applied sciences

MDPI

Article

Efficient Multi-Identity Full Homomorphic Encryption Scheme on Lattice

Huifeng Fan, Ruwei Huang * and Fengting Luo

School of Computer and Electronic Information, Guangxi University, Nanning 530004, China; 2013391012@st.gxu.edu.cn (H.F.); 2113391039@st.gxu.edu.cn (F.L.)
* Correspondence: ruweih@gxu.edu.cn

Abstract: Aiming at the problem that the fully homomorphic encryption scheme based on single identity cannot satisfy the homomorphic operation of ciphertext under different identities, as well as the inefficiency of trapdoor function and the complexity of sampling algorithm, an improved lattice MIBFHE scheme was proposed. Firstly, we combined MP12 trapdoor function with dual LWE algorithm to construct a new IBE scheme under the standard model, and prove that the scheme is IND-sID-CPA security under the selective identity. Secondly, we used the eigenvector method to eliminate the evaluation key, and transform the above efficient IBE scheme into a single identity IBFHE scheme to satisfy the homomorphic operation. Finally, we improved the ciphertext extension method of CM15 and constructed a new Link-mask system that supports the transformation of IBFHE scheme under the standard model, and then, converted the above IBFHE scheme into MIBFHE scheme based on this system. The comparative analysis results showed that the efficiency of this scheme is improved compared with similar schemes in the trapdoor generation and preimage sampling, and the dimension of lattice and ciphertext size are significantly shortened.

Keywords: lattice; full homomorphic encryption; multi-identity encryption; LWE problem

check for updates

Citation: Fan, H.; Huang, R.; Luo, F. Efficient Multi-Identity Full Homomorphic Encryption Scheme on Lattice. *Appl. Sci.* **2023**, *13*, 6343. https://doi.org/10.3390/app13106343

Academic Editor: Arcangelo Castiglione

Received: 19 March 2023
Revised: 13 May 2023
Accepted: 15 May 2023
Published: 22 May 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous development of cloud computing, cloud computing faces the security problem of how to ensure data privacy in the process of implementing applications. In 1978, Rivest et al. [1] proposed the idea of homomorphic encryption to protect data security. Homomorphic encryption has special properties that it can perform effective operations on ciphertext without decryption in the phase of processing data ciphertext, which is equivalent to encrypting the plaintext after corresponding operations. Therefore, how to construct a scheme with homomorphic properties became a difficult problem for cryptographers. Until 2009, Gentry [2] proposed the first FHE (full homomorphic encryption) scheme based on ideal lattice. Since then, FHE became a research hotspot in the field of cryptography. Cryptographers proposed a series of FHE schemes based on different theoretical foundations, including integer-based FHE schemes (such as [DGHV10] scheme [3]), RLWE-based (Ring Learning with Errors, RLWE) FHE schemes (such as [BV11a] scheme [4]), LWE-based FHE schemes (such as [BV11b, BGV12] scheme [5,6]) and FHE scheme with eigenvector (such as [GSW13] scheme [7]).

As an important extension of the public key encryption systems, FHE needs to consider the problem of identity authentication in the cloud computing environment. The general method is to introduce public key certificates for authentication. However, the existence of public key certificates brought additional costs to the entire cryptosystem in all aspects such as computing, storage, communication and management. Additionally, the existing FHE systems generally have the problem of large public key size.

In 1984, Shamir [8] first proposed the IBE (identity-based encryption) scheme. Its central idea is to generate a public key from the user's unique identity (such as e-mail

address, mobile phone number, etc.) and public parameters, so that there is no need to issue an additional public key for each user. The user's secret key can be generated by the trusted third party center (Key Generate Center, KGC) using the identity and the system's master secret key. It eliminates the additional overhead associated with public key certificates and can manage keys more efficiently. Therefore, scholars began to study how to combine homomorphic encryption and identity-based encryption to construct the scheme of IBFHE (identity-based full homomorphic encryption), which has the advantages of FHE and IBE at the same time. It can not only perform access control and homomorphic operation on identity ciphertext, but also effectively manage the key. In 2010, Naccache [9] first proposed the open issue of how to construct identity-based full homomorphic encryption scheme at the CRYPTO'2010 conference. In 2013, Gentry et al. [7] constructed the first IBFHE scheme based on the LWE problem with the method of eigenvectors, and also proposed a transformation mechanism that can transform the IBE scheme satisfying the corresponding conditions into the related IBFHE scheme, which solved the above open problem to some extent. However, it is only applicable to single-identity encryption scenarios. It can only perform homomorphic operations on ciphertext encrypted under the same identity, and cannot perform homomorphic operations on ciphertext encrypted based on different identities. However, in many real-world scenarios, homomorphic-encrypted ciphertexts are usually encrypted under different identities.

In 2014, Clear and McGoldrick [10] constructed a multi-identity based full homomorphic encryption (MIBFHE) scheme. However, the construction largely depended on indistinguishable obfuscation [11]. Since it is difficult to realize indistinguishable obfuscation at present, the current efficiency is very low, and the security of the scheme cannot be based on a recognized computational problem. In 2015, Clear and McGoldrick [12] extended the FHE scheme constructed by Gentry et al. [7] to the first MIBFHE scheme based on the standard LWE problem (this scheme is called CM15 scheme), but the process of ciphertext expansion is complex and the noise growth is too fast. In 2019, TU et al. [13] made use of the transformation mechanism of [12] and combined with the hierarchical identity-based encryption scheme proposed by Cash et al. [14] to construct a hierarchical multi-identity full homomorphic encryption scheme. In the same year, Shen et al. [15] proposed a hierarchical multi-identity fully homomorphic encryption scheme based on the multi-key scheme of Mukherjee et al. [16]. In 2020, Pal and Dutta [17] constructed a multi-identity multi-attribute MIBFHE scheme with chosen ciphertext security on the basis of multi-key full homomorphism, but their extension process uses Witness Pseudorandom Function (WPRF), which is a non-standard assumption. In 2021, Shen et al. [18] constructed a compressible multi-key and multi-identity fully homomorphic encryption based on the compressible FHE scheme proposed by Gentry et al. [19]. In 2022, Liu et al. [20] constructed a hierarchical multi-hop MIBFHE scheme based on the IBE scheme proposed by Gentry et al. [21] and the hierarchical multi-hop multi-key FHE scheme proposed by Peikert et al. [22].

The trapdoor generation of the above scheme is quite complex and too inefficient in terms of both operation and output's quality, which is not suitable for practice. It mainly used the trapdoor generation algorithm of [23,24], which involves the calculation of complex HNF (Hermite Normal Forms) and matrix inversion operations. Although the dimension and quality of its output are asymptotically optimal, the hidden constant factor is quite large. In addition, the preimage sampling algorithm of [21] needs to perform high-precision real number orthogonalization iterative operation during the sampling process, resulting in high complexity of the preimage sampling.

In 2012, Micciancio et al. [25] proposed a new trapdoor generation algorithm and corresponding preimage sampling algorithm (this scheme is called MP12 scheme). Compared with the structure of [23,24], it is essentially equivalent to one-time multiplication operation of two random matrices, which does not involve the calculation of complex HNF and matrix inversion operations. Its terms are chosen independently in the appropriate probability distribution, so it is more efficient. At the same time, Micciancio also pointed

out that MP12 trapdoor can be used to optimize all lattice-based IBE schemes, but no specific scheme is given.

Our Contribution. In view of the above problems, in order to make the lattice MIBFHE scheme more practical, solving the problem of inefficient trapdoor generation must be considered. In this paper, we proposed an improved scheme using the transformation mechanism of [12]. First, based on the trapdoor function designed by Micciancio et al. [25] and the IBE scheme of Agrawal et al. [26], we proposed a new IBE scheme under the standard model, and proved that the scheme is IND-sID-CPA security under selective identity. Then, based on the above efficient IBE scheme and the eigenvector method proposed by Gentry et al. [7], which eliminate the evaluation key, the IBE scheme in this paper is transformed into a single-identity IBFHE scheme that satisfies homomorphic operation. Finally, a Link–Mask system was reconstructed based on the ciphertext extension method of [12], and IBFHE was converted into MIBFHE using the reconstructed extended ciphertext method and the masking scheme.

Organization. The second chapter introduces some notation we need to use throughout the paper, and reviews important definitions, including the trapdoor generation algorithm and LWE hardness problem. The third chapter firstly constructs an efficient IBE scheme, and proves the correctness and security of the IBE scheme. The parameter setting of the scheme and the parameter comparison of other schemes are introduced. The fourth chapter introduces how to use the approximate eigenvector to transform the IBE scheme constructed in the third chapter into the IBFHE scheme, and proves the correctness and security of the scheme. The fifth chapter uses the **Link–Mask** algorithm constructed in this paper to transform the IBFHE scheme in the fourth chapter into MIBFHE scheme, and also gives the correctness and security proof of the scheme, as well as the efficiency comparison analysis of the scheme. The sixth chapter is a summary.

2. Preliminaries

Notation. *There are some notations that we will use throughout this paper. We denote* $\mathbb{Z}/q\mathbb{Z}$ *as* $\mathbb{Z}_q$ *and its elements are in the range of* $(-q/2,\ q/2]$. *We use bold uppercase letters (e.g.,* $\boldsymbol{A}, \boldsymbol{B}$*) to represent matrices, and bold lowercase letters (e.g.,* $\boldsymbol{a}, \boldsymbol{b}$*) to represent vectors. All vectors in this paper are default column vectors. For a vector* $\boldsymbol{a} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n) \in \mathbb{Z}_q^n$, $\boldsymbol{a}_i$ *denotes the i-th component scalar. For a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, $\boldsymbol{A}[i, j]$ *denotes the i-th row and the j-th column element of* $\boldsymbol{A}$. *Let denote the Euclidean norm of a vector* $\boldsymbol{a}$ *as* $\|\boldsymbol{a}\| = \sqrt{\sum \boldsymbol{a}_i^2}$ *and* $\boldsymbol{s}_1(\boldsymbol{R})$ *represent the maximum singular value of matrix* $\boldsymbol{R}$. *We denote* $[\boldsymbol{A}|\boldsymbol{B}]$ *as the concatenation of two matrices.*

Let n denote the security parameter. We define $[n] = \{1, 2, \ldots, n\}$ for any positive integer n. Let $negl(n)$ denote a negligible function that grows slower than n^{-c} for any constant $c > 0$ and any sufficiently large value of n. We say that an event happens with overwhelming probability if it happens with probability at least $1 - negl(n)$ for some negligible $negl(\cdot)$. Let $\omega(\cdot)$ denotes the degree of asymptotic when $f(n) = \omega(g(n))$. That is $\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$ for any positive integer $\boldsymbol{c}$ and a positive integer d satisfy $n > d$, $0 \leq \boldsymbol{c} \cdot g(n) < f(n)$.

2.1. Relevant Definitions of Lattice

Definition 1. ***(Lattice)*** *Let* $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m\} \in \mathbb{R}^n$ *be m linearly independent vectors on the n-dimensional Euclidean space* $\mathbb{R}^n$. *Set* $\boldsymbol{B} = \{\boldsymbol{b}_1 | \boldsymbol{b}_2 | \ldots | \boldsymbol{b}_m\} \in \mathbb{R}^{n \times m}$, *and lattice* $\Lambda(\boldsymbol{B})$ *can be expressed linearly by the integer coefficients of all these vectors of* $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m$, *as defined follows:*

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

where the linear independent vector $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_m\}$ which is a basis of the lattice form a lattice space, with dimension n and rank m, for $m > n$. When $n = m$, the $\Lambda(\boldsymbol{B})$ is a full-rank

lattice, the scheme is usually constructed with the full-rank lattice. Here, we are interested in integer lattices, i.e., when $\boldsymbol{s}$ is contained in $\mathbb{Z}^m$.

Definition 2. ***(q-Module Lattice)*** *For $n, m, q \in \mathbb{Z}$, where q is prime, $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ and $\boldsymbol{u} = \mathbb{Z}_q^n$, define:*

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

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

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

where $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$ is the coset of $\Lambda_q^{\perp}(\boldsymbol{A})$. $\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$ is a shift of $\Lambda_q^{\perp}(\boldsymbol{A})$ which satisfies $\boldsymbol{t} + \Lambda_q^{\perp}(\boldsymbol{A}) = \Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$, for $\boldsymbol{t} \in \Lambda_q^{\boldsymbol{u}}(\boldsymbol{A})$.

2.2. Discrete Gaussian Distribution

Definition 3. ***(Gaussian-Shaped Function*** [27]***)*** *For any real number $\sigma > 0$, any vector $\boldsymbol{c} \in \mathbb{R}^n$, and the standard deviation σ, where $\forall \boldsymbol{x} \in \mathbb{R}^n$. Gaussian-shaped function is defined as*

$$\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \exp\left(\frac{-\pi \parallel \boldsymbol{x} - \boldsymbol{c} \parallel^2}{\sigma^2}\right)$$

Definition 4. ***(Discrete Gaussian Distribution*** [27]***)*** *Let lattice $\Lambda \in \mathbb{R}^{n\times m}$, for any real number $\sigma > 0$, any vector $\boldsymbol{c} \in \mathbb{R}^n$, the standard deviation σ, where $\forall \boldsymbol{x} \in \Lambda$. The discrete Gaussian distribution with distribution center $\boldsymbol{c}$ is defined as*

$$\mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}(\boldsymbol{x}) = \frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})}{\rho_{\sigma,\boldsymbol{c}}(\Lambda)} = \frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})}{\sum_{\boldsymbol{v}\in\Lambda}\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{v})}$$

For convenience, we abbreviate $\rho_{\sigma,0}$ and $\mathcal{D}_{\Lambda,\sigma,0}$ as ρ_σ and $\mathcal{D}_{\Lambda,\sigma}$. When $\sigma = 0$, we use ρ to express ρ_1. Distribution $\mathcal{D}_{\Lambda,\sigma,\boldsymbol{c}}$ is usually defined over the lattice $\Lambda = \Lambda_q^{\perp}(\boldsymbol{A})$ for a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ or over a coset $\Lambda = \boldsymbol{t} + \Lambda_q^{\perp}(\boldsymbol{A})$, where $\boldsymbol{t} \in \mathbb{Z}^m$.

2.3. LWE Hardness Problem

The security of all our structures is reduced to the LWE problem, which was first defined by Regev [27] in 2005. It proved to be a non-deterministic polynomial (NP) problem with polynomial complexity.

Definition 5. ***(LWE Hardness Problem*** [27]***)*** *Consider a positive integer n, a prime q, a noise distribution χ over $\mathbb{Z}_q$, and uniformly random secret key $\boldsymbol{s} \in \mathbb{Z}_q^n$. An $(\mathbb{Z}_q, n, \chi) -$ LWE problem include accessing an unspecified challenge oracle $\mathcal{O}$, that is, the oracle can be a noisy pseudo-random sampler $\mathcal{O}_s$ with some constant random secret key $\boldsymbol{s} \in \mathbb{Z}_q^n$, or it can be a truly random sampler $\mathcal{O}_\$$. The behaviors of the two kinds of samplers are as follows.*

$\mathcal{O}_s$: outputs sample of the form $(\boldsymbol{u}_i, \boldsymbol{v}_i) = (\boldsymbol{u}_i, \langle \boldsymbol{u}_i, \boldsymbol{s}\rangle + \boldsymbol{x}_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\boldsymbol{s} \in \mathbb{Z}_q^n$ is a randomly uniform and invariant secret vector, $\boldsymbol{u}_i \in \mathbb{Z}_q^n$ is a randomly uniformly selected vector, and $\boldsymbol{x}_i \in \mathbb{Z}_q$ is fresh sample from χ.

$\mathcal{O}_\$$: outputs truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The $(\mathbb{Z}_q, n, \chi) -$ LWE problem allows repeated queries to the challenge oracle $\mathcal{O}$. For a random $\boldsymbol{s} \in \mathbb{Z}_q^n$, if $\mathrm{LWE} - \boldsymbol{adv}[\mathcal{A}] = \left|\Pr\left[\mathcal{A}^{\mathcal{O}_s} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}_\$} = 1\right]\right.$ is non-negligible, we say that algorithm $\mathcal{A}$ can solve the $(\mathbb{Z}_q, n, \chi) -$ LWE problem, where $\mathrm{LWE} - \boldsymbol{adv}[\mathcal{A}]$.represents the advantage of algorithm $\mathcal{A}$ in solving the $(\mathbb{Z}_q, n, \chi) -$ LWE problem.

Regev [27] showed that for some noise distributions χ, denoted $\overline{\Psi}_\alpha$. The LWE problem is as difficult as the worst-case SIVP and GapSVP under quantum reduction (see also [28]).

Definition 6. ([27]) *Consider a positive integer n, a real parameter $\alpha = \alpha(n) \in (0,1)$, and a prime q. Denote Ψ_α as the normal distribution on $\mathbb{Z}_q$ with the mean 0 as the Gaussian center and the standard deviation $\frac{\alpha}{\sqrt{2\pi}}$, whose corresponding discrete distribution is $\overline{\Psi}_\alpha$.*

Lemma 1. ([27]) *Consider positive integer n, q and $\alpha \in (0,1)$, if there is an efficient, possibly quantum, algorithm to solve the $\left(\mathbb{Z}_q, n, \overline{\Psi}_\alpha\right) -$ LWE problem for $\alpha q > 2\sqrt{n}$, then in the worst case, there is an efficient polynomial quantum algorithm to solve the SIVP and the GapSVP problems with an approximate factor of $\widetilde{O}(n/\alpha)$.*

2.4. Preimage Matrix

Lemma 2. ([25]) *Consider an odd prime q and a positive integer n, m, m'. For any $m \geq n \log q$, there exists a fixed efficiently computable preimage matrix $\boldsymbol{M} \in \mathbb{Z}_q^{n \times m}$ and an efficiently computable deterministic "short preimage" function $\boldsymbol{M}^{-1}(\cdot) : \mathbb{Z}q_{n \times m'}^{\rightarrow} \cdot \mathbb{Z}_q^{m \times m'}$ that satisfies the following conditions. For any m', when matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m'}$ is input, the function $\boldsymbol{M}^{-1}(\boldsymbol{A})$ outputs a bit-matrix $\boldsymbol{M}^{-1}(\boldsymbol{A}) \in \{0,1\}^{m \times m'}$ such that $\boldsymbol{M}\boldsymbol{M}^{-1}(\boldsymbol{A}) = \boldsymbol{A}$.*

We can regard $\boldsymbol{M}$ as a special matrix. For those familiar with GSW13 [7] encryption, multiplication $\boldsymbol{M}$ is the BitDecomp^{-1} operation, and the function $\boldsymbol{M}^{-1}(\cdot)$ is called BitDecomp. Note that $\boldsymbol{M}^{-1}(\cdot)$ itself is not a matrix, but rather an efficiently computable function.

Let $\boldsymbol{x}, \boldsymbol{y}$ be vectors of some dimension n over $\mathbb{Z}_q$. Let $k = \lceil \log q \rceil$ and $w = nk$. Let BitDecomp$(\boldsymbol{x})$ be the w-dimension vector $\boldsymbol{x}' = (\boldsymbol{x}_{1,0}, \ldots, \boldsymbol{x}_{1,k-1}, \ldots, \boldsymbol{x}_{n,0}, \ldots, \boldsymbol{x}_{n,k-1})$, where $\boldsymbol{x}_{i,j}$ is the j-th bit in $\boldsymbol{x}_i$'s binary representation. bits ordered least significant to most significant. Let BitDecomp$^{-1}(\boldsymbol{x}') = \left(\sum 2^j \cdot \boldsymbol{x}_{1,j}, \ldots, \sum 2^j \cdot \boldsymbol{x}_{n,j}\right) = \boldsymbol{x}$ be the inverse of BitDecomp, but well-defined even when the input is not a 0/1 vector. Let Flatten$(\boldsymbol{x}') =$ BitDecomp$\left(\text{BitDecomp}^{-1}(\boldsymbol{x}')\right)$, a w-dimension vector with 0/1 coefficients. BitDecomp$(\boldsymbol{A})$, BitDecomp$^{-1}(\boldsymbol{A})$, or Flatten$(\boldsymbol{A})$ be the matrix formed by applying the operation to each column of $\boldsymbol{A}$ separately. Finally, let Powersof2$(\boldsymbol{y}) = \left(\boldsymbol{y}_1, 2\boldsymbol{y}_1, \ldots, 2^{k-1}\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n, 2\boldsymbol{y}_n, \ldots, 2^{k-1}\boldsymbol{y}_n\right)$. Has the following properties:

(1) $\langle \text{BitDecomp}(\boldsymbol{x}) \rangle, \text{Powersof2}(\boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$.
(2) $\langle \boldsymbol{x}', \text{Powersof2}(\boldsymbol{y}) \rangle = \langle \text{BitDecomp}^{-1}(\boldsymbol{x}'), \boldsymbol{y} \rangle = \langle \text{Flatten}(\boldsymbol{x}'), \text{Powersof2}(\boldsymbol{y}) \rangle$.

2.5. Trapdoor Function and Trapdoor Generation Algorithm

Definition 7. ***(MP12 Trapdoor** [25]) For any integer $n > 1$, $q \geq 2$, $m = O(n \log q)$, $k = \lceil \log q \rceil$, $\overline{m} = m - nk$, $w = nk$, $m \geq w \geq n$. Set matrices $\boldsymbol{A} \in \mathbb{Z}^{n \times m}$ and $\boldsymbol{G} \in \mathbb{Z}_q^{n \times w}$, and the corresponding $\boldsymbol{G}$-trapdoor matrix of $\boldsymbol{A}$ is $\boldsymbol{R} \in \mathbb{Z}^{\overline{m} \times w}$, which satisfies $\boldsymbol{A}\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{I}_{nk} \end{bmatrix} = \boldsymbol{H}\boldsymbol{G}$, where $\boldsymbol{H} \in \mathbb{Z}_q^{n \times n}$ is an invertible matrix, and $\boldsymbol{H}$ is called label of the trapdoor. The trapdoor's quality depends on the maximum singular value $\boldsymbol{s}_1(\boldsymbol{R})$.*

Lemma 3. ***(Trapdoor Generation Algorithm** [25]) For $n \geq 1, q \geq 2, m = O(n \log q) \approx 2n \log q, \overline{m} = m - nk, w = nk, k = \lceil \log q \rceil$, modulus $q = q(n)$, invertible matrix $\boldsymbol{H} \to \mathbb{Z}_q^{n \times n}$, construct a gadget matrix $\boldsymbol{G} = \boldsymbol{I}_n \otimes \boldsymbol{g}^T \in \mathbb{Z}^{n \times nk}$, where $\boldsymbol{g}^T = \left[1, 2^1, 2^2, \ldots, 2^{k-1}\right] \in \mathbb{Z}_q^k$. Randomly choose uniform matrix $\overline{\boldsymbol{A}} \in \mathbb{Z}_q^{n \times \overline{m}}$. There exists a trapdoor generation algorithm $\boldsymbol{TrapGen}(1^n, 1^m, q)$, outputs matrix $\boldsymbol{A} = \left[\overline{\boldsymbol{A}} \middle| \boldsymbol{H}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right] \in \mathbb{Z}_q^{n \times m}$ and its trapdoor matrix $\boldsymbol{R} \in \mathbb{Z}^{\overline{m} \times w}$ where $\boldsymbol{A}$ is statistically indistinguishable from $\mathbb{Z}_q^{n \times m}$ and the size of trapdoor is $\boldsymbol{s}_1(\boldsymbol{R}) \leq \sqrt{m}\omega\left(\sqrt{\log n}\right)$.*

Lemma 4. *(**Sampling Algorithm** [25]) As same as the parameter of Lemma 3, let* $\boldsymbol{u} \in \mathbb{Z}_q^n$ *be an n-dimensional random vector, Gaussian parameter* $\sigma \geq \boldsymbol{s}_1(\boldsymbol{R}) \cdot \omega\left(\sqrt{\log n}\right)$*, and there exists a PPT (probability polynomial time) algorithm* $SampleD(\boldsymbol{A}, \boldsymbol{R}, \boldsymbol{u}, \sigma)$*, output a vector* $\boldsymbol{t} \in \mathbb{Z}_q^m$ *closing to the discrete Gaussian distribution* $\mathcal{D}_{\cdot \frac{u}{q}(\boldsymbol{A}), \sigma\omega(\sqrt{\log n})}$*, satisfying* $\boldsymbol{A} \cdot \boldsymbol{t} = \boldsymbol{u}$ *mod* q*, where* $Pr\left[\boldsymbol{t} \leftarrow \mathcal{D}_{\cdot \frac{u}{q}(\boldsymbol{A}), \sigma\omega(\sqrt{\log n})} : \boldsymbol{t} > \sigma\sqrt{m}\right] \leq negl(n)$.

3. Identity-Based Encryption Scheme

In order to construct a more efficient IBFHE scheme, we first need to construct an IBE scheme with better performance. Next, we improve the IBE scheme of Agrawal et al. [26] based on the MP12 trapdoor generation algorithm and sampling algorithm to make the parameters of the scheme more compact.

3.1. Construction

The basic parameter definition of the scheme: Let n as security parameter, $q = q(n)$ as modulus, $m = O(n \log q)$, randomly uniform matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ and its corresponding trapdoor $\boldsymbol{R} \in \mathbb{Z}^{\overline{m} \times w}$, where $\overline{m} = m - nk, w = nk, k = \lceil \log q \rceil, m' = m + 1$; Construct a gadget matrix $\boldsymbol{G} = \boldsymbol{I}_n \otimes \boldsymbol{g}^{\mathrm{T}} \in \mathbb{Z}^{n \times w}$ for $\boldsymbol{g}^T = \left[1, 2^1, 2^2, \ldots, 2^{k-1}\right] \in \mathbb{Z}_q^k$ and $\boldsymbol{I}_n$ is an $n \times n$ identity matrix; encoding function with FRD (full-rank differences) $\boldsymbol{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$.

- **IBE.Setup**(1^n) : Input the security parameter n and generate the basic parameter $q = q(n), m = O(n \log q)$. Randomly and uniformly choose a matrix $\overline{\boldsymbol{A}} \in \mathbb{Z}_q^{n \times \overline{m}}$ and an n-dimensional vector $\boldsymbol{u} \in \mathbb{Z}_q^n$. Run the trapdoor generation algorithm **TrapGen**$(1^n, 1^m, q)$ to generate matrix $\boldsymbol{A} = \left[\overline{\boldsymbol{A}} \middle| -\overline{\boldsymbol{A}}\boldsymbol{R}\right] \in \mathbb{Z}_q^{n \times m}$ and its trapdoor matrix $\boldsymbol{R} \in \mathbb{Z}^{\overline{m} \times w}$. Output master public key $MPK = (\boldsymbol{A}, \boldsymbol{u})$ and master secret key $MSK = \boldsymbol{R}$.
- **IBE.Extract**(MPK, MSK, id) : Input the master public key MPK, master secret key MSK, and user's identity vector $id \in \mathbb{Z}_q^n$. Using FRD encoding function $\boldsymbol{H} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$, map each user's id to an invertible matrix $\boldsymbol{H}_{id} \in \mathbb{Z}_q^{n \times n}$. Let $\boldsymbol{A}_{id} = \boldsymbol{A} + [0 | \boldsymbol{H}_{id}\boldsymbol{G}] = \left[\overline{\boldsymbol{A}} \middle| \boldsymbol{H}_{id}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right] \in \mathbb{Z}_q^{n \times m}$, run the sampling algorithm **SampleD**$(\boldsymbol{A}_{id}, \boldsymbol{R}, \boldsymbol{u}, \sigma)$ to generate secret key $\boldsymbol{t}_{id} \in \mathbb{Z}_q^m$ corresponding to each user's id, satisfying $\boldsymbol{A}_{id}\boldsymbol{t}_{id} = \boldsymbol{u}$ *mod* q. Set $\boldsymbol{A}'_{id} = \left[\boldsymbol{u} \middle| \boldsymbol{A}_{id}\right] \in \mathbb{Z}_q^{n \times m'}$. Output secret key $\boldsymbol{s}_{id} = (1, -\boldsymbol{t}_{id}) \in \mathbb{Z}_q^{m'}$, satisfying $\boldsymbol{A}'_{id}\boldsymbol{s}_{id} = 0$ *mod* q.
- **IBE.Enc**$(MPK, id, \boldsymbol{b})$: Input the master public key MPK, user's identity vector $id \in \mathbb{Z}_q^n$ and encrypted plaintext message $\boldsymbol{b} \in \{0, 1\}$. Let $\mu = \left(\boldsymbol{b}\frac{q}{2}, 0, \ldots, 0\right) \in \mathbb{Z}_q^{m'}$. Randomly choose a uniform vector $\boldsymbol{y} \xleftarrow{\$} \{0, 1\}^n$ and an error vector $\boldsymbol{e} \xleftarrow{\$} \chi_{\frac{m'}{\Psi_\alpha}}$ according to the LWE error distribution. Output ciphertext vector $\boldsymbol{c}_{id} = \boldsymbol{A}'^{\mathrm{T}}_{id}\boldsymbol{y} + \mu + \boldsymbol{e} \in \mathbb{Z}_q^{m'}$.
- **IBE.Dec**$(\boldsymbol{MPK}, \boldsymbol{s}_{id}, \boldsymbol{c}_{id})$: Input the master public key $\boldsymbol{MPK}$, user's secret key $\boldsymbol{s}_{id}$ and ciphertext $\boldsymbol{c}_{id}$ to decrypt. Compute $\boldsymbol{b}' = \boldsymbol{s}_{id}^{\mathrm{T}} \cdot \boldsymbol{c}_{id} \in \mathbb{Z}_q$. If $\|\boldsymbol{b}' - \left\lfloor \frac{q}{2} \right\rfloor\| < \left\lfloor \frac{q}{4} \right\rfloor$, output $\boldsymbol{b} = 1$; If $\|\boldsymbol{b}'\| < \left\lfloor \frac{q}{4} \right\rfloor$, output $\boldsymbol{b} = 0$.

3.2. Correctness and Parameters

Theorem 1. ([21]) *When* $m \geq 2n\log q$, $\alpha \leq \left(\sigma\sqrt{m+1} \cdot \omega\left(\sqrt{\log n}\right)\right)^{-1}$, $q \geq 5\sigma(m+1)$*, the IBE scheme constructed in* Section 3.1 *is successfully decrypted with great probability.*

Proof. It can be obtained from the decryption formula

$$\begin{aligned} \boldsymbol{s}_{id}^{\mathrm{T}} \cdot \boldsymbol{c}_{id} &= \boldsymbol{s}_{id}^{\mathrm{T}}\left(\boldsymbol{A}'^{T}_{id}\boldsymbol{y} + \mu + \boldsymbol{e}\right) \\ &= \boldsymbol{s}_{id}^{T}\boldsymbol{A}'^{T}_{id}\boldsymbol{y} + \left\langle \boldsymbol{s}_{id}^{T}, \mu \right\rangle + \left\langle \boldsymbol{s}_{id}^{T}, \boldsymbol{e} \right\rangle \\ &= \boldsymbol{b}\left\lfloor \frac{q}{2} \right\rfloor + \boldsymbol{s}_{id}^{T}, \boldsymbol{e} \end{aligned}$$

According to [21], when $\alpha \leq \left(\sigma\sqrt{m+1}\cdot\omega\left(\sqrt{\log n}\right)\right)^{-1}$, $q \geq 5\sigma(m+1)$ can satisfy $\left\langle \boldsymbol{s}_{id}^T, \boldsymbol{e}\right\rangle \leq \frac{q}{5}$ with a great probability. Due to $\left\langle \boldsymbol{s}_{id}^T, \boldsymbol{e}\right\rangle \leq \frac{q}{5} < \frac{q}{4}$, when $\left\langle \boldsymbol{s}_{id}^T, \boldsymbol{e}\right\rangle < \frac{q}{4}$, if $\boldsymbol{b} = 1$, then $\left\|\left\langle \boldsymbol{s}_{id}^T, \boldsymbol{c}_{id}\right\rangle - \frac{q}{2}\right\| < \frac{q}{4}$; If $\boldsymbol{b} = 0$, then $\left\|\left\langle \boldsymbol{s}_{id}^T, \boldsymbol{c}_{id}\right\rangle\right\| < \frac{q}{4}$, obviously the decryption algorithm can successfully decrypt with great probability.

According to the above analysis and Lemma 1, when α and q reach the extreme value, respectively, there is $\alpha\cdot q = \frac{5\sqrt{m+1}}{\omega\left(\sqrt{\log n}\right)} > \frac{5\sqrt{2n\log q}}{\omega\left(\sqrt{\log n}\right)} > 2\sqrt{n}$, satisfying the security requirements of LWE problem, that is $\alpha q > 2\sqrt{n}$. To meet the above requirements, set scheme parameter (m, q, σ, α): $m = 2n\log q$, $q = m^{\frac{3}{2}}\sqrt{n}\omega(\log n)$, $\sigma = \sqrt{m}\omega(\log n)$, $\alpha < \left(m\omega\left(\log^2 n\right)\right)^{-1}$.□

3.3. Security Reduction

Theorem 2. *When $m \geq 2n\log q$, if the $\left(\mathbb{Z}_q, n, \overline{\Psi}_\alpha\right) - LWE$ hardness assumption holds, the basic IBE scheme given in this section is IND-sID-CPA (Indistinguishable from Random, Select-Identity, Chosen-Plaintext Attachment) security.*

Proof. For the IBE scheme proposed in this paper, we use a series of IND-sID-CPA security games proposed by Agrawal et al. [26] under the standard model to prove the security. The security model is established by a sequence game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.The steps are as follows:

Game0 Game0 is a standard original IND-sID-CPA game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.

Game1 Let id^* be the identity of adversary $\mathcal{A}$ who plans to attack. Compared with Game0, the challenger changes the way to generate matrix $\boldsymbol{A}$, and randomly generates $\boldsymbol{A} = \left[\overline{\boldsymbol{A}}\middle|-\boldsymbol{H}_{id^*}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right]$. From lemma 3, we can see that **GenTrap** algorithm in Game0 generates matrix $\boldsymbol{A} = \left[\overline{\boldsymbol{A}}\middle|\boldsymbol{H}_{id}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right]$. From the Left over Hash lemma [29], distribution $\left(\overline{\boldsymbol{A}}, \overline{\boldsymbol{A}}\boldsymbol{R}\right)$ and distribution $\left(\overline{\boldsymbol{A}}, \boldsymbol{B}\right)$ are statistically indistinguishable, for $\boldsymbol{B} \in \mathbb{Z}_q^{n\times w}$. Therefore, in the view of adversary $\mathcal{A}$, the matrix in Game0 and in Game1 are statistically indistinguishable, and adversary $\mathcal{A}$ cannot distinguish Game0 and Game1 with negligible advantages.

Game2 The difference between Game2 and Game1 is that Challenger $\mathcal{C}$ changes the corresponding way to query $id \neq id^*$ secret key. Game2 uses **GenTrap** algorithm to generate matrix $\boldsymbol{G}$ and lattice $\Lambda_q^{\perp}(\boldsymbol{G})$ trapdoor matrix $\boldsymbol{R}_{\boldsymbol{G}}$. Keeping the form of $\boldsymbol{A} = \left[\overline{\boldsymbol{A}}\middle|-\boldsymbol{H}_{id^*}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right]$ in Game1. According to the definition of FRD encoding function, $(\boldsymbol{H}_{id} - \boldsymbol{H}_{id^*})$ is nonsingular. Challenger $\mathcal{C}$ can respond to the secret key query of adversary $\mathcal{A}$ through the trapdoor matrix $\boldsymbol{R}_{\boldsymbol{G}}$ to sample the preimage. Run sampling algorithm $\boldsymbol{t}_{id} \leftarrow \mathbf{SampleD}(\boldsymbol{A}_{id}, \boldsymbol{R}_{\boldsymbol{G}}, \boldsymbol{u}, \sigma)$ and output secret key $\boldsymbol{s}_{id} = (1, -\boldsymbol{t}_{id})$ to adversary $\mathcal{A}$. If $id = id^*$, then $(\boldsymbol{H}_{id} - \boldsymbol{H}_{id^*})$ is a singular matrix, and the game ends. The distribution $\mathcal{D}_{\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}_{id}), \sigma\omega\left(\sqrt{\log n}\right)}$ of $\boldsymbol{s}_{id}$ in Game2 and $\boldsymbol{s}_{id}$ in Game1 are statistically indistinguishable, so adversary $\mathcal{A}$ cannot distinguish Game1 and Game2 with negligible advantages.

Game3 The difference between Game3 and Game2 is that the challenge ciphertext is always selected as a random independent element of $\mathbb{Z}_q^{m'}$ in the ciphertext space, so the advantage of adversary $\mathcal{A}$ is zero.

For PPT adversary $\mathcal{A}$, it is still necessary to prove that the adversary cannot distinguish Game2 and Game3 in computation through the hardness of the LWE problem. Assuming adversary $\mathcal{A}$ has non-negligible advantage in distinguishing Game2 and Game3, we use adversary $\mathcal{A}$ to construct an LWE algorithm $\mathcal{E}$. Recall from definition 5 that an LWE problem instance is provided as a sampling $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or noise pseudo-random $\mathcal{O}_s$. Challenger $\mathcal{C}$ uses the adversary $\mathcal{A}$ to distinguish the two. The steps are as follows:

Instance Challenger $\mathcal{C}$ requests from $\mathcal{O}$ and receives $\overline{m}+1$ samples $(\boldsymbol{u}_i, \boldsymbol{v}_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, for $i = 0, 1, \ldots, \overline{m}+1$.

Target Adversary $\mathcal{A}$ declares to challenger $\mathcal{C}$ the target identity of the planned attack id^*.

Setup Challenger $\mathcal{C}$ sets MPK according to the target identity id^*.

(1) Challenger $\mathcal{C}$ uses the known samples to construct matrix $\overline{\boldsymbol{A}} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\overline{m}}) \in \mathbb{Z}_q^{n \times \overline{m}}$.
(2) Take $\boldsymbol{u}_0$ as a common random vector $\boldsymbol{u} = \boldsymbol{u}_0 \in \mathbb{Z}_q^n$.
(3) Select $\boldsymbol{R} \leftarrow \mathcal{D}^{\overline{m} \times w}$ from the distribution $\mathcal{D}$ and construct the matrix $\boldsymbol{A}_1 = -\boldsymbol{H}_{id^*}\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}$.
(4) Send the common parameter $\{\boldsymbol{u}, \overline{\boldsymbol{A}}, \boldsymbol{A}_1\}$ to the adversary $\mathcal{A}$.

From Left over Hash lemma [29], for an adversary $\mathcal{A}$, matrix $\boldsymbol{A}_1$ is uniformly indistinguishable.

Queries1 Similar to Game2, Challenger $\mathcal{C}$ responds to each secret key query of adversary $\mathcal{A}$.

Challenge The adversary submits challenge plaintext $\boldsymbol{b}^* \in \{0,1\}$ to challenger $\mathcal{C}$, and challenger $\mathcal{C}$ outputs challenge ciphertext $\boldsymbol{c}_{id}^*$ for target identity id^*:

(1) Let $\boldsymbol{v}^* = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{\overline{m}})^{\mathrm{T}} \in \mathbb{Z}_q^{\overline{m}}$.
(2) Hide plaintext message $\boldsymbol{b}^*$ through constructing $\boldsymbol{c}_0^* = \boldsymbol{v}_0 + \boldsymbol{b}^* \frac{q}{2}$.
(3) Let $\boldsymbol{c}_1^* = \begin{bmatrix} \boldsymbol{v}^* \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{v}^* + \boldsymbol{e}' \end{bmatrix} \in \mathbb{Z}_q^m$,for $\boldsymbol{e}' \xleftarrow{\Psi_\alpha} \mathbb{Z}_q^w$.
(4) Select random bit $r \xleftarrow{\$} \{0,1\}$. When $r = 0$, send $\boldsymbol{c}^* = (\boldsymbol{c}_0^*, \boldsymbol{c}_1^*)$ to the adversary $\mathcal{A}$; when $r = 1$, randomly and uniformly select $\boldsymbol{c}_{id} \in \mathbb{Z}_q^{m'}$ to pass to the adversary $\mathcal{A}$.

Attention: When $\mathcal{O} = \mathcal{O}_s$, the distribution of $\boldsymbol{c}^*$ is indistinguishable from the challenge ciphertext in Game2. From the definition of the LWE problem, we can know that $\boldsymbol{v}^* = \overline{\boldsymbol{A}}^{\mathrm{T}}\boldsymbol{s} + \boldsymbol{e}$ $\boldsymbol{A}_{id^*} = \left[\overline{\boldsymbol{A}} \middle| (\boldsymbol{H}_{id^*} - \boldsymbol{H}_{id^*})\boldsymbol{G} - \overline{\boldsymbol{A}}\boldsymbol{R}\right] = \left[\overline{\boldsymbol{A}} \middle| -\overline{\boldsymbol{A}}\boldsymbol{R}\right]$.Thus

$$\boldsymbol{c}_1^* = \begin{bmatrix} \boldsymbol{v}^* \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{v}^* + \boldsymbol{e}' \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{A}}^{\mathrm{T}}\boldsymbol{s} + \boldsymbol{e} \\ -\boldsymbol{R}^{\mathrm{T}}\left(\overline{\boldsymbol{A}}^{\mathrm{T}}\boldsymbol{s} + \boldsymbol{e}\right) + \boldsymbol{e}' \end{bmatrix} = \boldsymbol{A}_{id^*}^{\mathrm{T}}\boldsymbol{s} + \begin{bmatrix} \boldsymbol{e} \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{e} + \boldsymbol{e}' \end{bmatrix}$$

The right side of the equation is the challenge ciphertext $\boldsymbol{c}_1$ in Game 2; $\boldsymbol{c}_0^* = \boldsymbol{u}_0^T\boldsymbol{s} + \boldsymbol{e}'' + \boldsymbol{b}^* \left| \frac{q}{2} \right|$ is the challenge ciphertext $\boldsymbol{c}_0$ in Game2, for $\boldsymbol{e}'' \xleftarrow{\Psi_\alpha} \mathbb{Z}_q$. Thus $\boldsymbol{c}^*$ is a valid ciphertext of $\boldsymbol{b}^*$ corresponding to identity id^*.

When $\mathcal{O} = \mathcal{O}_\$$, $\boldsymbol{v}_0 \in \mathbb{Z}_q$ and $\boldsymbol{v}^* \in \mathbb{Z}_q^{\overline{m}}$ are then uniformly selected. According to the Left over Hash lemma [29], $-\boldsymbol{R}^{\mathrm{T}}\boldsymbol{v}^*$ obeys the discrete random distribution, so the $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{v}^* + \boldsymbol{e}'$ also obeys the discrete random distribution. Therefore, the distribution of challenge ciphertext $\boldsymbol{c}^*$ is indistinguishable from Game3, and is randomly selected by the challenger $\mathcal{C}$ from $\mathbb{Z}_q^{m'}$. **Queries2** The adversary $\mathcal{A}$ can continue to query the secret key in the same way as **Queries1.**

Guess The adversary $\mathcal{A}$ distinguishes whether the ciphertext is a random independent vector on $\mathbb{Z}_q^{m'}$ or a valid ciphertext of plaintext message $\boldsymbol{b}^*$, and the challenger $\mathcal{C}$ answers whether the samples in the LWE problem are from $\mathcal{O}_s$ or $\mathcal{O}_\$$ according to the guess results.

In summary, when $\mathcal{O} = \mathcal{O}_s$, the view of the adversary $\mathcal{A}$ is the same as Game2; when $\mathcal{O} = \mathcal{O}_\$$, adversary $\mathcal{A}$ has the same view as Game3. Because the advantage of algorithm $\mathcal{E}$ in solving the LWE problem is the same as that of adversary $\mathcal{A}$ in distinguishing Game2 and Game3, and because there is no PPT algorithm that can effectively solve the LWE problem; thus, the scheme is IND-sID-CPA secure, and the proof is over. □

3.4. Efficiency Analysis of IBE Scheme

We compared the parameters of the proposed IBE scheme with the ABB-IBE scheme proposed by Agrawal et al. [26] with the same security as this scheme. See Table 1 for comparison results.

Table 1. Comparison of main parameters of IBE scheme.

Scheme	Dimension	Ciphertext	Public Key	Secret Key	Gaussian Parameter
[26]	$6n \log q$	$2m+1$	$n \times (3m+1)$	$m \times m$	$m\omega(\log n)$
Ours-IBE	$2n \log q$	$m+1$	$n \times (m+1)$	$(m \times m)/4$	$\sqrt{m}\omega(\log n)$

From the analysis in Table 1, it can be seen that the main efficiency parameters of the IBE scheme in this paper were significantly optimized. Compared with the ABB10-IBE trapdoor generation algorithm based on [23], this scheme uses the MP12 trapdoor generation algorithm to reduce the lattice security dimension from $6n \log q$ to $2n \log q$, and the size of the master secret keys is selected from a short vector in a reasonable Gaussian distribution, so the scale of the public parameters, key size, and ciphertext size of this scheme are reduced.

4. Identity-Based Full Homomorphic Encryption Scheme

Based on the efficient IBE scheme proposed above, a new identity-based fully homomorphic encryption scheme was further constructed. We used the gadget matrix to replace Powersoft2, BitDecomp and Flatten to obtain new encryption and decryption forms. At the same time, we use the "approximate eigenvector" technology to eliminate the evaluation key in homomorphic encryption to obtain a more concise identity-based full homomorphic encryption scheme.

4.1. Construction

The basic parameter definition of the scheme: Let n as security parameter, $\boldsymbol{L}$ represents the maximum depth of homomorphic calculation allowed for the circuit, $q = q(n, \boldsymbol{L})$ is a sufficiently large prime, and $m, m', \overline{m}, w, k$ and FRD encoding function $\boldsymbol{H}$ are the same as the definitions in the above IBE encryption scheme. Define $N = (m+1)k$. We construct another gadget matrix $\boldsymbol{M} = \boldsymbol{I}_{m'} \otimes \boldsymbol{g}^{\mathrm{T}} \in \mathbb{Z}^{m' \times N}$, where $\boldsymbol{g}^T = \left[1, 2^1, 2^2, \ldots, 2^{k-1}\right] \in \mathbb{Z}_q^k$ and $\boldsymbol{I}_{m'}$ is a $m' \times m'$ identity matrix. According to Lemma 2, for any matrix $\boldsymbol{A} \in \mathbb{Z}_q^{m' \times N}$, there exists a function $\boldsymbol{M}^{-1}(\cdot)$ such that $\boldsymbol{M}^{-1}(\boldsymbol{A}) \in \{0,1\}^{N \times N}$, satisfying $\boldsymbol{M}\boldsymbol{M}^{-1}(\boldsymbol{A}) = \boldsymbol{A}$.

- **IBFHE.Setup**$\left(1^n, 1^{\boldsymbol{L}}\right)$: Input the security parameter n and the maximum depth $\boldsymbol{L}$ that the circuit allows homomorphic operations. Run the **IBE.Setup** algorithm to generate matrix $\boldsymbol{A} = \left[\overline{\boldsymbol{A}} \middle| -\overline{\boldsymbol{A}}\boldsymbol{R}\right] \in \mathbb{Z}_q^{n \times m}$. Output the master public key $\boldsymbol{MPK} = (\boldsymbol{A}, \boldsymbol{u})$ and the master secret key $\boldsymbol{MSK} = \boldsymbol{R}$.
- **IBFHE.Extract**$(\boldsymbol{MPK}, \boldsymbol{MSK}, id)$: Input the master public key $\boldsymbol{MPK}$, master secret key $\boldsymbol{MSK}$, and user's identity vector $id \in \mathbb{Z}_q^n$. Run the **IBE.Extract** algorithm to generate matrix $\boldsymbol{A}_{id}^{'} = \left[\boldsymbol{u} \middle| \boldsymbol{A}_{id}\right] \in \mathbb{Z}_q^{n \times m'}$. Output secret key $\boldsymbol{s}_{id} = (1, -\boldsymbol{t}_{id}) \in \mathbb{Z}_q^{m'}$, satisfying $\boldsymbol{A}_{id}^{'}\boldsymbol{s}_{id} = 0 \ mod \ q$.
- **IBFHE.Enc**$(\boldsymbol{MPK}, id, \mu)$: Input the master public key $\boldsymbol{MPK}$, user's identity vector $id \in \mathbb{Z}_q^n$ and encrypted plaintext message $\mu \in \{0,1\}$. Randomly choose uniform vectors $\boldsymbol{y}_i \xleftarrow{\$} \{0,1\}^n$ and error vectors $\boldsymbol{e}_i \xleftarrow{\$} \chi_{\frac{\Psi}{\alpha}}^{m'}$ according to the LWE error distribution. N vectors $\boldsymbol{y}_i$ are connected to form the matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N] \in \mathbb{Z}^{n \times N}$, and N vectors $\boldsymbol{e}_i$ are connected to form the matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N] \in \mathbb{Z}_q^{m' \times N}$, where $i \in [N]$. Output the ciphertext matrix $\boldsymbol{C} = \boldsymbol{A}_{id}^{'\mathrm{T}}\boldsymbol{Y} + \mu\boldsymbol{M} + \boldsymbol{E} \in \mathbb{Z}_q^{m' \times N}$.
- **IBFHE.Eval**$(\boldsymbol{MPK}, f, (\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t))$: Input the master public key $\boldsymbol{MPK}$, Boolean circuit f, and ciphertext $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t$ which are the different ciphertext of the same id with secret key $\boldsymbol{s}_{id}$. Output the operation ciphertext $\boldsymbol{C} = f(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_t)$, where the homomorphic addition is $\boldsymbol{C}_{Add} = \boldsymbol{C}_1 + \boldsymbol{C}_2$ and the homomorphic multiplication is $\boldsymbol{C}_{\boldsymbol{Mult}} = \boldsymbol{C}_1\boldsymbol{M}^{-1}(\boldsymbol{C}_2)$. According to the definitions of addition and multiplication, the homomorphic NAND gate operation is defined as $\boldsymbol{C}_{NAND} = \boldsymbol{M} - \boldsymbol{C}_1\boldsymbol{M}^{-1}(\boldsymbol{C}_2)$.

- **IBFHE.Dec**(MPK, s_{id}, C) : Input the master public key MPK, user's secret key s_{id} and ciphertext C to decrypt. Set the vector $\omega = \left[\left\lfloor\frac{q}{2}\right\rfloor, 0, \ldots, 0\right] \in \mathbb{Z}_q^{m'}$.Compute $\mu' = s_{id}^{\mathrm{T}} \cdot C \cdot M^{-1}(\omega)$ and output $\left\lfloor\frac{2\mu'}{q}\right\rfloor$.

4.2. Correctness and Parameters

Theorem 3. *When* $q = \sqrt{Nn}m^{\frac{3}{2}}\omega(\log n)$,$\sigma = \sqrt{m}\omega(\log n)$,$\alpha < \left(\sqrt{N}m\omega\left(log^2 n\right)\right)^{-1}$, *the IBFHE scheme constructed in* Section 4.1 *is successfully decrypted with great probability.*

Proof. For the initial ciphertext $C \in \mathbb{Z}_q^{m' \times N}$ and secret key $s_{id} \in \mathbb{Z}_q^{m'}$ of the id, there are

$$\begin{aligned} s_{id}^{\mathrm{T}} \cdot C &= s_{id}^{\mathrm{T}}\left(A_{id}^{'\mathrm{T}} Y + \mu M + E\right) \\ &= s_{id}^{\mathrm{T}} A_{id}^{'\mathrm{T}} Y + \mu s_{id}^{\mathrm{T}} M + s_{id}^{\mathrm{T}} E \\ &= \mu s_{id}^{\mathrm{T}} M + s_{id}^{\mathrm{T}} E \\ &= \mu s_{id}^{\mathrm{T}} M + e' \end{aligned} \tag{1}$$

It can be obtained from Equation (1) and decryption formula

$$\begin{aligned} \mu' &= s_{id}^{\mathrm{T}} \cdot C \cdot M^{-1}(\omega) \\ &= \left(\mu s_{id}^{\mathrm{T}} M + e'\right) M^{-1}(\omega) \\ &= \mu s_{id}^{\mathrm{T}} \omega + e' M^{-1}(\omega) \\ &= \mu \left\lfloor\frac{q}{2}\right\rfloor + E' \end{aligned}$$

In order to enable the decryption effective, it is necessary to ensure the ciphertext's noise $\| E' \|_\infty \leq \sqrt{N}\left(q\sigma\sqrt{m}\alpha\omega\left(\sqrt{\log n}\right)\right) < \frac{q}{5}$, where $\alpha < \left(\sqrt{N}\sigma\sqrt{m+1}\omega\left(\sqrt{\log n}\right)\right)^{-1}$, that is $\frac{2\mu'}{q} = \frac{2\mu\left\lfloor\frac{q}{2}\right\rfloor + 2\|E'\|}{q} < \mu + \frac{2}{5}$, satisfying $\frac{2\lfloor\mu'\rfloor}{q} = \mu$. The ciphertext can be successfully decrypted. To meet the above requirements, set scheme parameters (m, q, σ, α): $m = 2n\log q$, $q = \sqrt{Nn}m^{\frac{3}{2}}\omega(\log n)$, $\sigma = \sqrt{m}\omega(\log n)$, $\alpha < \left(\sqrt{N}m\omega\left(\log^2 n\right)\right)^{-1}$. □

4.3. Homomorphic Property

Definition 8. *Let* $C \in \mathbb{Z}_q^{m' \times N}$ *be the ciphertext matrix corresponding to plaintext* μ *of the identity id, and the secret key is* $s \in \mathbb{Z}_q^{m'}$. *If* $s^T C = \mu s^T M + e$ *where* $\|e\|_\infty \leq \beta$, C *is called the* β*-noise ciphertext of plaintext* μ.

Proof. Let C_1 and C_2 be the ciphertexts of identity *id* corresponding to plaintexts μ_1 and μ_2 respectively, namely $s^T C_1 = \mu_1 s^T M + e_1$, $s^T C_2 = \mu_2 s^T M + e_2$, where $\mu_1, \mu_2 \in \{0, 1\}$, $\|e_1\|_\infty \leq \beta_1$, $\|e_2\|_\infty \leq \beta_2$.

(1) Homomorphic addition: $C_{Add} = C_1 + C_2$, satisfy $s^{\mathrm{T}} C_{Add} = s^{\mathrm{T}}(C_1 + C_2) = (u_1 + \mu_2) s^{\mathrm{T}} M + e^+$, where $e^+ = e_1 + e_2$. Obviously C_{Add} is $\beta_1 + \beta_2$ noise ciphertext, that is, after one-time homomorphic addition, the error increases by two times the factor.

(2) Homomorphic multiplication: $C_{Mult} = C_1 M^{-1}(C_2)$, satisfy $s^{\mathrm{T}} C_{Mult} = s^{\mathrm{T}} C_1 M^{-1}(C_2) = (\mu_1 \mu_2) s^{\mathrm{T}} M + e^{\times}$, where $e^{\times} = e_1 M^{-1}(C_2) + \mu_1 e_2$. Obviously $\| e^{\times} \|_\infty \leq \left(\sqrt{N}\beta_1 + \beta_2\right)$, C_{Mult} is $\left(\sqrt{N}\beta_1 + \beta_2\right)$ noise ciphertext. The same calculation is also applicable to NAND gates. □

4.4. Security Reduction

Theorem 4. *If the* $\left(\mathbb{Z}_q, n, \overline{\Psi}_\alpha\right) - LWE$ *hardness assumption holds, the IBFHE scheme given in this section is IND-sID-CPA secure.*

Proof. The security of the IBFHE scheme proposed in this section can be proved based on the IBE scheme constructed in the previous section, because the homomorphic **IBFHE**.**Eval** algorithm in the IBFHE scheme is public and has no effect on the security of the scheme. Under the LWE assumption, let $\boldsymbol{C} = \boldsymbol{A}_{id}^{'T}\boldsymbol{Y} + \mu\boldsymbol{M} + \boldsymbol{E} \in \mathbb{Z}_q^{m' \times N}$ be the ciphertext obtained by encrypting the plaintext message 0 in the IBFHE scheme, which can be regarded as the concatenation of the N ciphertexts of a bit 0 in the IBE scheme. It can be seen from theorem 2 that $\boldsymbol{C}$ and any random uniform matrices in $\mathbb{Z}_q^{m' \times N}$ are indistinguishable. Therefore, according to the definition of the IND-sID-CPA security model, the IBFHE scheme proposed in this section is IND-sID-CPA security. □

5. Multi-Identity Based Full Homomorphic Encryption Scheme

5.1. Link-Mask Scheme

Based on the above IBFHE scheme, we constructed an efficient multi-identity fully homomorphic encryption scheme by using the extended ciphertext method and the masking scheme, which is denoted as mIBFHE.

Firstly, we introduce the general method of converting single identity IBFHE scheme into multi identity scheme. For the convenience of description, we describe our scheme as a simple example. Assuming that there are two parties $(D = 2)$, any polynomial number of parties D can be extended by this method.

Let $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ be the ciphertexts of the plaintext messages μ_1 and μ_2 corresponding to the parties' identities id_1 and id_2 in the IBFHE scheme, respectively, and the identities id_1 and id_2 correspond to the secret keys $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, respectively, which satisfy $\boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{C}_1 = \mu_1\boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{M} + \boldsymbol{e}_1$, $\boldsymbol{s}_2^{\mathrm{T}}\boldsymbol{C}_2 = \mu_2\boldsymbol{s}_2^{\mathrm{T}}\boldsymbol{M} + \boldsymbol{e}_2$. By extending ciphertext $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \mathbb{Z}_q^{m' \times N}$ according to the number of parties D to "extended" ciphertext $\hat{\boldsymbol{C}}_1, \hat{\boldsymbol{C}}_2 \in \mathbb{Z}_q^{2m' \times 2N}$, those satisfy

$$\left[\boldsymbol{s}_1^{\mathrm{T}}, \boldsymbol{s}_2^{\mathrm{T}}\right]\hat{\boldsymbol{C}}_1 = \mu_1\left[\boldsymbol{s}_1^{\mathrm{T}}, \boldsymbol{s}_2^{\mathrm{T}}\right]\begin{bmatrix}\boldsymbol{M} & 0\\ 0 & \boldsymbol{M}\end{bmatrix} + small\ error$$

$$\left[\boldsymbol{s}_1^{\mathrm{T}}, \boldsymbol{s}_2^{\mathrm{T}}\right]\hat{\boldsymbol{C}}_2 = \mu_2\left[\boldsymbol{s}_1^{\mathrm{T}}, \boldsymbol{s}_2^{\mathrm{T}}\right]\begin{bmatrix}\boldsymbol{M} & 0\\ 0 & \boldsymbol{M}\end{bmatrix} + small\ error$$

In this paper, the general method of converting single-identity IBFHE scheme into multi-identity mIBFHE scheme is to convert the encrypted ciphertext matrix under single identity into a $Dm' \times DN$-dimensional general extended matrix, and the scale of extended ciphertext is expanded by D^2. In this way, ciphertexts $\hat{\boldsymbol{C}}_1$ and $\hat{\boldsymbol{C}}$ corresponding to different identities id can be input into the same Boolean circuit $f \in \mathbb{C}$ for homomorphic operation.

In order to perform the above ciphertext expansion, we need to construct a masking scheme: this scheme allows each party (D_1, D_2) to independently generate key pairs, which are $(\boldsymbol{s}_1, pk_1)$, $(\boldsymbol{s}_2, pk_2)$ respectively. D_1 Run the **IBFHE**.**Enc** algorithm to encrypt plaintext message μ_1 under pk_1, and then use pk_2 and its own randomness to extend its ciphertext. In the ciphertext expansion step, D_1 runs the masking algorithm twice (the number of parties) to use pk_1, pk_2 to create matrices $X_1^j, \overline{X}_1^2$, where $j \in [2]$, $\boldsymbol{s}_1^{\mathrm{T}}X_1^1 \approx 0, \boldsymbol{s}_2^{\mathrm{T}}\left(\boldsymbol{C}_1 - X_1^2\right) \approx \mu_1\boldsymbol{M}$, and $\boldsymbol{s}_2^{\mathrm{T}}\overline{X}_1^2 \approx 0$. Then, we randomly chose a matrix $\boldsymbol{Q}$ and set a matrix $\boldsymbol{Q}_2$ such that $\boldsymbol{s}_2^{\mathrm{T}}\boldsymbol{Q}_2 \approx \boldsymbol{s}_1^{\mathrm{T}}\boldsymbol{Q}$. Therefore, the final multi-identity extended ciphertext form of D_1 is

$$\hat{\boldsymbol{C}}_1 = \begin{bmatrix}\boldsymbol{C}_1^1 & \boldsymbol{Q}\\ 0 & \boldsymbol{C}_1^2\end{bmatrix}$$

where $\boldsymbol{C}_1$ is a single identity IBFHE ciphertext of D_1, $\boldsymbol{C}_1^1 = \boldsymbol{C}_1 - X_1^1$ and $\boldsymbol{C}_1^2 = \boldsymbol{C}_1 - X_1^2 + \overline{X}_1^2 - \boldsymbol{Q}_2$. There is

$$\begin{aligned}[s_1^{\mathrm{T}}, s_2^{\mathrm{T}}]\hat{C}_1 &= [s_1^{\mathrm{T}}, s_2^{\mathrm{T}}]\begin{bmatrix} C_1^1 & Q \\ 0 & C_1^2 \end{bmatrix} \\ &= \left[s_1^{\mathrm{T}}C_1^1, s_1^{\mathrm{T}}Q + s_2^{\mathrm{T}}C_1^2\right] \approx \mu_1[s_1^{\mathrm{T}}, s_2^{\mathrm{T}}]\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}\end{aligned}$$

Similarly, the ciphertext C_2 is extended to $\hat{C}_2$, which can perform homomorphic operations on ciphertext $\hat{C}_1$ and $\hat{C}_2$ encrypted under different identities.

Before constructing a specific masking scheme, we need to reconstruct the ciphertext extension of CM15 on the basis of [30]. The operation is as follows.

Link–Mask. Let $Y \in \{0,1\}^{n\times N}$ be a 0-1 matrix, and $V^{(x,t)}$ be a IBFHE ciphertext of $Y[x,t]$ (x-th row and t-th column of Y, $x \in [n], t \in [N]$) under $(pk, sk) = (A, s)$. Let $(pk', sk') = (A', s')$ be another IBFHE key pair. There exists a polynomial-time deterministic algorithm $\mathbf{Link–Mask}\left(pk', \left(V^{1,1}, \ldots, V^{n,N}\right)\right)$, input pk' and encryptions $V^{(x,t)}$, return a matrix $X \in \mathbb{Z}_q^{m'\times N}$, satisfying $s^{\mathrm{T}}X = s^{\mathrm{T}}A'^{\mathrm{T}}Y + e$, where $\|e\|_\infty \leq n(m+1)^2\beta$. The algorithm is as follows (Algorithm 1).

Algorithm 1 ***Link–Mask***.

Input: pk' and $\left\{V^{(x,t)}\right\}_{x\in[n],t\in[N]}$

Output: $X \in \mathbb{Z}_q^{m'\times N}$

(1) Define $L_{x,t} \in \mathbb{Z}_q^{m'\times N}$, for $x \in [n], t \in [N]$ by

$$L_{x,t}[a,b] = \begin{cases} A'^{\mathrm{T}}[a,x] & t = b \\ 0 & \text{other} \end{cases}$$

(2) Output $X = \sum_{x=1}^{n}\sum_{t=1}^{N} V^{(x,t)}M^{-1}(L_{x,t}) \in \mathbb{Z}_q^{m'\times N}$.

Proof. Since $V^{(x,t)}$ is a IBFHE ciphertext of $Y[x,t]$ under $(pk, sk) = (A, s)$, we have $s^{\mathrm{T}}V^{(x,t)} = Y[x,t]s^{\mathrm{T}}M + e_{x,t}$. Hence, it holds that

$$\begin{aligned}\mathbf{s}^{\mathrm{T}}X &= s^{\mathrm{T}}\sum_{x,t}^{n,N} V^{(x,t)}M^{-1}(L_{x,t}) \\ &= \sum_{x,t}^{n,N}\left(Y[x,t]s^{\mathrm{T}}M + e_{x,t}\right)M^{-1}(L_{x,t}) \\ &= \sum_{x,t}^{n,N}\left(Y[x,t]s^{\mathrm{T}}L_{x,t} + e_{x,t}M^{-1}(L_{x,t})\right) \\ &= \sum_{x,t}^{n,N}\left(Y[x,t]s^{\mathrm{T}}L_{x,t} + e'_{x,t}\right). \\ &= s^{\mathrm{T}}\sum_{x,t}^{n,N} Y[x,t]L_{x,t} + \sum_{x,t}^{n,N} e'_{x,t}\end{aligned}$$

where $e'_{x,t} = e_{x,t}M^{-1}(L_{x,t})$ has a norm $\|e'_{x,t}\|_\infty \leq (m+1)\beta$.

Now it suffices to show that $\sum_{x,t}^{n,N} Y[x,t]L_{x,t} = A'^{\mathrm{T}}Y$. Note that $L_{x,t}$ has x-th column of A'^{T} on the t-th column and 0 elsewhere.

$$\sum_{x=1}^{n}\sum_{t=1}^{N}\boldsymbol{Y}[x,t]\boldsymbol{L}_{x,t}=\sum_{x=1}^{n}\sum_{t=1}^{N}\begin{bmatrix}0&\cdots&\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[1,x]&\cdots&0\\ \vdots&\ddots&\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[2,x]&\cdots&0\\ \vdots&\vdots&\vdots&\cdots&\vdots\\ 0&\cdots&\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[n,x]&\cdots&0\end{bmatrix}$$

$$=\sum_{t=1}^{N}\begin{bmatrix}0&\cdots&\sum_{x=1}^{n}\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[1,x]&\cdots&0\\ \vdots&\ddots&\sum_{x=1}^{n}\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[2,x]&\cdots&0\\ \vdots&\vdots&\vdots&\cdots&\vdots\\ 0&\cdots&\sum_{x=1}^{n}\boldsymbol{Y}[x,t]\boldsymbol{A}'^{T}[n,x]&\cdots&0\end{bmatrix}$$

$$=\sum_{t=1}^{N}\begin{bmatrix}0&\cdots&\left\langle\boldsymbol{A}_1'^{T_{row}},\boldsymbol{Y}_t^{col}\right\rangle&\cdots&0\\ \vdots&\ddots&\left\langle\boldsymbol{A}_2'^{T_{row}},\boldsymbol{Y}_t^{col}\right\rangle&\cdots&0\\ \vdots&\vdots&\vdots&\cdots&\vdots\\ 0&\cdots&\left\langle\boldsymbol{A}_n'^{T_{row}},\boldsymbol{Y}_t^{col}\right\rangle&\cdots&0\end{bmatrix}=\boldsymbol{A}'^{T}\boldsymbol{Y}$$

where $\boldsymbol{A}_l'^{T_{row}}$ denotes the l-th row of $\boldsymbol{A}'^{\mathrm{T}}$ and $\boldsymbol{Y}_l^{col}$ denotes the l-th column of $\boldsymbol{Y}$.

To sum up,

$$\boldsymbol{s}^{\mathrm{T}}X=\boldsymbol{s}^{\mathrm{T}}\sum_{x,t}^{n,N}\boldsymbol{Y}[x,t]\boldsymbol{L}_{x,t}+\sum_{x,t}^{n,N}\boldsymbol{e}_{x,t}'=\boldsymbol{s}^{\mathrm{T}}\boldsymbol{A}'^{\mathrm{T}}\boldsymbol{Y}+\boldsymbol{e}$$

where $\boldsymbol{e}=\sum_{x,t}^{n,N}\boldsymbol{e}_{x,t}'$ has norm $\|\boldsymbol{e}\|_\infty\leq n(m+1)^2\beta$. □

5.2. Construction

The basic parameter definition of the scheme: Let n as security parameter, $\boldsymbol{L}$ denote the maximum depth of homomorphic calculation allowed for the circuit, $q=q(n,\boldsymbol{L})$ be a sufficiently large prime, D denote the maximum number of distinct identities supported by the scheme, $m, m', \overline{m}, w, k, N=(m+1)k$ and FRD encoding function $\boldsymbol{H}$ are the same as the definitions in the above IBFHE encryption scheme. According to the notation in [16], the gadget matrix $\boldsymbol{M}\in\mathbb{Z}^{m'\times N}$ is extended to $\hat{\boldsymbol{M}}\in\mathbb{Z}_q^{Dm'\times DN}$. According to lemma 2, it is known that for any matrix $\boldsymbol{A}\in\mathbb{Z}_q^{Dm'\times DN}$, there exists a function $\hat{\boldsymbol{M}}^{-1}(\cdot)$ such that $\hat{\boldsymbol{M}}^{-1}(\boldsymbol{A})\in\{0,1\}^{DN\times DN}$, satisfying $\hat{\boldsymbol{M}}\hat{\boldsymbol{M}}^{-1}(\boldsymbol{A})=\boldsymbol{A}$.

- **mIBFHE.Setup**$\left(1^n,1^{\boldsymbol{L}},1^D\right)$: Input the security parameter n, the maximum depth $\boldsymbol{L}$ that the circuit allows homomorphic operations, and the maximum number of different identities D supported by the scheme. Run the **IBFHE.Setup** algorithm and output the master public key $\boldsymbol{MPK}=(\boldsymbol{A},\boldsymbol{u})$ and the master secret key $\boldsymbol{MSK}=\boldsymbol{R}$.
- **mIBFHE.Extract**$\left(\boldsymbol{MPK},\boldsymbol{MSK},\left[id_j\right]_{j\in[D]}\right)$: Input the master public key $\boldsymbol{MPK}$, master secret key $\boldsymbol{MSK}$, and user's identity vector $\left[id_j\right]_{j\in[D]}\in\mathbb{Z}_q^n$. Run the **IBFHE.Extract** algorithm to generate secret key $\boldsymbol{s}_{id_1},\ldots,\boldsymbol{s}_{id_D}$ corresponding to identity $id_1,\ldots,id_D$ and the related public key $\boldsymbol{A}_{id_1}',\ldots,\boldsymbol{A}_{id_D}'$, and construct the joint secret key by horizontally appending all the secret-keys in sequence $\hat{\boldsymbol{s}}=\left[\boldsymbol{s}_{id_1},\ldots,\boldsymbol{s}_{id_D}\right]\in\mathbb{Z}_q^{Dm'}$. Output the public key $\boldsymbol{A}_{id_1}',\ldots,\boldsymbol{A}_{id_N}'$ and the joint secret key vector $\hat{\boldsymbol{s}}$.
- **mIBFHE.Enc**$\left(\boldsymbol{MPK},\left[id_j\right]_{j\in[D]},\left[\boldsymbol{A}_{id_j}'\right]_{j\in[D]},\mu,i\right)$: Input the master public key $\boldsymbol{MPK}$, the user's identity vector $\left[id_j\right]_{j\in[D]}$ and its corresponding public key $\left[\boldsymbol{A}_{id_j}'\right]_{j\in[D]}$, the encrypted plaintext message $\mu\in\{0,1\}$ and the identity $i\in[D]$ that needs to be

extended. Run the algorithm to output the extended ciphertext $\hat{\boldsymbol{C}}_i$ corresponding to identity id_i. The specific operation steps are as follows:

1. **Single identity encryption step:** Run **IBFHE**.**Enc**$(\boldsymbol{MPK}, id_i, \mu)$ to generate identity id_i single identity IBFHE ciphertext $\boldsymbol{C} = \boldsymbol{A}_{id}^{'\mathrm{T}}\boldsymbol{Y} + \mu\boldsymbol{M} + \boldsymbol{E}$. In this step, the party (here the i-th party) keeps its $\boldsymbol{Y}$ for the next step;
2. **Multi-identity ciphertext expansion step:** Input a single-identity ciphertext $\boldsymbol{C}$, the public keys of the other parties, and a randomness $\boldsymbol{Y}$ selected from **IBFHE**.**Enc**. Execute steps (a)–(d) as follows:

(a) $$\left\{\boldsymbol{V}_{i,j}^{(\boldsymbol{x},\boldsymbol{t})}\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]} \leftarrow \left\{\textbf{IBFHE.Enc}\left(\boldsymbol{MPK}, id_j, \boldsymbol{Y}[\boldsymbol{x},\boldsymbol{t}]\right)\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]} \text{ for } j \in [D].$$
$$\left\{\begin{matrix}-(\boldsymbol{x},\boldsymbol{t})\\ \boldsymbol{V}_{i,j}\end{matrix}\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]} \leftarrow \left\{\textbf{IBFHE.Enc}\left(MPK, id_j, \overline{\boldsymbol{Y}}[\boldsymbol{x},\boldsymbol{t}]\right)\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]} \text{ for } j \in [D]\backslash\{i\},$$
where $\boldsymbol{Y}$ was chosen in the single identity encryption step and $\overline{\boldsymbol{Y}}$ is randomly chosen from $\{0,1\}^{n\times N}$.

(b) Compute
$$X_i^j \leftarrow \textbf{Link–Mask}\left(\left\{\boldsymbol{V}_{i,j}^{(\boldsymbol{x},\boldsymbol{t})}\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]}, \boldsymbol{A}_{id_i}^{'}\right), j \in [D].$$
$$\overline{X}_i^j \leftarrow \textbf{Link–Mask}\left(\left\{\begin{matrix}-(\boldsymbol{x},\boldsymbol{t})\\ \boldsymbol{V}_{i,j}\end{matrix}\right\}_{\boldsymbol{x}\in[n],\boldsymbol{t}\in[N]}, \boldsymbol{A}_{id_j}^{'}\right), j \in [D]\backslash\{i\}.$$

(c) Choose $\boldsymbol{Q} \xleftarrow{\$} \mathbb{Z}_q^{m'\times N}$. Set the matrix $\boldsymbol{Q}_h \in \mathbb{Z}_q^{m'\times N}$ having the last row $\boldsymbol{s}_{id_i}\boldsymbol{Q} + \bar{\boldsymbol{e}}_h$ and the rest rows zero, where $\boldsymbol{s}_{id_i}$ is the secret key of the party i, $\bar{\boldsymbol{e}}_h$ is chosen from χ^N, $\forall h \in [D]\backslash\{i\}$.

(d) Define the extended ciphertext matrix $\hat{\boldsymbol{C}}_i \in \mathbb{Z}_q^{Dm'\times DN}$ of the initial ciphertext $\boldsymbol{C}$ as

$$\hat{\boldsymbol{C}}_{\boldsymbol{i}}\begin{bmatrix} \boldsymbol{C}_{\boldsymbol{i}}^1 & 0 & \cdots & 0 & 0 \\ 0 & \boldsymbol{C}_{\boldsymbol{i}}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{Q} & \cdots & \boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{i}} & \cdots & \boldsymbol{Q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \boldsymbol{C}_{\boldsymbol{i}}^{\boldsymbol{D}} \end{bmatrix}$$

Which is concatenated by D^2 number of $m' \times N$ sub-matrices. The diagonal sub-matrices of $\hat{\boldsymbol{C}}_i$ are $\boldsymbol{C}_i^j = \boldsymbol{C} - X_i^j + \overline{X}_i^j - \boldsymbol{Q}_j$ for $j \in [D]\backslash\{i\}$ and the i-th diagonal sub-matrix is $\boldsymbol{C} - X_i^i$. Lastly, $\boldsymbol{Q}$ is on the i-th row and zero matrix $0^{m'\times N}$ is elsewhere.

- **mIBFHE**.**Eval**$\left(MPK, (\hat{\boldsymbol{C}}_1, \ldots, \hat{\boldsymbol{C}}_{\boldsymbol{t}}), f\right)$: Input the master public key MPK, Boolean circuit f, and the extended ciphertext $\hat{\boldsymbol{C}}_1, \ldots, \hat{\boldsymbol{C}}_{\boldsymbol{t}}$ which are the ciphertext encrypted under different identities id. Output the operation ciphertext $\hat{\boldsymbol{C}} = f\left(\hat{\boldsymbol{C}}_1, \ldots, \hat{\boldsymbol{C}}_{\boldsymbol{t}}\right)$, where the homomorphic addition is $\hat{\boldsymbol{C}}_{\boldsymbol{Add}} = \hat{\boldsymbol{C}}_1 + \hat{\boldsymbol{C}}_2$ and the homomorphic multiplication is $\hat{\boldsymbol{C}}_{Mult} = \hat{\boldsymbol{C}}_1\boldsymbol{M}^{-1}\left(\hat{\boldsymbol{C}}_2\right)$. According to the definitions of addition and multiplication, the homomorphic NAND operation is defined as $\hat{\boldsymbol{C}}_{\boldsymbol{NAND}} = \hat{\boldsymbol{M}} - \hat{\boldsymbol{C}}_1\hat{\boldsymbol{M}}^{-1}\left(\hat{\boldsymbol{C}}_2\right)$.
- **mIBFHE**.**Dec**$\left(MPK, \hat{\boldsymbol{s}}, \hat{\boldsymbol{C}}\right)$: Input the master public key MPK, the joint secret key $\hat{\boldsymbol{s}}$ and the extended ciphertext $\hat{\boldsymbol{C}}$ to be decrypted. Set a vector $\hat{\omega} = \left[\left|\frac{q}{2}\right|, 0, \ldots, 0\right] \in \mathbb{Z}_q^{Dm'}$, compute $\mu' = \hat{\boldsymbol{s}}^{\mathrm{T}}\cdot\hat{\boldsymbol{C}}\cdot\hat{\boldsymbol{M}}^{-1}(\hat{\omega})$, and output $\mu = \left\lfloor\frac{2\mu'}{q}\right\rfloor$.

Correctness. Let $\hat{C}_i$ be the multi-identity ciphertext of a bit μ by i-th user from the **mIBFHE**.**Enc** algorithm:

$$\hat{C}_i \leftarrow \text{mIBFHE.Enc}\left(MPK, \left[id_j\right]_{j\in[D]}, \left[A'_{id_j}\right]_{j\in[D]}, \mu, i\right)$$

where C is a single identity IBFHE ciphertext. For the joint secret key $\hat{s} = \left[s_{id_1}, \ldots, s_{id_D}\right] \in \mathbb{Z}_q^{Dm'}$ and the gadget matrix $\hat{M} \in \mathbb{Z}_q^{Dm'\times DN}$, if $\hat{C}_i$ satisfies the relation $\hat{s}^{\mathrm{T}}\hat{C}_i \approx \mu\hat{s}^{\mathrm{T}}\hat{M}$, then we can naturally generalize the arguments of the scheme in [7]. The correctness of encryption and evaluation can be realized, and an effective mIBFHE scheme can be obtained.

Now, we are ready to prove the correctness of multi-identity ciphertext. We recall that for a valid output X from $\textbf{\textit{Link–Mask}}\left(pk', \left(V^{1,1}, \ldots, V^{n,N}\right)\right)$ with respect to a 0-1 matrix Y, we have $s^{\mathrm{T}}X = s^{\mathrm{T}}A'^{\mathrm{T}}Y + e$ for $e_\infty \le n(m+1)^2\beta$. By the definition, we have

$$\begin{aligned}\hat{s}^{\mathrm{T}}\hat{C}_i &= \left[s_{id_1}^{\mathrm{T}}C_i^1 + s_{id_i}^{\mathrm{T}}Q, \ldots, s_{id_i}^{\mathrm{T}}C_i^i, \ldots, s_{id_D}^{\mathrm{T}}C_i^D + s_{id_i}^{\mathrm{T}}Q\right]\\ &= \left[s_{id_1}^{\mathrm{T}}\left(C - X_i^1 + \overline{X}_i^1 - Q_1\right) + s_{id_i}^{\mathrm{T}}Q, \ldots, s_{id_i}^{\mathrm{T}}\left(C - X_i^i\right),\right.\\ &\left.\ldots, s_{id_D}^{\mathrm{T}}\left(C - X_i^D + \overline{X}_i^D - Q_D\right) + s_{id_i}^{\mathrm{T}}Q\right]\end{aligned}$$

Let's see how the bit message μ is correctly recovered and check the error bound by using the following properties.

(1) $s_{id_j}^{\mathrm{T}}C = s_{id_j}^{\mathrm{T}}\left(A_{id_i}'^{\mathrm{T}}Y_i + \mu M + E\right) = s_{id_j}^{\mathrm{T}}A_{id_i}'^{\mathrm{T}}Y_i + \mu s_{id_j}^{\mathrm{T}}M + e'$, where $\| e' \|_\infty \le (m+1)\beta$;

(2) $s_{id_j}^{\mathrm{T}}X_i^j = s_{id_j}^{\mathrm{T}}A_{id_i}'^{\mathrm{T}}Y_i + e_j''$, where $\| e_j'' \|_\infty \le n(m+1)^4\beta$;

(3) $s_{id_i}^{\mathrm{T}}X_i^i = s_{id_i}^{\mathrm{T}}A_{id_i}'^{\mathrm{T}}Y_i + e_j'' = \widetilde{e}_i$, where $\| \widetilde{e}_i \|_\infty \le n\left[(m+1)^4 + m + 1\right]\beta$;

(4) $s_{id_j}^{\mathrm{T}}\overline{X}_i^j = s_{id_j}^{\mathrm{T}}A_{id_j}'^{\mathrm{T}}\overline{Y} + \overline{e}_j'' = \widetilde{e}_j$, where $\| \widetilde{e}_j \|_\infty \le n\left[(m+1)^4 + m + 1\right]\beta$;

(5) $s_{id_j}^{\mathrm{T}}Q_j = s_{id_i}^{\mathrm{T}}Q + \overline{e}_j$, where $\| \overline{e}_j \|_\infty \le (m+1)\beta$;

$$\begin{aligned}&\therefore s_{id_j}^{\mathrm{T}}\left(C - X_i^j + \overline{X}_i^j - Q_j\right) + s_{id_i}^{\mathrm{T}}Q = \mu s_{id_j}^{\mathrm{T}}M + \hat{e}_j\\ &s_{id_i}^{\mathrm{T}}\left(C - X_i^i\right) = \mu s_{id_i}^{\mathrm{T}}M + \hat{e}_i\end{aligned}$$

Therefore, we have $\hat{s}^{\mathrm{T}}\hat{C}_i = \mu\hat{s}^{\mathrm{T}}\hat{M} + \hat{e}$ where $\hat{e} = [\hat{e}_1, \ldots, \hat{e}_i, \ldots, \hat{e}_D] \in \mathbb{Z}_q^{D\times N}$ and $\|\hat{e}\|_\infty \le \left[2n(m+1)^4 + (n+1)(m+1)\beta\right]$. In the decryption procedure, this error is multiplied by $\sqrt{DN}$. By our choice of the parameter, $\sqrt{DN}\left[2n(m+1)^4 + (n+1)(m+1)\beta\right] < \frac{q}{4}$.

Homomorphic property. The homomorphic property of the mIBFHE scheme follows directly from the IBFHE scheme in the fourth chapter, because the **mIBFHE**.**Eval** algorithm is basically the same as the **IBFHE**.**Eval** algorithm except for the dimension expansion, the matrix $\hat{M}$ and the randomization function $\hat{M}^{-1}(\cdot)$. The following is the homomorphism analysis of the mIBFHE scheme:

Definition 9. Let $\hat{C}_1, \hat{C}_2$ be an extended ciphertext matrix corresponding to plaintext μ_1, μ_2, respectively, and the secret key is $\hat{s} \in \mathbb{Z}_q^{Dm'}$, satisfying $\hat{s}^{\mathrm{T}}\hat{C}_1 = \mu_1\hat{s}^{\mathrm{T}}\hat{M} + \hat{e}_1$, $\hat{s}^{\mathrm{T}}\hat{C}_2 = \mu_2\hat{s}^{\mathrm{T}}\hat{M} + \hat{e}_2$, where $\mu_1, \mu_2 \in \{0, 1\}$, $\|\hat{e}_1\|_\infty \le \hat{\beta}_1$, $\|\hat{e}_2\|_\infty \le \hat{\beta}_2$.

(1) **Homomorphic addition:** $\hat{C}_{Add} = \hat{C}_1 + \hat{C}_2$, satisfy $\hat{s}^{\mathrm{T}}\hat{C}_{Add} = \hat{s}^{\mathrm{T}}\left(\hat{C}_1 + \hat{C}_2\right) = (u_1 + \mu_2)$ $\hat{s}^{\mathrm{T}}\hat{M} + \hat{e}^+$,where $\hat{e}^+ = \hat{e}_1 + \hat{e}_2$. Obviously $\hat{C}_{Add}$ is $\hat{\beta}_1 + \hat{\beta}_2$ noise ciphertext, that is, after one-time homomorphic addition, the error increases by 2 times the factor.

(2) **Homomorphic multiplication:** $\hat{C}_{Mult} = \hat{C}_1\hat{M}^{-1}(\hat{C}_2)$, satisfy $\hat{s}^T\hat{C}_{Mult} = \hat{s}^T\hat{C}_1M^{-1}(\hat{C}_2) = (\mu_1\mu_2)\hat{s}^T\hat{M} + \hat{e}^{\times}$, where $\hat{e}^{\times} = \hat{e}_1\hat{M}^{-1}(\hat{C}_2) + \mu_1\hat{e}_2$. Obviously $\| \hat{e}'' \|_{\infty} \leq \left(\sqrt{DN}\hat{\beta}_1 + \hat{\beta}_2\right)$, $\hat{C}_{Mult}$ is $\left(\sqrt{DN}\hat{\beta}_1 + \hat{\beta}_2\right)$ noise ciphertext. The same calculation is also applicable to NAND gates.

Multi-identity ciphertext security. If the IBE scheme constructed in this paper is IND-sID-CPA secure, then the mIBFHE scheme proposed in this paper is also IND-sID-CPA secure.

By using constructive proof, the masking scheme constructed by **LinkMask** algorithm is IND-sID-CPA security. From theorem 2, it can be seen that the IBFHE scheme is IND-sID-CPA security. In summary, the mIBFHE scheme is also IND-sID-CPA security.

5.3. Efficiency Analysis of MIBFHE Scheme

The mIBFHE scheme proposed in this paper is compared with the CM15 scheme proposed by Clear et al. [12]. The comparison results are shown in Table 2.

Table 2. Comparison of main parameters of mIBFHE scheme.

Scheme	Dimension	q	Size of $\hat{s}$	Size of $\hat{C}$	Noise Rate
[12]	$6n\log q$	$8\omega\beta(DN+1)^L$	ND	$DN \times DN$	$DN+1$
Ours	$2n\log q$	$5\omega\beta\left(\sqrt{DN}+1\right)^L$	$m'D$	$Dm' \times DN$	$\sqrt{DN}+1$

From the analysis in Table 2, it can be seen that compared with the CM15 scheme based on the trapdoor generation algorithm in [21], the mIBFHE scheme in this paper used the MP12 trapdoor generation algorithm and the preimage matrix for encryption. The scheme is more concise and the encryption algorithm is simpler. Therefore, the main efficiency parameters of the mIBFHE scheme in this paper are significantly optimized. The lattice security dimension m is reduced from $6n\log q$ to $2n\log q$, the size of the joint secret key $\hat{s}$ is reduced from ND to $m'D$, and the size of extended ciphertext is reduced from $DN \times DN$ to $m'D \times DN$.

6. Conclusions

Aiming at the problem that low efficiency of trapdoor function and sampling algorithm in lattice-based multi-identity fully homomorphic encryption scheme, this paper first constructed an efficient and transformable IBE scheme based on MP12 trapdoor, which solves the problem that the trapdoor of IBE scheme is difficult to realize and the preimage sampling is complex. Based on the LWE hardness problem, it is proved that the scheme is IND-sID-CPA security under the standard model. Then, the IBE scheme is transformed into IBFHE scheme by using the approximate eigenvector to eliminate the evaluation key and the preimage matrix. This IBFHE scheme satisfies the homomorphism operation. Finally, the constructed masking scheme and the extended ciphertext method are used to transform the IBFHE scheme into mIBFHE scheme. Compared with similar schemes, our scheme is more concise and efficient, and the parameters are more compact.

Author Contributions: Conceptualization, H.F. and R.H.; formal analysis, H.F.; funding acquisition, R.H.; methodology, H.F.; validation, H.F., R.H. and F.L.; writing—original draft, H.F.; writing—review & editing, H.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation Project of China under Grant No. 62062009 and the Guangxi Innovation-driven Development Project under Grant Nos. AA17204058-17 and AA18118047-7.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur. Comput.* **1978**, *4*, 169–180.
2. Gentry, C. Fully homomorphic encryption using ideal lattices. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; pp. 169–178.
3. Van Dijk, M.; Gentry, C.; Halevi, S.; Vaikuntanathan, V. Fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2010, Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*; French Riviera, France, 30 May–3 June 2010, Proceedings 29; Springer: Berlin/Heidelberg, Germany, 2010; pp. 24–43.
4. Brakerski, Z.; Vaikuntanathan, V. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Advances in Cryptology–CRYPTO 2011, Proceedings of the 31st Annual Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2011; Proceedings 31. Springer: Berlin/Heidelberg, Germany, 2011; pp. 505–524.
5. Brakerski, Z.; Vaikuntanathan, V. Efficient Fully Homomorphic Encryption from (Standard) LWE. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 22–25 October 2011; pp. 97–106.
6. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) Fully homomorphic encryption without bootstrapping. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Berkeley, CA, USA, 31 January–3 February 2012; pp. 309–325.
7. Gentry, C.; Sahai, A.; Waters, B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Advances in Cryptology–CRYPTO 2013, Proceedings of the 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 2013; Proceedings, Part I.. Springer: Berlin/Heidelberg, Germany, 2013; pp. 75–92.
8. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Advances in Cryptology-Crypto'84, Santa Barbara, CA, USA, 19–22 August 1984; pp. 341–349.
9. Naccache, D. Is Theoretical Cryptography Any Good in Practice [OL]. Invited Talk at Crypto/CHES 2010. Available online: http://www.iacr.org/workshops/ches/ches2010 (accessed on 18 August 2010).
10. Clear, M.; McGoldrick, C. Bootstrappable identity-based fully homomorphic encryption. In Cryptology and Network Security, Proceedings of the 13th International Conference, CANS 2014, Heraklion, Greece, 22–24 October 2014; Proceedings 13. Springer International Publishing: Cham, Switzerland, 2014; pp. 1–19.
11. Garg, S.; Gentry, C.; Halevi, S.; Raykova, M.; Sahai, A.; Waters, B. Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), Berkeley, CA, USA, 26–29 October 2013; pp. 40–49.
12. Clear, M.; McGoldrick, C. Multi-identity and multi-key leveled FHE from learning with errors. In Advances in Cryptology—CRYPTO 2015, Proceedings of the 35th Annual Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; Proceedings, Part II 35. Springer: Berlin/Heidelberg, Germany, 2015; pp. 630–656.
13. TU, G.; Yang, X.; Zhou, T. Efficient identity-based multi-identity fully homomorphic encryption scheme. *J. Comput. Appl.* **2019**, *39*, 750.
14. Cash, D.; Hofheinz, D.; Kiltz, E.; Peikert, C. Bonsai trees, or how to delegate a lattice basis. In Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques, French Riviera, France, 30 May–3 June 2010; pp. 523–552.
15. Shen, T.; Wang, F.; Chen, K.; Wang, K.; Li, B. Efficient leveled (multi) identity-based fully homomorphic encryption schemes. *IEEE Access* **2019**, *7*, 79299–79310. [CrossRef]
16. Mukherjee, P.; Wichs, D. Two round multiparty computation via multi-key FHE. In Advances in Cryptology–EUROCRYPT 2016, Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016; Proceedings, Part II 35. Springer: Berlin/Heidelberg, Germany, 2016; pp. 735–763.
17. Pal, T.; Dutta, R. Chosen-ciphertext secure multi-identity and multi-attribute pure FHE. In Cryptology and Network Security, Proceedings of the 19th International Conference, CANS 2020, Vienna, Austria, 14–16 December 2020; Proceedings 19. Springer International Publishing: Cham, Switzerland, 2020; pp. 387–408.
18. Shen, T.; Wang, F.; Chen, K.; Shen, Z.; Zhang, R. Compressible multikey and multi-identity fully homomorphic encryption. *Secur. Commun. Netw.* **2021**, *2021*, 1–14. [CrossRef]
19. Gentry, C.; Halevi, S. Compressible FHE with applications to PIR. In Theory of Cryptography, Proceedings of the 17th International Conference, TCC 2019, Nuremberg, Germany, 1–5 December 2019; Proceedings, Part II.. Springer International Publishing: Cham, Switzerland, 2019; pp. 438–464.
20. Liu, W.; Wang, F.; Jin, X.; Chen, K.; Shen, Z. Leveled Multi-Hop Multi-Identity Fully Homomorphic Encryption. *Secur. Commun. Netw.* **2022**, *2022*, 1023439. [CrossRef]
21. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, Columbia, BC, Canada, 17–20 May 2008; pp. 197–206.
22. Peikert, C.; Shiehian, S. Multi-key FHE from LWE, revisited. In Theory of Cryptography, Proceedings of the 14th International Conference, TCC 2016-B, Beijing, China, 31 October–3 November 2016; Proceedings, Part II.. Springer: Berlin/Heidelberg, Germany, 2016; pp. 217–238.
23. Ajtai, M. Generating hard instances of the short basis problem. In Automata, Languages and Programming, Proceedings of the 26th International Colloquium, ICALP'99, Prague, Czech Republic, 11–15 July 1999; Proceedings 26. Springer: Berlin/Heidelberg, Germany, 1999; pp. 1–9.

24. Alwen, J.; Peikert, C. Generating Shorter Bases for Hard Random Lattices. In Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science STACS 2009, Freiburg, Germany, 26–28 February 2009; IBFI Schloss Dagstuhl: London, UK, 2009; pp. 75–86.
25. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. *Eurocrypt* **2012**, *7237*, 700–718.
26. Agrawal, S.; Boneh, D.; Boyen, X. Efficient lattice (h) ibe in the standard model. *Eurocrypt* **2010**, *6110*, 553–572.
27. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005; pp. 84–93.
28. Peikert, C. Public-key cryptosystems from the worst-case shortest vector problem. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; pp. 333–342.
29. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]
30. Kim, E.; Lee, H.S.; Park, J. Towards round-optimal secure multiparty computations: Multikey FHE without a CRS. In Information Security and Privacy, Proceedings of the 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, 11–13 July 2018; Proceedings 23; Springer International Publishing: Cham, Switzerland, 2018; pp. 101–113.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.