehat{\mathrm{C}}_s$): An NAND-circuit $f: \{0,1\}^t \longrightarrow \{0,1\}$ is applied to a set of ciphertexts $\widehat{\mathrm{C}}_1, \widehat{\mathrm{C}}_2, \ldots, \widehat{\mathrm{C}}_s$, which leads to a ciphertext $\widehat{\mathrm{C}}_f$.

4.4. Setting the Parameters. In this subsection, the correctness, homomorphic properties, security, and compression rate of the proposed compressible MIFHE are analyzed. If we use a somewhat (or leveled) version without bootstrapping, then the error bound is dependent on the maximal multiplication depth of allowed evaluated functions. In this paper, we would like to give an asymptotic analysis of various parameters of a fully homomorphic one with bootstrapping, so that all ciphertexts before being packed have an error bound similar to that of freshly expanded ciphertexts.

4.4.1. Correctness and Homomorphism. The correctness of compressed decryption may be less obvious, since the encryption and expansion algorithms are quite different from previous ones. During encryption, it holds that, for every identity id_i and $\left\{\mathrm{id}_j\right\}_{j\neq i}$,

$$\overline{S}_i \times \overline{C}_i = \mu\overline{S}_i\mathbf{G}_{(t+1)m} + \overline{S}_i\mathbf{E}_i \in \mathbb{Z}_q^{tm\times(t+1)m\ell}, \tag{46}$$

because $\overline{S}_i \times \overline{P}_i^T = 0$. One can calculate the error bound $\beta_{\mathrm{Enc}} \triangleq \alpha q m_0 \cdot \omega(\sqrt{\log n})^3$; that is, $\max_{a,b}|\overline{S}_i\mathbf{E}_i[a,b]| < \beta_{\mathrm{Enc}}$ with overwhelming probability.

Besides, for the additional helper information matrices, we have

$$\begin{aligned}\mathbf{S}_j \times \mathbf{C}_{i,j} &= \mu \mathbf{S}_j \mathbf{G}_m + \mathbf{U}^T \mathbf{Y}_{i,j} + \mathbf{S}_j \mathbf{E}_{i,j} \in \mathbb{Z}_q^{tm \times m\ell};\\ \mathbf{I}_{tm} \times \mathbf{V}_{i,j,1}^{(x,y,b,k)} &= \mathbf{U}^T \mathbf{Y}_{i,j}^{(x,y,b,k)} + \mathbf{E}_{i,j,1}^{(x,y,b,k)} + \mathbf{Y}_{i,j}^{(k)}[y,b]\mathbf{G}_m^{(x)}\\ &\in \mathbb{Z}_q^{tm \times m\ell};\\ \mathbf{S}_i \times \mathbf{V}_{i,j,2}^{(x,y,b,k)} &= \mathbf{U}^T \mathbf{Y}_{i,j}^{(x,y,b,k)} + \mathbf{S}_i \mathbf{E}_{i,j,2}^{(x,y,b,k)} \in \mathbb{Z}_q^{tm \times m\ell},\end{aligned} \tag{47}$$

because $\mathbf{S}_j \times \mathbf{A}_j^T = \mathbf{U}^T$. Similarly, one can easily verify that, with overwhelming probability,

$$\left|\mathbf{S}_j \mathbf{E}_{i,j}[a,b]\right|, \left|\mathbf{E}_{i,j,1}^{(x,y,b,k)}[c,d]\right|, \left|\mathbf{S}_i \mathbf{E}_{i,j,2}^{(x,y,b,k)}[e,f]\right| < \beta_{\mathrm{Enc}}. \tag{48}$$

During the expansion algorithm, those weird matrices $\mathbf{X}_{i,j}^{(1)}$ and $\mathbf{X}_{i,j}^{(2)}$ are customized to eliminate the undesired $\mathbf{U}^T\mathbf{Y}_{i,j}$ in $\mathbf{S}_j\mathbf{C}_{i,j}$. In particular, we have

$$\begin{aligned}\mathbf{I}_{tm} \times \mathbf{X}_{i,j}^{(1)} &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1} \mathbf{V}_{i,j,1}^{(x,y,b,k)} \cdot \mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right)\\ &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\left(\mathbf{U}^T\mathbf{Y}_{i,j}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) + \mathbf{E}_{i,j,1}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) + \underline{\mathbf{Y}_{i,j}^{(k)}[y,b]\overline{Z}_{i,j}^{(x,y,b,k)}}\right)\\ &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\left(\mathbf{U}^T\mathbf{Y}_{i,j}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) + \mathbf{E}_{i,j,1}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right)\right) + \underline{\mathbf{U}^T\mathbf{Y}_{i,j}},\end{aligned} \tag{49}$$

$$\begin{aligned}\mathbf{S}_i \times \mathbf{X}_{i,j}^{(2)} &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1} \mathbf{S}_i\mathbf{V}_{i,j,2}^{(x,y,b,k)} \cdot \mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right)\\ &= \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\left(\mathbf{U}^T\mathbf{Y}_{i,j}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right) + \mathbf{S}_i\mathbf{E}_{i,j,2}^{(x,y,b,k)}\mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right)\right).\end{aligned} \tag{50}$$

The underlined part can be a tricky step. We now analyze it as follows:

$$\begin{aligned}&\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\mathbf{Y}_{i,j}^{(k)}[y,b]\sum_{x=1}^{t}\left(\overline{Z}_{i,j}^{(x,y,b,k)}\right)\\ &= \sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\mathbf{Y}_{i,j}^{(k)}[y,b]\sum_{a=1}^{tm}\left(2^k \cdot \mathbf{U}^T[a,y] \times \mathbf{E}'_{a,b}\right)\\ &= \sum_{a=1}^{tm}\sum_{y=1}^{n}\sum_{b=1}^{ml}\mathbf{U}^T[a,y] \cdot \mathbf{Y}_{i,j}[y,b] \times \mathbf{E}'_{a,b},\\ &= \mathbf{U}^T\mathbf{Y}_{i,j},\end{aligned} \tag{51}$$

where $\mathbf{E}'_{a,b}$ is a $tm \times m\ell$ matrix that has 1 in the a^{th} row and the b^{th} column and 0's in all other entries.

Thus, for any expanded ciphertext, we have

$$\begin{aligned}\widehat{S} \times \widehat{\mathrm{C}}_i = \Bigg[&\overline{\mathrm{C}}_i^{(1,1)} + \mathbf{S}_i\overline{\mathrm{C}}_i^{(2,1)}\Big\|-\mathbf{X}_{i,1}^{(1)} + \mathbf{S}_1\mathbf{C}_{i,1} + \mathbf{S}_i\mathbf{X}_{i,j}^{(2)}\Big\| \cdots\\ &\cdot \Big\|\overline{\mathrm{C}}_i^{(1,2)} + \mathbf{S}_i\overline{\mathrm{C}}_i^{(2,2)}\Big\| \cdots \Big\| \| - \mathbf{X}_{i,N}^{(1)} + \mathbf{S}_N\mathbf{C}_{i,N} + \mathbf{S}_i\mathbf{X}_{i,N}^{(2)}\Bigg]\\ &\in \mathbb{Z}^{tm \times (t+N)m}.\end{aligned} \tag{52}$$

There are only two types of all blocks in the above seemingly cumbersome matrix. The first comes from equation (47) by partitioning

$$\begin{aligned}&\left[\mathbf{I}_{tm}\|\,\mathbf{S}_i\right] \times \begin{bmatrix}\overline{\mathrm{C}}_i^{(1,1)} & \overline{\mathrm{C}}_i^{(1,2)}\\ \overline{\mathrm{C}}_i^{(2,1)} & \overline{\mathrm{C}}_i^{(2,2)}\end{bmatrix}\\ &= \mu\left[\mathbf{I}_{tm}\mathbf{G}_{tm}\|\,\mathbf{S}_i\mathbf{G}_m\right] + \overline{\mathrm{S}}_i\mathbf{E}_i\\ &\triangleq \left[\mu\mathbf{I}_{tm}\mathbf{G}_{tm} + \widetilde{E}_0\|\,\mu\mathbf{S}_i\mathbf{G}_m + \widetilde{E}_i\right] \in \mathbb{Z}_q^{tm \times (t+1)m\ell}.\end{aligned} \tag{53}$$

The second comes from equations (48), (49), and (51):

$$\begin{aligned}&\mathbf{S}_j\mathbf{C}_{i,j} - \mathbf{X}_{i,j}^{(1)} + \mathbf{S}_i\mathbf{X}_{i,j}^{(2)}\\ &= \mu\mathbf{S}_j\mathbf{G}_m + \mathbf{S}_j\mathbf{E}_{i,j} + \sum_{x=1}^{t}\sum_{y=1}^{n}\sum_{b=1}^{ml}\sum_{k=0}^{\ell-1}\left(\mathbf{S}_i\mathbf{E}_{i,j,2}^{(x,y,b,k)} - \mathbf{E}_{i,j,1}^{(x,y,b,k)}\right)\\ &\quad \mathbf{G}_m^{-1}\left(\mathbf{Z}_{i,j}^{(x,y,b,k)}\right)\\ &\triangleq \mu\mathbf{S}_j\mathbf{G}_m + \widetilde{E}_j \in \mathbb{Z}_q^{tm \times m\ell}.\end{aligned} \tag{54}$$

Finally, we get the most important decryption invariant in all GSW-like encryptions from equations (52)–(54).

$$\begin{aligned}\widehat{S} \times \widehat{\mathrm{C}}_i &= \left[\mu\mathbf{I}_{tm}\mathbf{G}_{tm} + \widetilde{E}_0\Big\|\mu\mathbf{S}_1\mathbf{G}_m + \widetilde{E}_1\Big\| \cdots \Big\|\mu\mathbf{S}_N\mathbf{G}_m + \widetilde{E}_N\right]\\ &\triangleq \mu\widehat{S}\mathbf{G}_{(t+N)m} + \widehat{E} \in \mathbb{Z}_q^{tm \times (t+N)m\ell}.\end{aligned} \tag{55}$$

In terms of the error bound, we have $\beta_{\mathrm{Exp}} \triangleq tnm^2\ell^3\beta_{\mathrm{Enc}}$; that is, $\max_{a,b}|\widehat{\mathbf{E}}[a,b]| < \beta_{\mathrm{Exp}}$ with overwhelming probability.

With equation (55) in mind, everything turns familiar again. Homomorphic properties are obvious, and one can find them in most GSW-like FHE (or MKFHE) schemes. Thus, we just omit them here.

On the other hand, the correctness of compressed decryption also turns obvious. According to equation (55), we have

$$\begin{aligned}\widehat{S}\times\mathbf{C}^{*} &= \sum_{u\in[tm]}\sum_{v\in[tm]}\sum_{\omega\in[\ell]}\widehat{S}\widehat{C}_{u,v,\omega}\times\mathbf{G}^{-1}_{(t+N)m}\left(2^{\omega}\cdot\mathbf{T}_{u,v}\mathbf{H}\right)\\ &= \sum_{u,v,\omega}\left(\mu_{u,v,\omega}\widehat{S}\mathbf{G}_{(t+N)m}+\widehat{E}\right)\mathbf{G}^{-1}_{(t+N)m}\left(2^{\omega}\cdot\mathbf{T}_{u,v}\mathbf{H}\right)\\ &\triangleq \sum_{u,v,\omega}\left(2^{\omega}\mu_{u,v,\omega}\widehat{S}\mathbf{T}_{u,v}\mathbf{H}\right)+\mathbf{E}^{*}\\ &= \sum_{u,v,\omega}\left(2^{\omega}\mu_{u,v,\omega}\mathbf{E}'_{u,v}\right)\mathbf{H}+\mathbf{E}^{*}\\ &\triangleq \mathbf{MH}+\mathbf{E}^{*}\in\mathbb{Z}_{q}^{tm\times(t+N)m},\end{aligned}\tag{56}$$

where $\mathbf{E}'_{u,v}$ is a $tm\times tm$ matrix that has 1 in the u^{th} row and the v^{th} column and 0's in all other entries, and $\mathbf{M}\in\mathbb{Z}_{q}^{tm\times tm}$ is a compressed plaintext matrix that can be obtained from $\mathbf{Z}\triangleq\widehat{S}\times\mathbf{C}^{*}=\mathbf{MH}+\mathbf{E}^{*}$, by using Gentry and Halevi's idea of the nearly square gadget matrix, which was briefly described in 2.4. That is,

(1) $\mathbf{Y}\triangleq\mathbf{Z}\times\mathbf{F}=\mathbf{E}^{*}\mathbf{F}\ (\text{mod}\ q)$, where $\mathbf{F}\in\mathbb{Z}_{q}^{(t+N)m\times(t+N)m}$ is the public trapdoor matrix;

(2) $\mathbf{X}\triangleq\mathbf{Y}\times\mathbf{F}^{-1}=\mathbf{E}^{*}$ (note that $\mathbf{F}$ is full rank over R);

(3) $M'\triangleq(\mathbf{Z}-\mathbf{X})\times\mathbf{H}^{-1}=\mathbf{M}\ (\text{mod}\ q)$ (note that $\mathbf{H}$ is a row full-rank matrix modulo q, so that there exists a matrix $\mathbf{H}^{-1}\in\mathbb{Z}_{q}^{(t+N)m\times tm}$ such that $\mathbf{H}\times\mathbf{H}^{-1}=\mathbf{I}_{tm}$).

The matrix M' is the desired one if every entry of $\mathbf{E}^{*}\mathbf{F}$ does not wrap around $\mathbb{Z}_{q}$; that is, $(\mathbf{E}^{*}\mathbf{F}\text{mod}q)=\mathbf{E}^{*}\mathbf{F}$ over the integers. As for the error bound, we have $\beta_{\text{Comp}}\triangleq t^{2}(t+N)m^{3}\ell^{2}\beta_{\text{Exp}}$, i.e., $\max_{a,b}|\mathbf{E}^{*}[a,b]|<\beta_{\text{Comp}}$ with overwhelming probability. On the other hand, by the construction of the public trapdoor matrix $\mathbf{F}$, it is required that $\beta_{\text{Comp}}\le\lfloor q^{1/t'}\rfloor$.

4.4.2. Security and Compression Rate. The security level and compression rate of our construction remain to be shown.

The additional expansion and compression algorithms are public and do not compromise security. The key-extraction algorithm is similar to that of the underlying IBE in [15, 26] (ABB-IBE). In addition, during the encryption procedure, a fresh ciphertext for the i^{th} user CIPHER_{i} consists of three parts $\overline{\mathbf{C}}_{i}$, $\mathbf{C}_{i,j}$, and $(\mathbf{V}_{i,j,1}^{(x,y,b,k)},\mathbf{V}_{i,j,2}^{(x,y,b,k)})$, of which randomness matrices are mutually independent, all in a similar form. Exploiting a standard hybrid analysis, one can easily verify that the selective security of our compressible MIFHE is based solely on the DLWE assumption.

The view of the attacker is the following distribution: $(\text{params},\text{MPK},\text{PK}_{i},\overline{\mathbf{C}}_{i},\mathbf{U}_{i}=(\mathbf{C}_{i,j},\mathbf{V}_{i,j,1}^{(x,y,b,k)},\mathbf{V}_{i,j,2}^{(x,y,b,k)}))$, where $j\in[N]\backslash\{i\},x\in[t],y\in[n],b\in[ml],k+1\in[\ell]$ generated via $(\text{params},\text{MPK},\text{MSK})\leftarrow C-\text{MIFHE}\cdot\text{Setup}(1^{\lambda},1^{N})$, $(\text{PK}_{i},\text{SK}_{i})\leftarrow C-\text{MIFHE}\cdot\text{KeyGen}(\text{params})$, and $(\overline{\mathbf{C}}_{i},\mathbf{U}_{i})\leftarrow C-\text{MIFHE}\cdot\text{Enc}(\text{params},\text{PK}_{i},\mu)$, where $\mu\in\{0,1\}$.

We prove the selective security of our construction by relying on the selective security of the underlying ABB-IBE. The proof consists of the following hybrids:

(1) Firstly, we alter each of the ciphertexts $(\mathbf{V}_{i,j,1}^{(x,y,b,k)},\mathbf{V}_{i,j,2}^{(x,y,b,k)})$, where $j\in[N]\backslash\{i\},x\in[t],y\in[n],b\in[ml],k+1\in[\ell]$, so that, instead of being ciphertexts of $\mathbf{Y}_{i,j}^{(k)}[y,b]$, we change them to ciphertexts of 0. It follows from the selective security of ABB-IBE, which is based solely on the DLWE assumption.

(2) After the first step, no information about the encryption randomness $\mathbf{Y}_{i,j}$ is given out. Then, we choose $\overline{\mathbf{C}}_{i},\mathbf{C}_{i,j}$ as ciphertexts of 0. It also follows from the selective security of ABB-IBE.

(3) Finally, the distribution is totally irrelevant to the plaintext bit μ, which completes the proof.

Then, let us consider the compression rate of our construction. Recall that $t=(t'-1)N$ is the compression parameter. In order to achieve a compression rate of $1-\varepsilon$ for some small $\varepsilon>0$, it is sufficient to set $t'\triangleq(2/\varepsilon)$, so that the compression rate

$$\gamma=\frac{tm\cdot tm}{(t+N)m\cdot(t+N)m}=\left(\frac{t'-1}{t'}\right)^{2}\ge 1-\varepsilon.\tag{57}$$

Thus, we need

$$\alpha q\cdot\text{poly}\left(\frac{nN}{\varepsilon}\right)=\beta_{\text{Comp}}\le\frac{\lfloor q^{1/t'}\rfloor}{2}=\frac{\lfloor q^{\varepsilon/2}\rfloor}{2}.\tag{58}$$

Setting $\alpha q=2\sqrt{n}$, we need $q=\text{poly}(nN/\varepsilon)^{\Theta(1/\varepsilon)}$. This means that the selective security of our construction is based on approximation lattices problems with gap $\text{poly}(nN/\varepsilon)^{\Theta(1/\varepsilon)}$. In particular, if we view the compression rate γ and maximal number of parties involved N as constants, then the hardness is only based on ones with polynomial gap. Formally, we have the following theorem.

Theorem 3. *For any $\varepsilon=\varepsilon(\lambda)>0$, there exists a rate-$(1-\varepsilon)$ compressible multi-identity fully homomorphic scheme that is selectively secure assuming the hardness of approximate lattices problems with gap $\text{poly}(\lambda N/\varepsilon)^{(1/\varepsilon)}$.*

5. Conclusion

Many outsourced computations require homomorphic evaluations on data provided by different owners (often mutually distrusting), thus encrypted using their own keys. Single-key FHE only allows homomorphic evaluation over ciphertexts encrypted under the same key, while MKFHE or MIFHE under different keys or different identities thus solves this issue. However, the compression rate (the ratio of plaintext size to the ciphertext size) is often too small to be tolerated. Our main technical contribution is that we proposed a new compressible expansion algorithm. Furthermore, we presented the first compressible MKFHE

scheme that is semantically secure and the first compressible MIFHE scheme that is selectively secure, both under the decisional learning with error assumption, and both can reach an optimal compression rate. Our future direction is to concretely construct a homomorphic decompression algorithm to unpack compressed ciphertexts, which is a limitation of our constructions because homomorphic evaluations (except some special ones) are not allowed after compression.

Data Availability

No data were required in this work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2017YFB0802000), the National Natural Science Foundation of China (nos. U1705264, 61972124, 61672030, and 11974096), the Zhejiang Provincial Natural Science Foundation of China (no. LY19F020019), the Research Foundation of Guangxi Key Laboratory of Cryptography and Information Security (no. GCIS201725), and the Research Foundation of Hangzhou Normal University (no. 2017QDL002).

References

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," *STOC*, vol. 2009, pp. 169–178, 2009.

[2] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," *EUROCRYPT 2010. LNCS*, vol. 6110, pp. 24–43, 2010.

[3] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," *CRYPTO 2012. LNCS*, vol. 7417, pp. 868–886, 2012.

[4] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," *CRYPTO 2013, Part I. LNCS*, vol. 8042, pp. 75–92, 2013.

[5] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," *STOC*, vol. 2012, pp. 1219–1234, 2012.

[6] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," *CRYPTO 2015, Part II, LNCS 9216*, vol. 2015, pp. 630–656, 2015.

[7] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," *EUROCRYPT 2016, Part I. LNCS*, vol. 9665, pp. 735–763, 2016.

[8] C. Peikert and S. Shiehian, "Multi-key FHE from LWE, revisited," 2020.

[9] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," *CRYPTO 2016, Part I. LNCS*, vol. 9814, pp. 190–213, 2016.

[10] L. Chen, Z. Zhang, and X. Wang, "Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension," *TCC 2017. LNCS*, vol. 10678, pp. 597–627, 2017.

[11] E. Kim, H. S. Lee, and J. Park, "Towards round-optimal secure multiparty computations: multikey FHE without a CRS," *ACISP 2018, LNCS*, vol. 10946, pp. 101–113, 2018.

[12] H. Chen, I. Chillotti, and Y. Song, "Multi-key homomorphic encryption from TFHE," *Lecture Notes in Computer Science*, vol. 2019, pp. 446–472, 2019.

[13] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.

[14] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," *STOC*, vol. 2008, pp. 197–206, 2008.

[15] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," *EUROCRYPT 2010, LNCS*, vol. 6110, pp. 553–572, 2010.

[16] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *FOCS*, vol. 2011, pp. 97–106, 2011.

[17] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully homomorphic encryption without bootstrapping," *ITCS*, vol. 2012, pp. 309–325, 2012.

[18] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," *CRYPTO 2008. LNCS*, vol. 5157, pp. 554–571, 2008.

[19] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," 2020.

[20] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," *PKC 2013, LNCS*, vol. 7778, pp. 1–13, 2013.

[21] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," *PKC 2015*, vol. 9020, pp. 699–715, 2015.

[22] N. Genise, C. Gentry, S. Halevi, B. Li, and D. Micciancio, "Homomorphic encryption for finite automata," *ASIACRYPT 2019. LNCS*, vol. 11922, pp. 473–502, 2019.

[23] C. Gentry and S. Halevi, "Compressible FHE with applications to PIR," *TCC 2019. LNCS*, vol. 11892, pp. 438–464, 2019.

[24] Z. Brakerski, N. Dottling, S. Garg, and G. Malavolta, "Leveraging linear decryption: rate-1 fully-homomorphic encryption and time-lock puzzles," *TCC 2019. LNCS*, vol. 11892, pp. 407–437, 2019.

[25] O. Regev, "On lattices, learning with errors, random linear codes, and Cryptography," *STOC*, vol. 2005, pp. 84–93, 2005.

[26] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," *EUROCRYPT 2012. LNCS*, vol. 7237, pp. 700–718, 2012.

[27] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," *CRYPTO 2014, Part I, LNCS*, vol. 8618, pp. 297–314, 2014.

[28] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classal hardness of learning with errors," *STOC*, vol. 2013, pp. 575–584, 2013.

[29] C. Peikert, "Public key cryptosystems from the worst-case shortest vector problem," *STOC*, vol. 2009, pp. 333–432, 2009.

[30] 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.

[31] F. Wang, K. Wang, and B. Li, "An efficient leveled identity-based FHE," *Network and System Security*, vol. 9408, pp. 303–315, 2015.