Research Article

Leveled Multi-Hop Multi-Identity Fully Homomorphic Encryption

Wen Liu,1 Fuqun Wang,1,2,3 Xiaodan Jin,1 Kefei Chen,1,2,3 and Zhonghua Shen1,2

1School of Mathematics, Hangzhou Normal University, Hangzhou 311121, China
2Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China
3Westone Cryptologic Research Center, Beijing 100071, China

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

Received 28 November 2021; Revised 29 January 2022; Accepted 15 February 2022; Published 22 March 2022

Copyright © 2022 Wen Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Gentry, Sahai, and Waters (CRYPTO 2013) proposed the notion of multi-identity fully homomorphic encryption (MIFHE), which allows homomorphic evaluation of data encrypted under multiple identities. Subsequently, Clear and McGoldrick (CANS 2014, CRYPTO 2015) proposed leveled MIFHE candidates. However, the proposed MIFHE is either based on \( \mathcal{O} \), which is a nonstandard assumption or single hop; that is, an arbitrary “evaluated” ciphertext under a set of identities is difficult to further evaluate when new ciphertexts are encrypted under additional identities. To overcome these drawbacks, we propose a leveled multi-hop MIFHE scheme. In a multi-hop MIFHE scheme, one can evaluate a group of ciphertexts under a set of identities to obtain an “evaluated” ciphertext, which can be further evaluated with other ciphertexts encrypted under additional identities. We also show that the proposed MIFHE scheme is secure against selective identity and chosen-plaintext attacks (IND-sID-CPA) under the learning with errors (LWE) assumption.

1. Introduction

The idea of homomorphic encryption was first proposed by Rivest et al. [1] in 1978. How to construct a scheme with homomorphic properties is used to be a difficult problem for cryptographers. With the advent of the information age and the development of cloud computing technology, it is particularly urgent to solve this problem. It was not until 2009 that Gentry proposed the first fully homomorphic encryption (FHE) system based on ideal lattices, allowing anyone without a secret key to compute any efficiently computable function over encrypted data [2]. Because FHE is suitable to apply cloud computing without compromising security, it has quickly become a research hot topic [3–7].

All of the above FHE schemes are single-key homomorphic, that is, only suitable for the homomorphic evaluation of ciphertext encrypted under a single key. However, in many realistic scenarios, the ciphertext of homomorphic encryption is usually encrypted under multiple different keys. Therefore, at STOC 2012, López-Alt, Tromer, and Vaikuntanathan [8] proposed the first cryptographic construction of multi-key full homomorphic encryption (MKFHE) based on NTRU cryptography, which enables the evaluation of data encrypted under different keys. Subsequently, a large number of articles appeared to improve MKFHE, including single hop only [9, 10], multi-hop with bootstrapping [11–16], and multi-hop without bootstrapping [17].

Although (MK) FHEs have extensive applications, they require complex certificate management in implementation. To simplify the certificate management, Naccache [18] introduced a notion of identity-based fully homomorphic encryption (IBFHE), where there is no user-specific key that the evaluator must use. In particular, in an IBFHE scheme, data encrypted under a single identity can perform homomorphic operations by any evaluator with only public parameters. In 2013, Gentry et al. [7] constructed an IBFHE scheme from GSW-FHE. The IBFHE scheme only allows homomorphic evaluation of encrypted data under single identity but not multiple identities. Clear and McGoldrick...
[19] gave an MIFHE based on the indistinguishability obfuscation \((i \otimes)\) [20] to overcome the disadvantage of single identity at CANS 2014. Then, they [9] proposed a leveled MIFHE candidate under LWE in the random oracle model at CRYPTO 2015. However, the later scheme needs to set the number of users participating in homomorphic evaluation in advance, and new users cannot be added to the operation process, which is single hop in MIFHE cryptography. In 2017, Canetti et al. [21] proposed two MIFHE schemes. The first combines MKFHE and identity-based encryption (IBE) on the fly. Therefore, the ciphertext extension depends on the number of ciphertexts, which is not compact. The second is nonleveled, but uses \(i \otimes\). In 2020, with the help of an MKFHE, Pal and Dutta [22] extended IBE to a CCA1 secure MIFHE scheme. However, their extension process uses witness pseudorandom function (WPRF), which is a non-standard assumption. Recently, Shen et al. [23] proposed a compressible MIFHE scheme based on [9, 10, 24]. The scheme is selectively secure under the LWE assumption and can reach an optimal compression rate, but it is single hop.

Thus, it is interesting to construct a compact MIFHE scheme with the multi-hop homomorphism under standard assumption, where one can evaluate a group of ciphertexts under a set of identities to obtain an “evaluated” ciphertext, which can be further evaluated with other ciphertexts encrypted under additional identities.

1.1. Contribution. We propose a leveled multi-hop MIFHE scheme adapted from the GPV-FHE [25], following the construction method of the PS-MKFHE, which is a leveled multi-hop MKFHE scheme built by Peikert and Shiehian [17]. We show that it is compact and secure against the IND-sID-CPA attack under the LWE assumption in the random oracle model. In our construction, we use a fully homomorphic commitment to commit the plaintext bit to help homomorphic operations. Additionally, by combining the transformation of MIFHE to the nonadaptive chosen-plaintext attack (CCA1) secure FHE proposed by Canetti et al. [21], we can obtain a CCA1 secure FHE with multi-hop homomorphism. Finally, we note that our construction can be applied to the ring setting as [9] for shorter parameters.

1.2. Technical Overview. It is well known that the efficient multi-hop MKFHE obtained by bootstrapping is difficult to generalize MIFHE, because the public homomorphic evaluation key cannot be extracted from identity secret key \(SK_{id}\) as it is a ciphertext of \(SK_{id}\) and even the \(SK_{iq}\) is not generated before decryption. Therefore, we focus on the PS-MKFHE [17], which does not use bootstrapping. Our key observation is that we can construct a multi-hop MIFHE scheme following the ideas introduced by Peikert and Shiehian [17]. They built multi-hop MKFHE schemes to overcome the single hop drawback of the MKFHE schemes [9, 10]. In their first construction, a ciphertext consists of \((C, F, D)\), where \(C\) is a GSW-FHE ciphertext [7] that encrypts message \(\mu\), \(F\) is a fully homomorphic commitment [26] to the same message, and \(D\) is a special encryption of the commitment randomness implied in \(F\) under the same key to \(C\). Here, \((D, F)\) provides the power of expanding \(C\) to \(C'\) with additional keys (with a part of the public key) and preserves some invariant, which can be further used. In total, after expanding or computing, the form of \((C, F, D)\) remains. This finding is the reason to support the multi-hop computation with respect to additional keys.

However, it is nontrivial to construct a multi-hop MIFHE scheme because PS-MKFHE [17] is built from GSW-FHE [7] but not from GPV-FHE [26]. For simplicity, we now informally describe a multi-hop MKFHE from GPV-FHE [26], which can be converted into a multi-hop MIFHE scheme (see Section 3).

In a multi-hop MKFHE scheme based on GPV-FHE, a ciphertext under a secret key \(t \in \mathbb{Z}^{m}\), consists of four components \((C, D, E, F)\):

1. A GPV-FHE ciphertext \(C \in \mathbb{Z}_{q}^{mk \times m}\) that encrypts \(\mu\) under \(t\).
2. A GPV-FHE style fully homomorphic commitment \(D \in \mathbb{Z}_{q}^{m \times m}\) to the same message \(\mu\) with underlying commitment randomness \((R, X)\).
3. A special encryption \(E \in \mathbb{Z}_{q}^{m \times m}\) under \(t\) of the former part of the commitment randomness \(R\).
4. Another special encryption \(F \in \mathbb{Z}_{q}^{m \times m}\) under \(t\) of the latter part of the commitment randomness \(X\).

To expand \(C\) to an additional secret key \(\tilde{t} \in \mathbb{Z}_{q}^{m}\), we define

\[
C' = \begin{pmatrix} C & U \\ 0 & D \end{pmatrix} \in \mathbb{Z}_{q}^{(m(k+1) \times m(k+1))},
\]

where \(U\) is derived from \(E\). The commitment \(D\) is preserved, and \(E, F\) are padded with zeros to fit the long secret key \((t, \tilde{t})\). Moreover, the homomorphic evaluation can be simply designed as GPV-FHE (see Section 3 for more details about our MIFHE scheme).

1.3. Paper Organization. First, we recall some notions, definitions, and facts in Section 2. Then, we propose our MIFHE scheme that satisfies IND-sID-CPA secure in Section 3. In the end, we conclude in Section 4.

2. Preliminaries

Let us start with the following notations that will be used throughout the study. We use the bold uppercase letters (e.g., \(A, B\)) to represent matrices. Similarly, the bold lowercase letters (e.g., \(a, b\)) represent column vectors. We use \(a_i\) to denote the \(i\) entry of \(a\) and \(a_{ij}\) to denote the \((i, j)\) entry of \(A\). \([A|B]\) is used to denote the concatenation of two matrices. Similarly, \((a, b)\) is used to denote the concatenation of two column vectors. Let \(\lambda\) denote the security parameter. We define \([n] = \{1, 2, \ldots, n\}\) for any positive integer \(n\). Let \(\negl(\lambda)\) denote a negligible function that grows slower than \(\lambda^{-c}\) for any constant \(c > 0\) and any sufficiently large value of \(\lambda\). An event occurs with overwhelming probability; i.e., it occurs with a probability of at least \(1 - \negl(\lambda)\).
2.1. Basic Notions

2.1.1. Approximations. Very recently, Peikert and Shiehian [17] suggested a simple method to indicate that the two sides in some “noisy equations” extensively used in lattice-based cryptography were approximately equal to an additive error. We will follow their notation using:

\[ x = y (\text{error } E). \]  

(2)

To indicate that \( x = y + e \) for some \( e \in [-E, E] \), the notation can be naturally expanded to the vector or matrix type using the infinite norm.

2.1.2. Tensor Product. For matrices \( A \in \mathbb{Z}^{m \times n}, B \in \mathbb{Z}^{s \times t} \), the tensor product \( A \otimes B \) is an \((ms \times nt)\) matrix, which consists of \((m \times n)\) blocks, whose \((i, j)\) block is \( a_{ij} \cdot B \).

In this work, we widely use the mixed-product property: for any matrices \( A, B, C, D \) with compatible dimensions, it holds that:

\[ (A \otimes B) \cdot (C \circ D) = (A \cdot C) \otimes (B \cdot D). \]  

(3)

2.2. Background on Lattices

2.2.1. Lattices. For matrix \( A \in \mathbb{Z}^{m \times n} \), we define the \( q\)-ary integer lattice in this way:

\[ \Lambda^+ (A) = \{ x \in \mathbb{Z}^m : Ax = 0 \mod q \}. \]  

(4)

For vector \( u \in \mathbb{Z}^n \), we now define the coset (or “shifted” lattice) in this way:

\[ \Lambda_u^+ (A) = \{ x \in \mathbb{Z}^m : Ax = u \mod q \}. \]  

(5)

2.2.2. LWE. The learning with errors (LWE) problem was first introduced by Regev [27] as an extension of the “learning parity with noise.” In this study, we define the decisional learning with errors (DLWE) problem that is equivalent to LWE for certain parameters as follows.

Definition 1. (Decisional Learning with Errors (DLWE)) For positive integers \( n, m, q \) and an error distribution \( \chi \) over \( \mathbb{Z} \), the DLWE \((n, m, q, \chi)\) problem is to distinguish with nonnegligible advantage between \((A, b = A^T \cdot t + e)\), where \( A \in \mathbb{Z}_q^{m \times n} \), \( t \in \mathbb{Z}_q^n \), \( e \sim \chi \), and \((A, b)\) sampled uniformly at random from \( \mathbb{Z}_q^{m \times n} \).

In this work, we rely on the tensor form of LWE denoted by TLWE and the matrix form of LWE denoted by MLWE, whose specific definitions are introduced below.

Definition 2. (The Matrix Form of Learning with Errors (MLWE)) For positive integers \( n, m, k, q \) and an error distribution \( \chi \) over \( \mathbb{Z} \), the MLWE \((n, m, k, q, \chi)\) problem is to distinguish with nonnegligible advantage between \((A, B = A^T \cdot T + E)\), where \( A \in \mathbb{Z}_q^{n \times m}, T \in \mathbb{Z}_q^{m \times k}, E \sim \chi \), and \((A, B)\) sampled uniformly at random from \( \mathbb{Z}_q^{n \times m} \).

Definition 3. (The Tensor Form of Learning with Errors (TLWE)) For positive integers \( n, m, k, t, q, \chi \) and an error distribution \( \chi \) over \( \mathbb{Z} \), the TLWE \((n, m, k, t, q, \chi)\) problem is to distinguish with nonnegligible advantage between \((A, B = I \otimes A^T \cdot T + E)\), where \( A \in \mathbb{Z}_q^{n \times m}, T \in \mathbb{Z}_q^{m \times k}, E \sim \chi \), and \((A, B)\) sampled uniformly at random from \( \mathbb{Z}_q^{n \times m \times k} \).

According to the standard mixed argument, we can get that MLWE is equivalent to DLWE with at most \( k \) factor loss in the distinguishing advantage, and TLWE is equivalent to MLWE with at most \( t^2 \) factor loss in the distinguishing advantage.

2.2.3. Lattice Trapdoor. We recall some cryptographic facts about trapdoor generation and preimage sampling algorithms with important properties [30]. Since all the details of implementation are not strictly necessary in this work, we ignore them. Note that there are improved algorithms but only within \( O(\log \lambda) \) factor [31].

Lemma 1 (see [30]). Let \( n = \text{poly}(\lambda), q \geq 3 \) be odd, and \( m = \lceil n \log q \rceil \). There is a probabilistic polynomial time (PPT) algorithm \( \text{TrapGen}(n, m, q) \) that outputs a pair \((A \in \mathbb{Z}_q^{n \times m}, T_A \in \mathbb{Z}^{m \times n})\), such that \( A \) is statistically close to a uniform matrix in \( \mathbb{Z}_q^{n \times m} \), and \( T_A \) is a basis for \( \Lambda^+ (A) \) satisfying \( \| T_A \| \leq O(\log q) \) and \( \| T_A \| \leq O(\sqrt{n \log q}) \) with overwhelming probability in \( \lambda \).

Lemma 2 (see [30]). Let \( n = \text{poly}(\lambda) \) be a positive integer, \( q \) be a prime, and \( m \geq 2n \log q \). Then, for all but a \( 2^{-n} \) fraction of all \( A \in \mathbb{Z}_q^{n \times m} \) and for any \( r \geq \sqrt{\log m} \), the distribution of the syndrome \( u = \text{As} \mod q \) is statistically close to uniform over \( \mathbb{Z}_q^m \), where \( u \in \mathbb{Z}_q^m \).

Lemma 3 (see [30]). Set \((A \in \mathbb{Z}_q^{n \times m}, T_A \in \mathbb{Z}_q^{m \times n}) \)---\( \text{TrapGen}(n, m, q) \)---from Lemma 1. Then, for a parameter \( r \geq \| T_A \| \cdot \omega(\sqrt{\log q}) \) and a uniformly random vector \( u \in \mathbb{R}^n \), there is a PPT algorithm, which outputs vector \( s \in \Lambda_u^+ (A) \) sampled from a statistically similar distribution to \( \mathbb{D}^\perp \Lambda_u^+ (A) \); thus, \( As = u \) whenever \( \Lambda_u^+ (A) \) is not empty.

2.2.4. Gadget Matrices and Bit Decomposition. We recall a useful notion of gadget matrix, which was first introduced in [31], to decompose vectors or matrices over \( \mathbb{Z}_q \) into short vectors or matrices over \( \mathbb{Z} \).
For integer \( q \), let \( \ell = \lceil \log q \rceil \). Gadget matrix \( g^T = (1, 2, 2^2, \ldots, 2^{\ell-1}) \) and bit decomposition function \( g^{-1}: \mathbb{Z}_q \rightarrow \{0,1\}^\ell \) are defined, which outputs a binary column vector that consists of the binary representation of its argument, such that \( g^T \cdot g^{-1}(a) = a \), for any \( a \in \mathbb{Z}_q^\ell \).

More generally, for any positive integer \( k \), \( G_k = I_k \otimes g^T \in \mathbb{Z}_q^{k \times \ell} \) is defined, where \( I_k \) denotes the \( k \)-dimensional identity matrix. For any \( t \), the general bit decomposition function \( g^{-1}: \mathbb{Z}_q^{k \times t} \rightarrow \{0,1\}^{k \times t} \) outputs a binary \((kt \times t)\) matrix (invoking \( g^{-1} \)), such that \( G \cdot g^{-1}(A) = A \), for \( A \in \mathbb{Z}_q^{k \times t} \). Additionally, we often write \( G^{-1}(A) = (G^T(A))^T \) for simplicity.

2.3. Multi-Identity Fully Homomorphic Encryption. We begin with the definition of the leveled multi-hop MIFHE, which is adapted and summarized from the definition of the single-hop MIFHE in [9], definition of the single-hop MKFHE in [10], and definition of the multi-hop MKFHE in [17]. Here, we require a bound \( \ell \) on the NAND circuit depth and a bound \( L \) on the number of identities in one evaluation, and we mainly focus on the bit encryption scheme and [17].

Now, a ciphertext is called a “fresh” ciphertext if it is generated by the encryption algorithm \( Enc \) defined below (i.e., it corresponds to a single identity), an “expanded” ciphertext if it is the output of expansion algorithm \( Expand \) (which relates to multiple identities), or an “evaluated” ciphertext if it is the output of homomorphic evaluation algorithm \( Eval \).

**Definition 4.** A leveled multi-hop multi-identity fully homomorphic encryption scheme consists of six PPT algorithms \( \{\text{Setup}, \text{Extract}, \text{Enc}, \text{Expand}, \text{Eval}, \text{and Dec}\} \) defined as follows:

(i) **Setup** \((1^\lambda, 1^d, 1^L)\): on inputting a security parameter \( \lambda \), a bound \( d \) on the NAND circuit depth, and a bound \( L \) on the number of identities involved in one evaluation, generate a master public key MPK and a master secret key MSK, and then output \((\text{MPK}, \text{MSK})\). Here, the security parameter \( \lambda \) also defines an identity space \( \mathcal{I} \).

(ii) **Extract** (MPK, MSK, id): on inputting the MPK, MSK, and identity \( id \in \mathcal{I} \), extract a user-specific secret key \( SK_{id} \), and output it.

(iii) **Enc** (MPK, id, \( \mu \in \{0,1\} \)): on inputting the MPK, identity \( id \in \mathcal{I} \), and bit \( \mu \in \{0,1\} \), output a “fresh” ciphertext \((id; \mu)\).

(iv) **Expand** (MPK, id_{k+1}, (id_1, id_2, \ldots, id_k; c)): on inputting the MPK, identity \( id_{k+1} \), and any (“fresh,” “expanded,” or “evaluated”) ciphertext \((id_1, id_2, \ldots, id_k; c)\) under \( k \) identities \((id_1, id_2, \ldots, id_k)\), compute and output an “expanded” ciphertext \((id_1, id_2, \ldots, id_k, id_{k+1}; c)\) under \((k+1)\) identities \((id_1, id_2, \ldots, id_k, id_{k+1})\).

(v) **Eval** (MPK, \( f, c_1, c_2, \ldots, c_N \)): on inputting MPK, an NAND circuit \( f \), and \( N \) ciphertexts \((c_1, c_2, \ldots, c_N)\), output an “evaluated” ciphertext \( c_f \).

(vi) **Dec** \((SK_{id_1}, SK_{id_2}, \ldots, SK_{id_k}, (id_1, id_2, \ldots, id_k; c))\): on inputting \( k \) secret keys \((SK_{id_1}, SK_{id_2}, \ldots, SK_{id_k})\), which correspond to identities \((id_1, id_2, \ldots, id_k)\) and any ciphertext \((id_1, id_2, \ldots, id_k; c)\), output a bit \( \mu \).

We underline that we will homomorphically evaluate any NAND circuit gate by gate as described in [17], which indicates that the evaluation is multi-hop as previous multi-key FHE schemes [8, 17].

2.3.1. Correctness. A leveled multi-hop MIFHE scheme is correct if it satisfies the following conditions. For all positive integers \( \lambda, d, L \), for every NAND circuit \( f \) of depth at most \( d \) with \( N \) input wires, for every function \( \pi: \{N\} \rightarrow \{L\} \) (which relates each input wire to a key pair), and for every \( \mu \in \{0,1\}^N \), the following experiment succeeds with overwhelming probability: \((\text{MPK}, \text{MSK}) \rightarrow \text{Setup}(1^\lambda, 1^d, 1^L), \text{generate identity key pairs (PK_{id_1}, SK_{id_1})}) \rightarrow \text{Extract}(\text{MPK}, \text{MSK}, id_i) \) for every \( i \in \{N\} \), generate ciphertext \( c_f \leftarrow \text{Enc}(\text{MPK}, id_i, \mu \in \{0,1\}) \) for every \( i \in \{N\} \), compute \( c_f \leftarrow \text{Eval}(\text{MPK}, f, c_1, c_2, \ldots, c_N) \) (may invoke algorithm \( \text{Expand} \)), and finally check whether \( \text{Dec}(\{SK_{id_i}\}_{i \in \{N\}}, c_f) = f(\mu_1, \mu_2, \ldots, \mu_N) \).

2.3.2. Compactness. A leveled multi-hop MIFHE scheme is compact if there exists a polynomial \( p(\cdot, \cdot, \cdot) \) such that \( |c_f| \leq p(\lambda, d, L) \) in the experiment from Definition 4. In other words, the length of \( c_f \) is independent of both \( f \) and \( N \) but can depend polynomially on \( \lambda, d, \) and \( L \).

2.3.3. Security. The security game of MIFHE is the same as that of IBE, but there is no reference to the expansion algorithm and evaluation algorithm because they are public and do not impact the security. In this study, we will mainly focus on the semantically secure under selective identity and chosen-plaintext attack (IND-sID-CPA) security game for MIFHE between a challenger \( \mathcal{C} \) and a PPT attacker \( \mathcal{A} \), which is defined as follows:

(i) Initial Stage. Attacker \( \mathcal{A} \) is given bound \( d \) on the NAND circuit depth and bound \( L \) on the number of identities and outputs target identity \( id^* \).

(ii) Setup. Challenger \( \mathcal{C} \) runs \( \text{Setup}(1^\lambda, 1^d, 1^L) \) to generate \((\text{MPK}, \text{MSK})\) and sends MPK to attacker \( \mathcal{A} \).

(iii) Query Stage 1. Adversary \( \mathcal{A} \) adaptively issues a query on any identity \( id \) such that \( id \neq id^* \). Challenger \( \mathcal{C} \) runs \( \text{Extract}(\text{MPK}, \text{MSK}, id) \) to obtain identity secret key \( sk_{id} \) that corresponds to \( id \) and sends \( sk_{id} \) back to \( \mathcal{A} \).

(iv) Challenge. Challenger \( \mathcal{C} \) selects a uniformly random bit \( \mu^* \leftarrow \{0,1\} \), computes a challenge ciphertext \( c^* \leftarrow \text{Enc}(\text{MPK}, id^*, \mu^*) \), and sends it to attacker \( \mathcal{A} \).

(v) Query Stage 2. Adversary \( \mathcal{A} \) issues additional adaptive identity secret key queries, and challenger \( \mathcal{C} \) responds as in query stage 1.
The advantage of the attacker in the above IND-sID-CPA security game is defined as $\Pr[\mu' = \mu^*] - 1/2$, where the probability is taken over the random bits used by all algorithms in the game.

**Definition 5.** A leveled multi-identity fully homomorphic encryption scheme is IND-sID-CPA secure if any PPT attacker has at most a negligible advantage in the IND-sID-CPA security game defined above.

### 3. Multi-Identity Fully Homomorphic Encryption

#### 3.1. MIFHE Scheme

In this section, we will describe the proposed MIFHE scheme. We present one more algorithm MIFHE.NAND to help understand MIFHE.Eval.

We parameterize the system by dimension $n = \text{poly}(\lambda)$, modulus $q$, and error distribution $\chi$ for the underlying LWE problem; we set $m = \lfloor 6n\log q \rfloor$, $r = O(\sqrt{\log q} \cdot \omega(\sqrt{\log n}))$, and $B = \Theta(r\sqrt{n})$. For the worst-case security, we set $\chi$ to be the standard discrete Gaussian distribution over $\mathbb{Z}$ with parameter $2\sqrt{n}$, which implies that the samples drawn from $\chi$ have magnitudes bounded by $E = \Theta(n)$ except with probability $2^{-n}$. Modulus $q$ is set in the following Section 3.2 based on the bound on the maximum depth of the supported circuit and the bound of the number of identities. The scheme is described as follows:

(i) **MIFHE.Setup** $(1^\lambda, 1^d, 1^L)$: On inputting security parameter $\lambda$, bound $d$ on the NAND circuit depth, and bound $L$ on the number of identities in one evaluation, do:

1. Run algorithm TrapGen ($n, m - 1, q$) to generate a uniformly random matrix $A_i \in \mathbb{Z}_q^{m \times (m-1)}$ with a short basis $T_{A_i} \in \mathbb{Z}_q^{(m-1)\times (m-1)}$ for $A_i^\perp(A_i)$ such that $\|T_{A_i}\| \leq O(\sqrt{\log q})$.
2. Choose a vector $a \sim \mathbb{Z}_q^n$ and set $A = [A_i, [a]] \in \mathbb{Z}_q^{nm}$.
3. Output MPK $= A = [A_i, [a]]$ as the master public key and output MSK $= T_{A_i}$ as the master secret key.

(ii) **MIFHE.Extract** $(\text{MPK}, \text{MSK}, id)$: On inputting MPK, MSK, and identity $id \in \mathcal{I}$, do:

1. If $(id, u, SK_{id}) \in \text{storage}$ is from a previous inquiry on identity $id$, then return $SK_{id}$.
   Otherwise, compute $u = H(id) \in \mathbb{Z}^n$, where $H$ is a hash function modeled as a random oracle.
2. Run SamplePre$(A_i, T_{A_i}, u - a, r)$ to output a vector $s \in \mathbb{Z}_q^{m-1}$ such that $A_i s = u - a$. Set user-specific secret key $SK_{id} = t = (s, 1)$, and store $(id, u, t)$ locally. Note that $At = u$ and $\|t\|_\infty \leq B$.
3. Output $SK_{id} = t$.

(iii) **MIFHE.Enc** $(\text{MPK}, id, \mu \in \{0, 1\})$: On inputting master public key MPK, identity $id \in \mathcal{I}$, and bit message $\mu \in \{0, 1\}$, do:

1. Set $u = H(id) \in \mathbb{Z}^n$ and compute $B = A - e_i^T \otimes u \in \mathbb{Z}_q^{nm}$, where $e_n$ is the $n$th standard unit (column) vector. (Remark: observe that $Bt = 0$).
2. Choose a uniformly random matrix $Q \sim \mathbb{Z}_q^{nm}$ and a discrete Gaussian matrix $W \sim \chi^{nm}$, and define the following:
   $$C = B^T Q + W + \mu G_m \in \mathbb{Z}_q^{nm}.$$

Note that $C$ is nicely a GPV-FHE ciphertext [26] encrypting $\mu$ under the secret key $t$. In particular,

$$t^T C = t^T W + \mu t^T G_m = \mu t^T G_m. \quad \text{error } \beta_C$$

(7)

(3) Choose a matrix $R \sim \mathbb{Z}_q^{n \times m}$ and a discrete Gaussian matrix $X \sim \chi^{m \times \ell}$, and define the following:

$$D = A^T R + X + \mu G_m \in \mathbb{Z}_q^{n \times m}.$$

(8)

Here, $D$ is regarded as a commitment to the message $\mu$ under commitment randomness $(R, X)$.

(4) Choose a matrix $S \sim \mathbb{Z}_q^{n \times m}$ and a discrete Gaussian matrix $Y \sim \chi^{m \times \ell}$, and define the following:

$$E = (I_m \otimes B^T)S + Y + R \otimes g \otimes e_m \in \mathbb{Z}_q^{n \times m \times \ell}.$$

(9)

Note that

$$\left(I_m \otimes t^T\right) \cdot E = \left(I_m \otimes t^T\right) \cdot (Y + R \otimes g \otimes e_m) \\ \approx R \otimes g. \quad \text{error } \beta_E$$

(10)

Therefore, $E$ is regarded as a sort of encryption of $R$ (the tensor product with $g$ corresponding to some bit decomposition appeared in expansion algorithm is vital to control the error growth), the former part of the commitment randomness used in $D$.

(5) Choose a uniformly random matrix $T \sim \mathbb{Z}_q^{n \times m}$ and a discrete Gaussian matrix $Z \sim \chi^{m \times \ell}$, and define the following:

$$F = (I_m \otimes B^T)T + Z + X \otimes e_m \in \mathbb{Z}_q^{n \times m \times \ell}.$$

(11)

Note that

$$\left(I_m \otimes t^T\right) \cdot F = \left(I_m \otimes t^T\right) \cdot (Z + X \otimes e_m) \approx X. \quad \text{error } \beta_F$$

(12)

Therefore, $F$ is regarded as a sort of encryption of $X$, the latter part of commitment randomness used in $D$.

(6) Output a “fresh” ciphertext $(id, C, D, E, F)$ to identity $id$. 
(iv) **MIFHE.Expand** \((\text{MPK}, \text{id}_{k+1}, (id_1, id_2, \ldots, id_k; C, D, E, F))\): on inputting MPK, identity \(id_{k+1}\), and ciphertext \((id_1, id_2, \ldots, id_k; C \in \mathbb{Z}_q^{m \times k+1}, D \in \mathbb{Z}_q^{m \times m}, E \in \mathbb{Z}_q^{m \times k\times m}, F \in \mathbb{Z}_q^{m \times k\times m})\) encrypting \(\mu\) under identities \((id_1, id_2, \ldots, id_k)\), do:

1. Set \(u_{k+1} = H(id_{k+1}).\)
2. We define the following:

\[
C' = \begin{pmatrix} C & U \\ 0 & D \end{pmatrix} e \in \mathbb{Z}_q^{m(k+1)\times m(k+1)}.
\]

where \(U\) is defined as follows:

\[
\begin{pmatrix} G_{n,m}^T (-u_{k+1}) \otimes I_{mk} \\ 0 \end{pmatrix} e \in \mathbb{Z}_q^{m\times m\times k\times m}.
\]

(3) We leave the commitment and its randomness unchanged: \(C' = D, R' = R\), and \(X' = X\).

(4) We define the following:

\[
E' = \begin{pmatrix} L_0 \otimes (I_{mk}) \\ 0 \end{pmatrix} e \in \mathbb{Z}_q^{m(k+1)\times m\times m}.
\]

(5) Similarly, we define the following:

\[
F' = \begin{pmatrix} L_0 \otimes (I_{mk}) \\ 0 \end{pmatrix} e \in \mathbb{Z}_q^{m\times m\times m\times m}.
\]

(6) Output \((id_1, id_2, \ldots, id_{k+1}; C', D', E', F')\) as the “expanded” ciphertext to identities \(id_1, id_2, \ldots, id_{k+1}\).

(v) **MIFHE.NAND** \( ((id_1, id_2, \ldots, id_k; C_1, D_1, E_1, F_1), (id_1, id_2, \ldots, id_k; C_2, D_2, E_2, F_2))\): on inputting two cipherscripts \((id_1, id_2, \ldots, id_k; C_1, D_1, E_1, F_1), (id_1, id_2, \ldots, id_k; C_2, D_2, E_2, F_2)\) that encrypt \(\mu_1, \mu_2\) under identities \((id_1, id_2, \ldots, id_k)\), do:

1. We define the following:

\[
C_{\text{NAND}} = G_{mk} - C_1 \cdot G_{mk}^T(C_2) e \in \mathbb{Z}_q^{m \times k \times m \times m}.
\]

(2) We define the following:

\[
D_{\text{NAND}} = G_m - D_1 \cdot G_m^T(D_2) e \in \mathbb{Z}_q^{m \times m \times m \times m}.
\]

(3) We define the following:

\[
E_{\text{NAND}} = \begin{pmatrix} (E_1 \cdot G_m^T(D_2) + (I_{mk} \otimes C_1) \\ G_{mk}^T(E_2) \end{pmatrix} e \in \mathbb{Z}_q^{m \times m \times m \times m}.
\]

(4) We define the following:

\[
F_{\text{NAND}} = \begin{pmatrix} (F_1 \cdot G_m^T(D_2) + (I_{mk} \otimes C_1) \\ G_{mk}^T(F_2) \end{pmatrix} e \in \mathbb{Z}_q^{m \times m \times m \times m}.
\]

(5) Finally, output \((id_1, id_2, \ldots, id_k; C_{\text{NAND}}, D_{\text{NAND}}, E_{\text{NAND}}, F_{\text{NAND}})\) as the “evaluated” NAND ciphertext.

(vi) **MIFHE.Eval** \((\text{MPK}, f, c_1, c_2, \ldots, c_N)\): on inputting MPK, NAND circuit \(f: [0,1]^N \rightarrow [0,1]\), and any \(N\) ciphertexts \((c_1, c_2, \ldots, c_N)\), compute homomorphically \(f\) over any \(N\) ciphertexts \((c_1, c_2, \ldots, c_N)\) gate by gate by invoking MIFHE.Expand and MIFHE.NAND, and output an “evaluated” ciphertext \(c_f\).

(vii) **MIFHE.Dec** \( ((SK_{id_1}, SK_{id_2}, \ldots, SK_{id_k}), (id_1, id_2, \ldots, id_k; C, D, E, F))\): on inputting the secret keys \((SK_{id_1}, SK_{id_1}, \ldots, SK_{id_k})\) and a ciphertext \((id_1, id_2, \ldots, id_k; C \in \mathbb{Z}_q^{m \times k \times m}, D \in \mathbb{Z}_q^{m \times m}, E \in \mathbb{Z}_q^{m \times k \times m \times m}, F \in \mathbb{Z}_q^{m \times k \times m \times m})\) under identities \((id_1, id_2, \ldots, id_k)\), let \(t \in \mathbb{Z}_q^{mk}\) be the (column) concatenation of the secret keys \((SK_{id_1}, SK_{id_1}, \ldots, SK_{id_k})\), and compute:

\[
t^TC \approx t^TG_{mk}: (\text{error } \beta_C).
\]

If \(\beta_C < q/4\), where \(q\) is set in the next section, we can recover \(\mu\) from the last term of vector \(\mu^T G_{mk}\): if this term is closer to 0, output 0; otherwise, output 1.

3.2. Analyzing the Noise Growth and Setting the Parameters.

Now, we provide the reasons for definitions and analyze the noise growth in MIFHE.Expand and MIFHE.NAND to easily set the parameters. We instantiate the parameters and ensure correctness of MIFHE.

First, as described in the previous section in the MIFHE.Expand algorithm, let us do the following analysis:

(1) We have the following:

\[
C' = \begin{pmatrix} C & U \\ 0 & D \end{pmatrix} e \in \mathbb{Z}_q^{m(k+1)\times m(k+1)}.
\]

Given \(t \in \mathbb{Z}_q^{mk}\) that is the (column) concatenation of the secret keys corresponding to identities \((id_1, id_2, \ldots, id_k)\), respectively, and a new secret key \(t_{k+1}\), which is the secret key of \(id_{k+1}\), we set \(t' = (t, t_{k+1})\) and then have the following:

\[
t'^T \cdot C' = (t^T \cdot C, t^T \cdot U + t_{k+1}^T \cdot D).
\]

Given \(t \in \mathbb{Z}_q^{mk}\) that is the (column) concatenation of the secret keys corresponding to identities \((id_1, id_2, \ldots, id_k)\), respectively, and a new secret key \(t_{k+1}\), which is the secret key of \(id_{k+1}\), we set \(t' = (t, t_{k+1})\) and then have the following:

\[
t'^T \cdot C' = (t^T \cdot C, t^T \cdot U + t_{k+1}^T \cdot D).
\]
which indicates that (7) holds. In general, the error implied in the "expanded" ciphertext $C'$ is as follows:

$$\beta_{C'} = \beta_C + n\ell\beta_E + m\text{BE}. \quad (24)$$

(2) This visibly satisfies equation (8), and the error implied in the "expanded" ciphertext $D'$ is as follows:

$$\beta_{D'} = \beta_D. \quad (25)$$

(3) We have the following:

$$E' = \left( I_m \otimes \left( I_{mk} \begin{pmatrix} 0 \\ o_{\text{monk}} \end{pmatrix} \right) \right) \cdot E \in \mathbb{Z}_q^{\ell(m^k + 1) \times \ell m t}. \quad (26)$$

It is obvious that:

$$(I_m \otimes t^T) \cdot E' = (I_m \otimes t^T) \cdot E = R \otimes g = R' \otimes g \text{ (error $\beta_E$).} \quad (27)$$

Thus, (10) is kept up to expansion.

$$t^T \cdot C_{\text{NAND}} = t^T \left( G_{mk} - C_1 \cdot G_{mk}^{-1}(C_2) \right) = t^T \cdot G_{mk} - (t^T \cdot C_1 \cdot G_{mk}^{-1}(C_2)) = t^T \cdot G_{mk} - \mu_1 t^T \cdot C_2 = t^T \cdot G_{mk} - \mu_1 (\mu_2 t^T \cdot G_{mk}) (\text{error $\beta_C$}) = \left( 1 - \mu_1 \mu_2 t^T \right) \cdot G_{mk},$$

which indicates that (7) holds. In total, the error implied in the NAND ciphertext $C_{\text{NAND}}$ is as follows:

$$\beta_{C_{\text{NAND}}} = m\ell t \cdot \beta_{C_1} + \beta_{C_2}. \quad (32)$$

(2) We have the following:

$$D_{\text{NAND}} = G_m - D_1 \cdot G_m^{-1}(D_2) \in \mathbb{Z}_q^{\ell m \times m t}, \quad (33)$$

and the commitment randomness is as follows:

$$R_{\text{NAND}} = -R_1 \cdot G_m^{-1}(D_2) - \mu_1 R_2, \quad (34)$$

$$X_{\text{NAND}} = -X_1 \cdot G_m^{-1}(D_2) - \mu_1 X_2.$$ By simply computing, we can see that (8) is preserved:

$$D_{\text{NAND}} = G_m - D_1 \cdot G_m^{-1}(D_2) = G_m \left( A^T R_1 + X_1 + \mu_1 G_m \right) \cdot G_m^{-1}(D_2) = G_m \left( A^T R_1 + X_1 \right) \cdot G_m^{-1}(D_2) - \mu_1 D_2 = G_m \left( A^T R_1 + X_1 \right) \cdot G_m^{-1}(D_2) - \mu_1 (A^T R_2 + X_2 + \mu_2 G_m) = A^T \cdot \left( -R_1 \cdot G_m^{-1}(D_2) - \mu_1 R_2 \right) + (X_1 \cdot G_m^{-1}(D_2) - \mu_1 X_2) + (1 - \mu_1 \mu_2) G_m = A^T \cdot R_{\text{NAND}} + X_{\text{NAND}} + (1 - \mu_1 \mu_2) G_m.
(3) We have the following:

\[
E_{\text{NAND}} = (E_1 \cdot G_m^{1}(D_2) + (I_{n} \otimes C_1) \cdot G_{mnk}^{1}(E_2)) \in \mathbb{Z}_q^{m \times k \times mT}.
\]

To see that (10) holds for \( E_{\text{NAND}} \), first note that:

\[
\begin{align*}
(\mu_1 \cdot G_m^{1}(D_2)) \cdot G_{mnk}^{1}(E_2) &= (\mu_1 \otimes \mathbf{g}) \cdot G_{mnk}^{1}(E_2) \\
&= (\mu_1 \mathbf{R}_2) \otimes \mathbf{g} \cdot (\text{error } \beta_{E_2}).
\end{align*}
\]

Finally, from the above, we have the following:

\[
\begin{align*}
(\mu_1 \cdot G_m^{1}(D_2)) \cdot G_{mnk}^{1}(E_2) &= (\mathbf{R}_1 \cdot G_m^{1}(D_2)) \otimes \mathbf{g} \cdot (\text{error } \beta_{E_2}) \\
&= (\mu_1 \mathbf{R}_2) \otimes \mathbf{g} \cdot (\text{error } \beta_{E_2}).
\end{align*}
\]

(4) We have the following:

\[
F_{\text{NAND}} = (F_1 \cdot G_m^{1}(D_2) + (I_{m} \otimes C_1) \cdot G_{m^T_{nk}}^{1}(F_2)) \in \mathbb{Z}_q^{m \times k \times mT}.
\]

To see that (12) holds for \( F_{\text{NAND}} \), first note that:

\[
\begin{align*}
(I_{m} \otimes t^T) \cdot (F_1 \cdot G_m^{1}(D_2)) &= (I_{m} \otimes (t^T \cdot C_1)) \cdot G_{m^T_{nk}}^{1}(F_2) \\
&= (I_{m} \otimes (t^T \cdot G_{nk})) \cdot G_{m^T_{nk}}^{1}(F_2) \\
&= (I_{m} \otimes (t^T \cdot \mathbf{g}^T)) \cdot G_{m^T_{nk}}^{1}(F_2) \\
&= (I_{m} \otimes t^T) \cdot (I_{m} \otimes \mathbf{g}^T) \cdot G_{m^T_{nk}}^{1}(F_2) \\
&= (I_{m} \otimes \mathbf{g}) \cdot (I_{m} \otimes \mathbf{g}) \cdot G_{m^T_{nk}}^{1}(F_2) \\
&= (I_{m} \otimes \mathbf{g}) \cdot (I_{m} \otimes \mathbf{g}) \cdot G_{m^T_{nk}}^{1}(F_2)
\end{align*}
\]

which indicates that (12) holds.

Then, as [17], we now instantiate the parameters by bounding the worst-case error growth when homomorphically computing a depth \( d \) NAND circuit for up to \( L \) distinct identities. For a ciphertext \((id_1, id_2, \ldots, id_k; C, D, E, F)\) with commitment randomness \((R, X)\), the max error is defined:

\[
\beta = \max (\beta_{C}, \beta_{E}, \beta_{F}, \beta_{E} \cdot \beta_{F}).
\]

With the bounds from the above, for any ciphertext with errors bounded by \( \beta \), its “expanded” ciphertext has a max error of at most \( \beta(\epsilon + m + 1) \). Similarly, when we homomorphically compute an NAND gate of two ciphertexts with errors bounded by \( \beta \), the result has a max error of at most \( \beta(\epsilon + m + 1) \). Thus, after computing any depth \( d \) NAND circuit on “fresh” ciphertexts under \( L \) distinct keys, the result has a max error of at most:

\[
\text{poly}(n, \ell, L)^{d+L}.
\]

Thus, we can set \( 4 \cdot \text{poly}(n, \ell, L)^{d+L} = q \) for the correctness of decryption. Recall that \( \ell = \Theta(\log q) = O(d + L); \) therefore, \( \text{poly}(n, \ell, L)^{d+L} = \text{poly}(n, d, L)^{d+L} \). Thus, the security of our scheme corresponds to a worst-case \( n \)-dimensional lattice problem with an approximation factor of \( \text{poly}(n, d, L)^{d+L} \).

Finally, the compactness requirement is satisfied because any ciphertext in our construction is bounded by \( \text{poly}(n, d, L) = \text{poly}(\lambda, d, L) \).

3.3. Security. Now, we prove that the proposed scheme MIFHE is IND-SID-CPA secure under the hardness of the DLWE assumption in the random oracle model.

Theorem 1. The multi-hop multi-identity fully homomorphic encryption scheme MIFHE, which was constructed in
Section 3.1, is IND-sID-CPA secure in the random oracle model assuming that the DLWE \( n \times \lambda \) assumption holds.

Proof. We prove the security of the proposed scheme MIFHE using a sequence of hybrid games. The first one of this game is the real IND-sID-CPA security game in Definition 5, and the last one is the ideal game, where the challenge ciphertext (except challenge identity \( id^* \)) is uniformly random and independent of the challenge bit \( \mu^* \). We proceed by considering a sequence of hybrid games as follows:

Game 0: This is the original game described in Definition 5, and it is IND-sID-CPA security. Recall that \( id^* \in \mathcal{I} \) is the target identity; that is, attacker \( \mathcal{A} \) plans to attack \( id^* \), and the challenge ciphertext is \( (id^* ; C^*; D^*; E^*; F^*) \) encrypting \( \mu^* \).

Game 1: In this game, we change the methods of generating master public key \( A \), answering hash (random oracle) queries and answering identity secret key queries as follows.

- Uniformly select at random a matrix \( A_1 \in \mathbb{Z}^{mx(m-1)} \) and a vector \( a \in \mathbb{Z}^n \), and set \( A = [A_1, a] \in \mathbb{Z}^{m \times n} \).
- Uniformly select at random a vector \( u_{id^*/} \in \mathbb{Z}^q \).
- When attacker \( \mathcal{A} \) issues a hash query on identity \( id \in \mathcal{I} \), do:
  - If \( id = id^* \), return \( u_{id^*/} \).
  - Otherwise, return \( u_{id} \).
- Otherwise, sample vector \( s_{id} \in \mathcal{D}_Z \), and compute \( u_{id} = A_1 s_{id} + a \), set \( t_{id} = (s_{id}, 1) \), and store \( id, u_{id}, t_{id} \) locally. Finally, return \( u_{id} \).
- When attacker \( \mathcal{A} \) issues an identity secret key query on identity \( id \in \mathcal{I} \), where \( id \neq id^* \), without loss of generality, we assume that \( \mathcal{A} \) has queried \( H \) on \( id \) and return \( t_{id} \), where \( (id, u_{id}, t_{id}) \in \text{store} \).

Game 2: this game is the same as Game 1 except that \( (C^*, E^*, F^*) \), which is a part of the challenge ciphertext, is selected as uniformly random independent elements in \( \mathbb{Z}^{m \times n} \times \mathbb{Z}^{m \times n} \times \mathbb{Z}_q \).

Game 3: this game is the same as Game 2, except that \( D^* \) is selected as a uniformly random element in \( \mathbb{Z}^{m \times n} \).

In fact, Game 3 is the ideal game.

We show the indistinguishability among all sequential hybrid games.

Lemma 4. Game 0 and Game 1 are statistically indistinguishable.

Proof. We show that Game 0 is statistically close to Game 1 by analyzing that the changes are undetectable by any attacker between them step by step using Lemmas 1–3.

First, note that while the former part of master public key \( A_1 \) is generated by running algorithm \( \text{TrapGen}(n, m = 1, q) \) (with a trapdoor \( T_{A_1} \)) in Game 0, \( A_1 \) is sampled from the uniform distribution over \( \mathbb{Z}_q^{mx(m-1)} \) in Game 1. By Lemma 1, \( A_1 \) in Game 0 is distributed statistically close to a uniform distribution over \( \mathbb{Z}_q^{mx(m-1)} \) as in Game 1.

Second, in regard to the simulation of hash query \( H \), we discuss two cases:

1. If \( id = id^* \), then return \( u_{id^*/} \), which was uniformly sampled from \( \mathbb{Z}_q^n \) at random. This perfectly simulates hash query \( H(id^*) \).
2. Otherwise, sample a Gaussian vector \( s_{id} \in \mathcal{D}_Z \), compute \( u_{id} = A_1 s_{id} + a \), and return \( u_{id} \). Here, \( u_{id} \) is distributed statistically close to the uniform distribution over \( \mathbb{Z}_q^n \) since \( A_1 s_{id} \) is distributed statistically close to uniform distribution over \( \mathbb{Z}_q^n \) by Lemma 2 and \( a \in \mathbb{Z}_q^n \).

Finally, note that the identity secret key of \( id \) is \( t_{id} = (s_{id}, 1) \), where \( s_{id} \) is generated by running algorithm \( \text{SamplePre}(A_1, T_{A_1}, u_{id} - a, r) \) in Game 0, and the identity secret key of \( id \) is \( t_{id} = (s_{id}, 1) \), where \( s_{id} \) is sampled from \( \mathcal{D}_Z \), such that \( u_{id} = A_1 s_{id} + a \) in Game 1. By Lemma 3, \( s_{id} \) in Game 0 is distributed statistically close to \( \mathcal{D}_Z \) such that \( A_1 s_{id} = u_{id} - a \). Thus, the identity secret key \( t_{id} \) in Game 0 is distributed statistically close to that in Game 1. Therefore, Game 0 and Game 1 are statistically indistinguishable.

Lemma 5. Game 1 and Game 2 are computationally indistinguishable.

Proof. The computational indistinguishability between Game 1 and Game 2 follows the assumed intractability of DLWE (and MLWE and TLWE). To show this behavior, we present a simulator \( \mathcal{S} \) that can draw samples and form \( (B, C^*, E^*, F^*) \); it simulates Game 1 when the samples are DLWE samples; it simulates Game 2 when they are uniformly random samples. \( \mathcal{S} \) proceeds as follows:

1. Draw sufficient samples, and form \( (B, C^*, E^*, F^*) \).
2. Uniformly select at random a vector \( u_{id} \in \mathbb{Z}_q^n \) and let \( A = B + e \in \mathbb{Z}_q^{m \times n} \).
3. The hash query and identity secret key query are identical to those in Game 1. Generate \( D^* \) exactly as in MIFHE.Enc.

Therefore, if \( (B, C^*, E^*, F^*) \) is sampled from DLWE, then \( \mathcal{S} \) perfectly simulates Game 1. In contrast, if \( (B, C^*, E^*, F^*) \) is uniformly random, then \( \mathcal{S} \) simulates perfectly Game 2previous.

Lemma 6. Game 2 and Game 3 are computationally indistinguishable.

Proof. Similar to the previous proof of Lemma 5, the computational indistinguishability between Game 2 and Game 3 follows the assumed intractability of DLWE (or MLWE). To show this effect, we present a simulator \( \mathcal{S} \) that can draw samples and form \( (A, D^*) \); it simulates Game 2 when the samples are DLWE samples; it simulates Game 3 when they are uniformly random samples. \( \mathcal{S} \) proceeds as follows:
(1) Draw sufficient samples, and form \((A, D^s)\). Uniformly choose at random a vector \(u_m \leftarrow \mathbb{Z}_q^n\), and let 
\[ B = A - e_k^* \otimes u_m \in \mathbb{Z}_q^{mn}. \]
(2) The hash query and identity secret key query are identical to those in Game 2. Uniformly choose random \((C^*, E^*, F^*)\).

Therefore, if \((A, D^s)\) is sampled from DLWE, then \(\mathcal{D}\) perfectly simulates Game 2. In contrast, if \((A, D^s)\) is uniformly random, then \(S\) perfectly simulates Game 3. \(\square\)

There is no information of message \(\mu^*\) in Game 3. Additionally, Game 0 and Game 3 are computationally indistinguishable by Lemmas 4–6. Therefore, if the DLWE is difficult, attacker \(\mathcal{D}\) only has negligible advantage, which completes the proof of the IND-sID-CPA security of MIFHE.

4. Conclusion and Open Problem

We present a multi-hop MIFHE scheme, which is IND-sID-CPA secure in the random oracle model under the standard LWE assumption. However, the proposed MIFHE scheme is only leveled homomorphic. Therefore, it is interesting to construct a nonleveled multi-hop MIFHE scheme (i.e., there is no a priori bound on the depth of the circuits) under standard assumptions such as LWE (without unfalsifiable iO or WPRF).

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 Nature Science Foundation of China (Nos. 11974096, 61972124, and U1705264) and the Zhejiang Provincial Natural Science Foundation of China (No. LY19F020019).

References

[1] R. Rivest, L. Adleman, and M. Deryouzou, On Data banks and Privacy Homomorphisms. Foundations of Secure Computation, pp. 169–180, Academic Press, NY, USA, 1978.
[2] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the forty-first annual ACM symposium on Theory of computing, pp. 169–178, Bethesda MD USA, May 2009.
[3] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, H. Gilbert, Ed., pp. 24–43, Monaco and Nice, France, May 2010.
[4] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 97–106, IEEE Computer Society, Palm Springs, CA, USA, October 2011.
[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, S. Goldwasser, Ed., pp. 309–325, MA, USA, January 2012.
[6] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in Proceedings of the Annual Cryptology Conference, R. Safavi-Naini and R. Canetti, Eds., pp. 868–886, Santa Barbara, CA, USA, August 2012.
[7] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in Proceedings of the Annual Cryptology Conference, J. A. Canetti and R. Garay, Eds., pp. 75–92, Santa Barbara, CA, USA, August 2013.
[8] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multipartity computation on the cloud via multik ey fully homomorphic encryption," in Proceedings of the forty-fourth annual ACM symposium on Theory of computing, pp. 1219–1234, NY, USA, May 2012.
[9] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," in Proceedings of the Annual Cryptology Conference, R. Gennaro and M. Robshaw, Eds., pp. 630–656, Santa Barbara, CA, USA, August 2015.
[10] P. Mukherjee and D. Wichs, "Two round multipartity computation via multi-key FHE," Edited by M. Fischlin and J.-S. Coron, Eds., in Proceedings, Part II, of the 35th Annual International Conference on Advances in Cryptology, vol. 9665, pp. 735–763, Vienna, Austria, May 2016.
[11] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in Proceedings of the Annual Cryptology Conference, pp. 190–213, Santa Barbara, CA, USA, August 2016.
[12] L. Chen, Z. Zhang, and X. Wang, "Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension," in Proceedings of the Theory of Cryptography Conference, pp. 597–627, Baltimore, USA, November 2017.
[13] H. Chen, I. Chillotti, and Y. Song, "Multi-key homomorphic encryption from TFHE," in Proceedings of the Advances in Cryptology-ASIACRYPT, pp. 446–472, Springer, Kobe, Japan, December 2019.
[14] P. Ananth, A. Jain, Z. Jin, and G. Malavolta, "Multi-key fully-homomorphic encryption in the plain model," in Proceedings of the Theory of Cryptography Conference, pp. 28–57, Durham, NC, USA, December 2020.
[15] T. Zhou, Z. Zhang, L. Chen, X. Che, W. Liu, and X. Yang, "Multi-key fully homomorphic encryption scheme with compact ciphertext," 2021, Available at:http://eprint.iacr.org/2021/1131.
[16] C. Biswas and R. Dutta, "Secure and efficient multi-key FHE scheme supporting multi-bit messages from LWE preserving non-interactive decryption," 2021, Available at:http://eprint.iacr.org/2021/1431.
[17] C. Peikert and S. Shiehian, "Multi-key FHE from LWE, revisited," in Proceedings of the Theory of Cryptography Conference, pp. 217–238, Beijing, China, November 2016.
[18] D. Naccache, "Is theoretical cryptography any good in practice? 2010, http://www.iacr.org/workshops/ches/ches2010.
[19] M. Clear and C. McGoldrick, "Bootstrappable identity-based fully homomorphic encryption," in Proceedings of the International Conference on Cryptology and Network Security, pp. 1–19, Springer, Heraklion, Greece, December 2014.
[20] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, “Candidate indistinguishability obfuscation and functional encryption for all circuits,” *FOCS*, IEEE Computer Society Press, pp. 40–49, December 2013.

[21] R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan, “Chosen-ciphertext secure fully homomorphic encryption,” in *Proceedings of the IACR International Workshop on Public Key Cryptography*, pp. 213–240, Amsterdam, The Netherlands, March, 2017.

[22] T. Pal and R. Dutta, “Chosen-ciphertext secure multi-identity and multi-attribute pure FHE,” in *Proceedings of the Book Cryptology and Network Security*, pp. 387–408, Edinburgh, United Kingdom, May 2020.

[23] T. Shen, F. Wang, K. Chen, Z. Shen, and R. zhang, “Compressible multikey and multi-identity fully homomorphic encryption,” *Security and Communication Networks*, vol. 2021, Article ID 6619476, 14 pages, 2021.

[24] T. Shen, F. Wang, K. Chen, K. Wang, and B. Li, “Efficient leveled (multi) identity-based fully homomorphic encryption schemes,” *IEEE Access*, vol. 7, Article ID 79310, 2019.

[25] F. Wang and K. Wang, “Fully homomorphic encryption with auxiliary inputs,” in *Proceedings of the International Conference on Information Security and Cryptology*, D. Lin, M. Yung, and J. Zhou, Eds., pp. 220–238, Beijing, China, December 2014.

[26] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, “Leveled fully homomorphic signatures from standard lattices,” in *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pp. 469–477, Portland Oregon USA, June 2015.

[27] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in *Proceedings of the thirty-seventh annual ACM symposium on Theory of Computing*, pp. 84–93, Baltimore MD USA, May 2005.

[28] C. Peikert, “Public key cryptosystems from the worst-case shortest vector problem,” in *Proceedings of the forty-first annual ACM symposium on Theory of Computing*, pp. 333–342, Bethesda MD USA, May 2009.

[29] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, “Classical hardness of learning with errors,” in *Proceedings of the forty-fourth annual ACM symposium on Theory of Computing*, pp. 575–584, Palo Alto California USA, June 2013.

[30] C. Gentry, C. Peikert, and V. Vaikuntanathan, ”Trapdoors for hard lattices and new cryptographic constructions,” in *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, pp. 197–206, Victoria British Columbia Canada, May 2008.

[31] D. Micciancio and C. Peikert, ”Trapdoors for lattices: simpler, tighter, faster, smaller,” in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, D. Pointcheval and T. Johansson, Eds., pp. 700–718, Cambridge, United Kingdom, April 2012.