Hindawi
Mobile Information Systems
Volume 2020, Article ID 8862920, 10 pages
https://doi.org/10.1155/2020/8862920

Research Article

Achieving Message-Encapsulated Leveled FHE for IoT Privacy Protection

Weiping Ouyang,[1] Chunguang Ma,[1,2] Guoyin Zhang,[1] and Keming Diao[1]

[1]*Harbin Engineering University, Harbin, China*
[2]*Shandong University of Science and Technology, Qingdao, China*

Correspondence should be addressed to Weiping Ouyang; ouyangweiping@hrbeu.edu.cn

Received 15 April 2020; Revised 15 July 2020; Accepted 16 September 2020; Published 14 October 2020

Academic Editor: Kathiravan Srinivasan

The rapid development of the Internet of Things has made the issue of privacy protection even more concerning. Privacy protection has affected the large-scale application of the Internet of Things. Fully Homomorphic Encryption (FHE) is a newly emerging public key encryption scheme, which can be used to prevent information leakage. It allows performing arbitrary algebraic operations on data which are encrypted, such that the operation performed on the ciphertext is directly transformed into the corresponding plaintext. Recently, overwhelming majority of FHE schemes are confined to single-bit encryption, whereas how to achieve a multibit FHE scheme is still an open problem. This problem is partially (rather than fully) solved by Hiromasa-Abe-Okamoto (PKC′15), who proposed a packed message FHE scheme which only supports decryption in a bit-by-bit manner. Followed by that, Li-Ma-Morais-Du (Inscrypt′16) proposed a multibit FHE scheme which can decrypt the ciphertext at one time, but their scheme is based on dual LWE assumption. Armed with the abovementioned two schemes, in this paper, we propose an efficient packed message FHE that supports the decryption in two ways: single-bit decryption and one-time decryption.

1. Introduction

In recent years, the Internet of Things (IoT) has become an attractive system paradigm to drive a substantive leap on goods and services and has been widely used in intelligent transportation, intelligent power grid, environmental monitoring and perception, intelligent home appliances, and other fields. It covers traditional equipment to general household equipment, which brings more efficiency and convenience to the users. Because many of the data transmitted in the Internet of Things are confidential information or personal privacy information, it usually needs to be encrypted first. With more and more encrypted data stored on the server, it is very frequent for us to retrieve and process these data. Although there are some algorithms for retrieving encrypted data, they are only suitable for small-scale data, and the cost is too high. The encrypted data retrieval method based on the Fully Homomorphic Encryption (FHE) can solve this problem. By directly retrieving the encrypted data, it not only ensures that the retrieved data will not be analyzed, but also carries out homomorphic operation on the retrieved data without changing the sequence of the corresponding plaintext. It can not only protect the user's data security but also improve the retrieval efficiency. Since the first introduction of Gentry in 2009, the construction and optimization of the Fully Homomorphic Encryption scheme have been paid special attention by researchers. However, most of the existing Fully Homomorphic Encryption schemes only allow cryptographic calculations for a single bit, and the efficiency is not satisfactory. Although the cascading (or simple combination) approach can be used to implement message-encapsulated calculations, the performance of such a simple message-encapsulated FHE is not ideal.

In an application scenario, in many cases, it is necessary to calculate data of multiple bits at a time, and thus, constructing an efficient Message-encapsulation Fully Homomorphic encryption becomes an urgent requirement. At present, the research in this area has made initial progress [1, 2], which has increased the efficiency of FHE to a certain extent, but comprehensively, its efficiency still needs to be improved. Specifically, the following are considered:

(1) Brakerski's scheme [1] is constructed on the basis of the Brakerski's [3] scheme and is a typical representative of the second generation of FHE. But, the latter scheme needs to implement homomorphic calculations by calculating the evaluation key, which increases the computational cost.

(2) Hiromasa-Abe-Okamoto (HAO) [2] is based on the GSW [4] scheme and is a typical representative of the third generation of FHE. HAO constructs a message-encapsulation FHE scheme in the form of encapsulated messages, but it cannot implement one-time decryption and only decrypts the ciphertext bit-by-bit, so the scheme is still very inefficient.

An important question arises: Besides those mentioned above, is it possible to design an efficient method to decrypt the ciphertext of the message-encapsulation GSW-FHE scheme at one time?

Li et al. [5] used dual Regev [6] to construct a public key with multiple instances of the small short integer solution (SIS). Inspired by this work, we will construct public keys with multiple instances of LWEs (Learning with errors), and this constructs a Message-Encapsulation FHE scheme that can be decrypted at one time.

1.1. Our Contribution.

1.1. Our Contribution. Firstly, the public key of the Message-encapsulation Fully Homomorphic Encryption scheme of Hiromasa et al. [2] is as follows:

$$(\mathbf{B} := \mathbf{A} \cdot \mathbf{T} + \mathbf{E}(\bmod q) \mid \mathbf{A}) \in \mathbb{Z}_q^{m\times t} \times \mathbb{Z}_q^{m\times n}. \quad (1)$$

Among them are the secret matrix $\mathbf{T} \longleftarrow \mathbb{Z}_q^{n\times t}$ and the noise matrix $\mathbf{E} \longleftarrow \chi^{m\times t}$. Then, the plaintext message is encapsulated in a matrix, and the public key of the abovementioned form is used to encrypt the message. However, the obtained ciphertext matrix cannot recover all the plaintext bits at one time, but can only be decrypted bit-by-bit.

Secondly, we notice that the public key matrix of the message-encapsulated fully homomorphic encryption scheme constructed by Li et al. [5] is as follows:

$$\left(\mathbf{A} \cdot \mathbf{e}_1, \ldots, \mathbf{A} \cdot \mathbf{e}_t \mid \mathbf{A}\right) \in \mathbb{Z}_q^{m\times t} \times \mathbb{Z}_q^{m\times n}. \quad (2)$$

Among them, there is $\mathbf{e}_1, \ldots, \mathbf{A} \cdot \mathbf{e}_t \longleftarrow \chi^{n\times 1}$. Although Li et al.'s scheme [5] supports bit-by-bit encryption and one-time decryption, the scheme relies on the minimum integer solution hypothesis (see detailed analysis in [7]), and its parameter size depends on $m\,(m \geq n \log q)$ instead of causing the size of the evaluation key and the ciphertext to be too large.

Based on the abovementioned observations, in this paper, we construct a public key matrix first with multiple LWE instances. Different from the typical FHE scheme [3, 4, 8] and follow-up works [9–13], its public key matrix contains only one LWE instance. Then, using the new public key, we construct a message-encapsulation GSW-class FEH scheme (MFHE). We give an overview of the scheme in the following:

(1) Firstly, we use a new public key matrix with multiple LWE instances as follows:

$$\mathbf{A}' = \left[\mathbf{b}_1, \ldots, \mathbf{b}_t \mid \mathbf{A}\right] \in \mathbb{Z}_q^{m\times(n+t)}. \quad (3)$$

Among them, $\mathbf{b}_1 = \mathbf{A} \cdot \mathbf{t}_i + e_i\,(\bmod q)$ and $i \in [t]$ is an LWE instance. This is significantly different from existing message-encapsulation PKE schemes (for example, [14, 15]) and message-encapsulation FHE schemes (for example, [1, 2]) and is also the fundamental difference between other schemes and the FHE scheme constructed in this paper. Private keys corresponding to the public key $[\mathbf{b}_1, \ldots, \mathbf{b}_t]|\mathbf{A}$ is shaped as follows:

$$\mathbf{sk}_i := \left[0, \ldots, 1, \ldots, 0 \,\middle|\, \mathbf{t}_i\right] \in \mathbb{Z}_q^{1\times(n+t)},\ i \in [t]. \quad (4)$$

(2) Next, we use the public key matrix $\mathbf{A}'$ we constructed to encrypt multibit messages. The difference is that we use the message-encapsulation method of Li et al. [5] and Hiromasa et al. [2] to embed multibit messages into the plaintext of a diagonal matrix. That is,

$$\mathbf{M} := \operatorname{diag}\left(m_1, \ldots, m_t \,\middle|\, 1, \ldots, 1\right) \in \mathbb{Z}_q^{(n+t)\times(n+t)}, \quad (5)$$

and while constructing a private key matrix with private keys,

$$\mathbf{S} := \left[\mathbf{E} \,\middle|\, \begin{pmatrix} t_1 \\ \vdots \\ t_t \end{pmatrix}\right] \in \mathbb{Z}_q^{(n+t)\times(n+t)}. \quad (6)$$

$\mathbf{E}\,(n \times n)$ is the identity matrix, and we can get

$$\mathbf{S} \cdot \mathbf{M} := \left[\operatorname{diag}\left(m_1, \ldots, m_t\right) \,\middle|\, \begin{pmatrix} t_1 \\ \vdots \\ t_t \end{pmatrix}\right]. \quad (7)$$

Finally, using the matrix $\mathbf{W} := [\operatorname{diag}(\lfloor (q/2) \rfloor, \ldots, \lfloor (q/2) \rfloor) \mid 0]$ we constructed, calculation of $\mathbf{SM} \cdot GG - 1(W)$ can directly recover the message vector $(m_1, \ldots, m_t)$. See Section 4 for a detailed analysis.

1.2. Organization and Structure of the Paper. The rest of this paper is organized as follows. In Section 2, the definitions and symbols used in this paper are introduced. In Section 3, we review the scheme of Gentry-Sahai-Waters et al. In Section 4, we introduce the Message-encapsulation FHE (MFHE) scheme we constructed. Finally, we give a summary of the full paper in Chapter 5.

2. Preliminaries

In this section, we give the preparatory knowledge needed, including definitions and lemmas.

2.1. Symbols. For $n \in \mathbb{N}$, we use $[n]$ to represent aggregation $\{1, \ldots, n\}$. For a real number $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ to represent the largest integer that is not greater than x, $\lfloor x \rceil := \lfloor x + (1/2) \rfloor$ to represent the nearest integer to x. We represent vectors in bold lowercase letters, for example, $\mathbf{x}$, and the matrix in bold uppercase letters, for example, $\mathbf{A}$. In addition, we use $\mathbf{A}_{i,j}$ to represent elements in $\mathbf{A}_{i,j}$ from row i and column j. We use ":=" to indicate the assignment. It is worth noting that we use the definition of computationally indistinguishable and statistics indistinguishable and they are represented by $\approx_c$ and $\approx_s$. In addition to this, we also define $\|\mathbf{v}\|_\infty = \max\{|v_1|, \ldots, |v_n|\}$ and $\|\mathbf{R}\| = \max_i \|\mathbf{r}_i\|$. For convenience, we use $\|v\|$ to represent its l_2 norm.

We need to use the following variant of the Left-over Hash Lemma (LHL) [16].

Lemma 1 *(Matrix-Vector LHL). Let* $\lambda \in \mathbb{Z}, n, q \in \mathbb{N}, m \geq n \log q + 2\lambda, \mathbf{r} \xleftarrow{R} \{0,1\}^m$ *and* $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$. *We select a uniform random matrix* $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{m \times n}$, *and then, the statistical distance of the distribution* $(\mathbf{A}, \mathbf{A}^T \mathbf{r})$ *and* $(\mathbf{A}, \mathbf{y})$ *is as follows:*

$$\Delta\left(\left(\mathbf{A}, \mathbf{A}^T \cdot \mathbf{r}\right), (\mathbf{A}, \mathbf{y})\right) \leq 2^{-\lambda}. \tag{8}$$

2.2. Learning with Errors (LWEs). LWEs is the main computational assumption that cryptosystems and our variants rely on.

Definition 1 (LWE Distribution). For safety parameters, let $n = n(\lambda)$ and $m = m(\lambda)$ be integers, let $\chi = \chi(\lambda)$ be the $\mathbb{Z}$ error distribution with the bound of $B = B(\lambda)$, and let $q = q(\lambda) \geq 2$ be an integer modulo of any polynomial $p = p(\lambda)$ that meets $q \geq 2^p \cdot B$. Then, we select a vector $\mathbf{s} \in \mathbb{Z}_q^{n \times 1}$ and call it a secret, the LWE distribution $\mathscr{A}_{s,\chi}$ in $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is selected uniformly and randomly, and we select $\mathbf{e} \longleftarrow \chi^{m \times 1}$ and output $(\mathbf{A}, \mathbf{b} = A \cdot s + e (\bmod q))$.

There are two kinds of the LWE hypothesis: the search-LWE and the decision-LWE. The decision-LWE is defined as follows:

Definition 2 (Decision-LWE$_{n,q,\chi,m}$). Assume an independent selected $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times 1}$, which is selected according to one of the following distributions: (1) for $\mathscr{A}_{s,\chi}$ from a uniform and random $\mathbf{s} \in \mathbb{Z}_q^n$ (i.e., $\{(\mathbf{A}, \mathbf{b})$: $\mathbf{A} \longleftarrow \mathbb{Z}_q^{m \times n}, \mathbf{s} \longleftarrow \mathbb{Z}_q^{n \times 1}, \mathbf{e} \longleftarrow \chi^{m \times 1}, \mathbf{b} = \mathbf{A} \cdot s + e (\bmod q)\}$) or (2) uniform distribution (i.e., $\{(\mathbf{A}, \mathbf{b})$: $\mathbf{A} \longleftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{b} \longleftarrow \mathbb{Z}_q^{m \times 1}\}$). The two distributions mentioned above are computable indistinguishable.

Note 1. Regev and others [6, 17–19] introduce the convention between the approximate shortest vector problem (for appropriate parameters) in the LWE hypothesis. We have omitted the lemma of the results of these schemes; see [6, 17–19] for details.

2.3. Discrete Gauss. In our structure, we need to analyze the behavior of choosing the wrong element from the Gaussian distribution.

Definition 3 (B Bounded [3]). A distribution $\chi = \chi(\lambda)$ on an integer if the following exists:

$$\Pr_{x \xleftarrow{\$} \chi} [|x| \geq B] \leq 2^{-\widetilde{\Omega}(n)}, \tag{9}$$

and then, it is called B-bound (represented as $|\chi| \leq B$).

For the analysis of our scheme, the vector selected from the Gaussian distribution needs to have a certain bound on its norm.

Lemma 2 *(See [20]). 1. For* $\forall k > 0, \Pr[|e| > k \cdot \sigma, e \longleftarrow D_\sigma^1] \leq 2 \cdot \exp(-(k^2/2))$; *2. for* $\forall k > 0$, *there is* $\Pr[\|\mathbf{e}\| > k \cdot \sigma \cdot \sqrt{m}\, \mathbf{e} \longleftarrow D_\sigma^m] \leq k^m \cdot \exp((m/2) \cdot (1 - k^2))$ *Therefore, in this paper, we set* $|e| \leq B$ *and* $\|e\| \leq 2\sqrt{m}B$.

In this paper, we assume $\sigma \geq 2\sqrt{n}$. *So, if* $\mathbf{e} \longleftarrow D_\sigma^m$, *then on average,* $\|e\| \approx \sqrt{m} \cdot \sigma$. *It can be known from Lemma 2.2 (2) that there is a high possibility that* $\|\mathbf{e}\| \leq 2\sigma\sqrt{m}$. *Therefore, in this paper, we set* $|e| \leq B$ *and* $\|\mathbf{e}\| \leq 2\sqrt{m}B$.

2.4. Leveled Fully Homomorphic Encryption. In public-key cryptography, the cipher keeps a public key and encrypts the message in order that the corresponding private key holder can recover the original plaintext message.

Definition 4 (See [21]). Let a fixed function $L = L(\lambda)$ be the level of Fully Homomorphic Encryption. For a kind of circuit $\{\mathscr{C}_\lambda\}_{\lambda \in N}$, the L-FHE scheme includes four Probabilistic Polynomial Times (PPTs), and the algorithm is as follows:

$$(\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}). \tag{10}$$

The key generation algorithm (KeyGen) is a randomization algorithm that inputs security parameters 1^λ and outputs public keys (pk) and private keys (sk)

The encryption algorithm Enc is a randomization algorithm that inputs a public key (pk) and a message $m \in \{0,1\}^*$ and outputs a ciphertext c

The decryption algorithm Dec is a deterministic algorithm that inputs the private key sk and ciphertext and outputs the decrypted message $m \in \{0,1\}^*$

The homomorphic algorithm Eval inputs a public key pk, a circuit $C \in \mathscr{C}_\lambda$, and a sequence of ciphertexts $c_1, \ldots, c_{\ell(\lambda)}$, here let $\ell(\lambda)$ be a polynomial related to λ the and outputs the computed ciphertext c^*

The correctness requirements are as follows:

For arbitrary $\lambda, m \in \{0,1\}^*$ and (pk, sk) output by $\text{KeyGen}(1^\lambda)$, we have

$$m = \mathrm{Dec}(\mathrm{sk}, (\mathrm{Enc}(\mathrm{pk}, m))). \tag{11}$$

For arbitrary λ, arbitrary $m_1, \ldots, m_l \in \{0,1\}^*$, and $C \in \mathscr{C}_\lambda$, we have

$$\mathscr{C}(m_1, \ldots, m_\ell) = \mathrm{Dec}(\mathrm{sk}, (\mathrm{Eval}(\mathrm{pk}, (C, \mathrm{Enc}(pk, m_1), \ldots, \mathrm{Enc}(pk, m_\ell))))). \tag{12}$$

Definition 5 (CPA Security [21]). One FHE scheme is indistinguishable from the choice of plaintext attack (IND – CPA): the condition that security needs to be satisfied is that for any PPT adversary $\mathscr{A}$, the following probabilities related to are negligible:

$$\begin{aligned} &|\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{Enc}(\mathrm{pk}, m_0)) = 1], \\ &-\Pr[\mathscr{A}(\mathrm{pk}, \mathrm{Enc}(\mathrm{pk}, m_1)) = 1]| = \mathrm{negl}(\lambda). \end{aligned} \tag{13}$$

Among them, $(\mathrm{pk}, \mathrm{sk}) \longleftarrow \mathrm{KeyGen}(1^\lambda)$ and $m_0 \cdot m_1$ is arbitrarily selected from the plaintext space by the adversary.

The security definition of a message-encapsulation GSW (MFHE) is the same as GSW for a single bit. Because in public key settings, the security of single message encryption implies the security of multiple message encryption. See section 11 in [22] for more details.

Definition 6 (Compactness [21]). For a class of loops $\{\mathbb{C}_k\}_{k\in\mathbb{N}}$, if there is a polynomial $\alpha = \alpha(\lambda)$ such that the length of output ciphertext of Eval is at most α, then an L Fully Homomorphic Encryption is compact (if it is nontrivial, then for all λ, some $C \in \{\mathbb{C}\}_\lambda$, and we have $\alpha(\lambda) \le |C|$).

2.5. Basic Tools. Let us review some of the basic tools proposed by Brakerski and Vaikuntanathan [23] and Gentry et al. [4]. We fix $q, m \in \mathbb{N}$. Let $l = \lfloor \log(q) \rfloor + 1$, and therefore, $2^{l-1} \le q < 2^l$ and $N = m \cdot l$.

Definition 7 (See [24, 25]). The algorithm BitComp enters a vector $\mathbf{v} \in \mathbb{Z}_q^m$ and outputs an N-dimensional vector $(v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1})^T \in \{0,1\}^N$ where $v_{i,j}$ is the j bit in the binary representation of v_i (sorted by minimum impact to maximum impact). In other words,

$$v_i = \sum_{j=0}^{l-1} 2^j v_{i,j}. \tag{14}$$

Definition 8 (See [24, 25]). Algorithm enters a vector

$$\mathbf{v} = \left(v_{1,0}, \ldots, v_{1,l-1}, \ldots, v_{m,0}, \ldots, v_{m,l-1}\right)^T \in_q^N \tag{15}$$

and output $(\sum_{j=0}^{l-1} 2^j, \ldots, v_{1,j}, \ldots, \sum_{j=0}^{l-1} 2^{j v_{m,j}})^T \in_q^m$.

Note that the input vector $\mathbf{v}$ does not need to be binary and any of the input vector algorithms in $\mathbb{Z}^N$ are already defined.

Definition 9 (See [24, 25]). The algorithm Flatten enters a vector $\mathbf{v} \in \mathbb{Z}_q^N$ and outputs an N-dimension binary vector (i.e., an element from $0, 1^N$) defined as

$$\mathrm{Flatten}(\mathbf{v}) = \mathrm{BitDecomp}\left(\mathrm{BitDecomp}^{-1}(\mathbf{v})\right). \tag{16}$$

Definition 10 (See [24, 25]). The algorithm PoweOftwo enters an m-dimension vector $\mathbf{v} \in \mathbb{Z}_q^N$ and outputs an N-dimension vector in $\mathbb{Z}_q^N$. The output is as follows:

$$\left(v_1, 2v_1, \ldots, 2^{l-1} v_1, \ldots, v_m, 2v_m, \ldots, 2^{l-1} v_m\right)^T. \tag{17}$$

Lemma 3 *(See [26]). For any $N \ge m\lfloor \log q \rfloor$, there is a fixed effective computable matrix $\mathbf{G} \in \mathbb{Z}_q^{m\times N}$ and a valid computable deterministic “short-image” function $\mathbf{G}^{-1}(\cdot)$ that meets the following conditions. For arbitrary m', we enter a matrix $\mathbf{M} \in \mathbb{Z}_q^{m\times m'}$ and the inverse function $\mathbf{G}^{-1}(\mathbf{M})$ outputs a matrix $\mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{N\times m'}$ so that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$.*

Note 2. In fact, we can also express the abovementioned definitions and results as follows using the language of $\mathbf{G}$ and $\mathbf{G}^{-1}$. Micciancio and Peikert’s [26] matrix $\mathbf{G}$ can be expressed as $\mathbf{G} = \mathbf{I}_m \otimes \in \mathbb{Z}_q^{m\times N}$, where $\mathbf{g} = (1, 2, 4, \ldots, 2^{l-1})^T$. For $\mathbf{v} \in \mathbb{Z}_q^m$, there is $(\mathbf{v}) = \mathbf{v}^T\mathbf{G}$. For $\mathbf{v} \in \mathbb{Z}_q^N$, there is $\mathrm{BitDecomp}^{-1}(\mathbf{v}) = \mathbf{G}\mathbf{v}$. For $\mathbf{a} \in \mathbb{Z}_q^m$, the algorithm $\mathrm{BitDecomp}(\mathbf{a})$ is renamed as $\mathbf{G}^{-1}(\mathbf{a})$. For $\mathbf{v} \in \mathbb{Z}_q^m$, there is $\mathrm{Power\,Of\,two}(\mathbf{v}) = \mathbf{v}^T\mathbf{G}$. For $\mathbf{v} \in \mathbb{Z}_q^N$, there is $\mathrm{BitDecomp}^{-1}(\mathbf{v}) = \mathbf{G}\mathbf{v}$. For $\mathbf{a} \in \mathbb{Z}_q^m$, the algorithm $\mathrm{BitDecomp}(a)$ is renamed as $\mathbf{G}^{-1}(\mathbf{a})$.

3. Gentry–Sahai–Waters (GSW) Scheme

Before our work, we first review the GSW scheme and, then, summarize the safety of the scheme of Gentry et al. [4].

We review the algorithms which make up the GSW scheme [4]. These algorithms were originally defined based on functions BitDecomp, $\mathrm{BitDecomp}^{-1}$, and Flatten, but the ideas from [19, 27] borrowed into this paper are defined using tool matrix $\mathbf{G}$. Let λ be the security parameter and L be the number of levels of homomorphic encryption.

GSW.Setup$(1^\lambda, 1^L)$:

(1) Select a module q of bit $\mathscr{K} = \mathrm{mathcal}K(\lambda, L)$, error distribution $\chi = \chi(\lambda, L)$ on the parameter $n = n(\lambda, L) \in \mathbb{N}$ and $\mathbb{Z}$, so that the $(q, n, \chi) - \mathrm{LWE}$ problem is at least 2^λ secure for known attacks. Choose a parameter $m = m(\lambda, L) = O(n\log(q))$.

(2) Output: $\mathrm{params} = (n, q, \chi, m)$. We express $l = \lfloor \log(q) \rfloor + 1$ and $N = (n+1) \cdot l$.

GSW.KeyGen(params):

(1) Select $\mathbf{t} = (t_1, \ldots, t_n)^T \longleftarrow \mathbb{Z}_q^n$ and calculate

$$\mathbf{s} \longleftarrow \left(1, -\mathbf{t}^T\right)^T = \left(1, -t_1, \ldots, -t_n\right)^T \in \mathbb{Z}_q^{(n+1)\times 1}. \tag{18}$$

(2) Generate a matrix $\mathbf{B} \longleftarrow \mathbb{Z}_q^{mm\times n}$ and a vector $\mathbf{e} \longleftarrow \chi^m$.

(3) Calculate $\mathbf{b} = \mathbf{Bt} + \mathbf{e} \in \mathbb{Z}_q^m$ and construct matrix $\mathbf{A} = (b \mid B) \in \mathbb{Z}_q^{m\times (n+1)}$. Obviously, we observed.

(4) Return to sk$\longleftarrow \mathbf{s}$ and pk$\longleftarrow \mathbf{A}$.

GSW.Enc$(\text{params}, \text{pk}, \mu)$: in order to encrypt a single-bit message $\mu \in \{0, 1\}$,

(1) Let $\mathbf{G}$ be the abovementioned matrix $(n+1)\times N$

(2) Select a matrix $\mathbf{R} \longleftarrow \{0, 1\}^{m\times N}$ evenly

(3) Calculate

$$\mathbf{C} = \mu G + A^T \mathbf{R} \pmod q \in \mathbb{Z}_q^{(n+1)\times N} \tag{19}$$

In the original GSW scheme,

$\text{Flatten}(\mu\mathbf{I} + \text{BitDecomp}(\mathbf{RA})) \in \{0, 1\}^{N\times N}$, where $\mathbf{I}$ is an identity matrix.

GSW.Dec$(\text{params}, \text{sk}, C)$:

(1) We have $\text{sk} = \mathbf{s} \in \mathbb{Z}_q^{n+1}$.

(2) Let I meet $(q/4) < 2^{I-1} \leq (q/2)$. Let $\mathbf{C}_I$ be column I of $\mathbf{C}$.

(3) Calculate $x \longleftarrow \langle \mathbf{C}_I, \mathbf{s}\rangle \pmod q$ within the scope of $(-(q/2), (q/2)]$; note $\langle \mathbf{C}_I, \mathbf{s}\rangle = \mathbf{C}_I^T \mathbf{s}$ and

$$\mathbf{C}^T\mathbf{s} = \mu \mathbf{G}^T\mathbf{s} + \mathbf{R}^T\mathbf{As} = \mu(1, 2, 4, \ldots)^T + \mathbf{R}^T\mathbf{e}. \tag{20}$$

From that mentioned above, it can be seen that column I of the ciphertext matrix $\mathbf{C}$ selected in the calculation corresponds to coordinate I of the vector $\langle \mathbf{C}_I, \mathbf{s}\rangle$, i.e. $\mu 2^{I-1} + \mathbf{R}_I^T\mathbf{e}$.

(4) Output $\mu' = \left|\left\lfloor (x/2)^{I-1} \right\rceil\right|$.

So, if it is $|x| < 2^{I-2} \leq (q/4)$, then it returns to 0, and if it is $|x| > 2^{I-2}$, then it returns to 1.

GSW.Eval$(\text{params}, C_1, \ldots, C_l)$:

GSW.Mult$(\mathbf{C}_1, \mathbf{C}_2)$: calculate and output

$$\begin{aligned}\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) &= \left(\mu_1\mathbf{G} + \mathbf{A}^T\mathbf{R}_1\right)\mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mu_1\mathbf{C}_2 + \mathbf{A}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) \\ &= \mathbf{A}^T\left(\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mu_1\mathbf{R}_2 + \mu_1\mu_2\mathbf{G}(\text{mod} q)\right).\end{aligned} \tag{21}$$

GSW.Add$(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+1)\times N}$: output

$$\mathbf{C}_1 + \mathbf{C}_2 = (\mu_1 + \mu_2)\mathbf{G} + \mathbf{A}^T(\mathbf{R}_1 + \mathbf{R}_2). \tag{22}$$

Note that $\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+1)\times N}$. In addition, use $\mathbf{G} - \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$ to calculate homomorphic NAND gates.

Note 3. Note that, in [19], the decryption algorithm is to select a suitable vector $\mathbf{w}$ and calculate $\mathbf{sCG}^{-1}(\mathbf{w}^T)$. It is much less efficient than the original one (all about calculation time and error item size). So, we used the GSW decryption algorithm in our scheme.

When q is a power of 2, there is also a variant of the message in $\mathbb{Z}_q$. See more details in [4].

3.1. Security. A brief proof of the following theorem is given in [4].

Theorem 1. *Let (n, q, χ) be public parameter so that the $LWE_{(n,q,\chi)}$ hypothesis is true, and let $m = O(n\log(q))$. Then, we can say that the GSW scheme is IND – CPA safe.*

The most important step of the proof is to prove that $(\mathbf{A}, \mathbf{RA})$ and the uniform distribution is computational indistinguishable.

Note 4. The correctness of the GSW scheme is obtained by analyzing the scale of the noise during encryption, decryption, and homomorphism. Always ensure that the maximum noise level in the abovementioned process is still less than 1/4, which can be decrypted correctly. This work is not the focus of this paper, so it will not be repeated. See more details [4].

4. Message-Encapsulation FHE

4.1. Message-Encapsulation FHE (MFHE Scheme). Now, we introduce our MFHE scheme as follows: a message-encapsulation public-key encryption scheme based on the difficulty of the LWE hypothesis. We give the security parameter λ, set t to be the private keys number, and then, can encrypt the t-bit messages at one time.

Let $q = q(\lambda)$ be an integer, and let $\chi = \chi(\lambda)$ be a distribution set on $\mathbb{Z}$. The definition of the variant of the GSW scheme is similar to the cryptosystem proposed in [19, 27, 28]. More specifically,

params$\longleftarrow$MFHE.Setup$(1^\lambda, 1^L)$:

(1) In particular, we first select the modulo $q = q(\lambda)$, and the dimension of lattice $n = n(\lambda, L)$. We appropriately select the error distribution for $\chi = \chi(\lambda, L)$ for 2^λ security against known LWE attacks, Finally, we select the parameter $m = m(\lambda, L) = O(n\log q)$ and a parameter $t = O(\log(n))$.

(2) Let $l = \lfloor \log q \rfloor + 1$ and $N = (n+t)\cdots l$, and then, output params $= (n, q, \chi, m, t)$.

(pk, sk)$\longleftarrow$MFHE.KeyGen(params):

(1) For $i \in [t]$, select $\mathbf{t}_i^T = (t_{i,1}, \ldots, t_{i,n})$ from $\mathbb{Z}_q^{1\times n}$ and output

$$\begin{aligned}\mathbf{sk}_i &:= \mathbf{s}_i = \left(\mathbf{I}_i \mid -\mathbf{t}_i^T\right)^T \\ &= \left(0, \ldots, 1, \ldots, 0 \mid -t_{i,1}, \ldots, -t_{i,n}\right)^T \in \mathbb{Z}_q^{(n+t)\times 1},\end{aligned} \tag{23}$$

the i position of which is 1.

(2) Select a matrix $\mathbf{B} \longleftarrow \mathbb{Z}_q^{m\times n}$ and t vectors $\mathbf{e}_i \longleftarrow \chi^{m\times 1}$, $i \in [t]$ evenly, and then, calculate $\mathbf{b}_i = \mathbf{B} \cdot t_i + \mathbf{e}_i \,(\mathrm{mod}\, q)$ and output

$$\mathrm{pk} = \mathbf{P} = \left[\mathbf{b}_1 | \cdots | \mathbf{b}_t \,|\, \mathbf{B}\right] \in \mathbb{Z}_q^{m\times(n+t)}, \qquad (24)$$

where the size of pk is $O(nm \cdot \log^2 q)$. In addition, we observed that $\mathbf{P} \cdot s_i = \mathbf{e}_i \,(\mathrm{mod}\, q)$.

(3) Output $\mathrm{pk} \longleftarrow \mathbf{P}$ and $\mathrm{sk} \longleftarrow \mathbf{S} \coloneqq \{\mathbf{s}_1, \ldots, \mathbf{s}_t\}$. It is worth noting that $\mathbf{P} \cdot S = [\mathbf{e}_1, \ldots, \mathbf{e}_t] \,(\mathrm{mod}\, q)$.

$\mathbf{C} \longleftarrow \mathrm{MFHE.Enc}\,(\mathrm{params}, \mathrm{pk}, \mathbf{M})$:

(1) To encrypt t-bit $\mu_i \in 0, 1$, $\mu_i \in 0, 1$, embed the t bits into a $(t \times t)$-dimension matrix first, $\mathbf{U} = \mathrm{diag}\,(\mu_{1,1}, \ldots, \mu_{t,t}) \in 0, 1^{t\times t}$, where $\mu_{i,j} = 0$, $i \neq j$, and $j \in [t]$. Later, for simplicity, $\mu_{i,j}$ will be abbreviated as μ_i, and the message matrix is constructed using a plaintext matrix $\mathbf{U}$.

$$\mathbf{M} = \begin{pmatrix} \mathbf{U}_{t\times t} & 0_{t\times n} \\ 0_{n\times t} & \mathbf{E}_{n\times n} \end{pmatrix} \in \{0, 1\}^{(n+t)\times(n+t)}, \qquad (25)$$

where $\mathbf{U}$ is a random diagonal matrix, and note that $\mathbf{E}$ is a $(n \times n)$-dimensional matrix.

(2) Then, select a uniform matrix $\mathbf{R} \longleftarrow 0, 1^{m\times N}$. Calculate and output cipher text:

$$\mathbf{C} = M \cdot G + \mathbf{P}^T \cdot \mathbf{R} \,(\mathrm{mod}\, q) \in \mathbb{Z}_q^{(n+t)\times N}. \qquad (26)$$

Now, we propose a decryption algorithm for the MFHE scheme which allows us to recover all the message bits at the one time.

$\mathbf{U} \longleftarrow \mathrm{MFHE.Dec}\,(\mathrm{params}, \mathrm{pk}, \mathbf{C})$:

(1) First, assume that the user has a private key matrix $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_t) \in \mathbb{Z}_q^{(n+t)\times t}$ as follows:

$$\mathbf{S} \coloneqq \left(\mathbf{s}_1, \ldots, \mathbf{s}_t\right) = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ -t_{1,1} & \cdots & -t_{t,1} \\ \vdots & \ddots & \vdots \\ -t_{1,n} & \cdots & -t_{t,n} \end{pmatrix}. \qquad (27)$$

What needs to be noted here is

$$\mathbf{P} \cdot S = \left[\mathbf{b}_1 - \mathbf{B}\mathbf{t}_1, \ldots, \mathbf{b}_t - \mathbf{B}\mathbf{b}_t\right] = \left[\mathbf{e}_1, \ldots, \mathbf{e}_t\right]\left(\mathrm{mod} \in \mathbb{Z}_q^{m\times t}\right). \qquad (28)$$

Therefore, it is easy for us to get the bound of $\mathbf{P} \cdot S$ which is less than or equal to $t|\mathbf{e}|$, i.e. $\|\mathbf{P} \cdot S\| \leq t|\mathbf{e}|$.

(2) Define the matrix $\mathbf{W}\mathbb{Z}_q^{t\times((+t)}$ as follows:

$$\mathbf{W}^T \coloneqq \begin{pmatrix} \lceil \frac{q}{2} \rceil & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lceil \frac{q}{2} \rceil \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}. \qquad (29)$$

(3) Calculate and output

$$\mathbf{V}_{i,j} = \langle \mathbf{S}, \mathbf{C} \rangle \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) (\mathrm{mod}\, q) \in \mathbb{Z}_q^{t\times t}. \qquad (30)$$

Among them, we have $\langle \mathbf{S}, \mathbf{C} \rangle \in \mathbb{Z}_q^{t\times t}$, i.e.,

$$\langle \mathbf{S},\mathbf{C} \rangle = \mathbf{S}^T\mathbf{P}^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G} = \left[\mathbf{e}_1,\ldots,\mathbf{e}_t\right]^T\mathbf{R} + \mathbf{S}^T\mathbf{M}\mathbf{G}\,(\mathrm{mod}\, q). \qquad (31)$$

(4) Finally, use the results mentioned above to output the complete message $\mathbf{U} = \left\|\left\lfloor\left(\mathbf{V}_{i,j}/(q/2)\right)\right\rceil\right\| \in \{0, 1\}^{t\times t}$.

MFHE.Eval $(\mathrm{params}, \mathbf{C}_1, \ldots, \mathbf{C}_l)$:there are two algorithms, which are, homomorphic addition and homomorphic multiplication. For any two plaintext matrices $\mathbf{U}_1, \mathbf{U}_2 \in \{0, 1\}^{t\times t}$, we get the ciphertext separately.

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{M}_1 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_1, \\ \mathbf{C}_2 &= \mathbf{M}_2 \cdot \mathbf{G} + \mathbf{P}^T \cdot \mathbf{R}_2. \end{aligned} \qquad (32)$$

Therefore, the homomorphic addition and multiplication are as follows:

$\mathrm{MFHE.Mult}\,(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output

$$\begin{aligned} \mathbf{C}_1\mathbf{G}^{-1}\left(\mathbf{C}_2\right) = \left(\mathbf{M}_1\mathbf{G} + \mathbf{P}^T\mathbf{R}_1\right) \cdot \mathbf{G}^{-1}\left(\mathbf{C}_2\right) = \mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}\left(\mathbf{C}_2\right) \\ + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2 + \mathbf{M}_1\mathbf{M}_2\mathbf{G}\,(\mathrm{mod}\, q). \end{aligned} \qquad (33)$$

$\mathrm{MFHE.Add}\,(\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$: output $\mathbf{C}_1 + \mathbf{C}_2 = (\mathbf{M}_1 + \mathbf{M}_2)\mathbf{G} + \mathbf{P}^T(\mathbf{R}_1 + \mathbf{R}_2)$.

Here, we can calculate a homomorphic NAND gate from the output.

Note 5. Generally, we can choose different private keys sk_i to decrypt column j of the ciphertext $\mathbf{C}_j$ bit-by-bit and get the i

bit message of C_j, that is, we can get the bit in row i and column j under the i private key. However, it is actually possible to recover the entire message using the private key matrix $\mathbf{S}$ based on the abovementioned decryption algorithm. We calculate $\mathbf{V}_{i,j} = \mathbf{S}^T\mathbf{C} \cdot G^{-1}(\mathbf{W}^T)$ as follows:

$$\mathbf{V}_{i,j} = \lceil \frac{q}{2} \rceil \cdot \mathbf{U} + \begin{pmatrix} \mathbf{e}_1^T\mathbf{R} \\ \vdots \\ \mathbf{e}_t^T\mathbf{R} \end{pmatrix} \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) \in \mathbb{Z}_q^{t\times t}. \quad (34)$$

The magnitude of the noise can be simply calculated and verified to grow linearly compared to single-bit decryption algorithm.

$\mu_{i,j} \longleftarrow$ MFHE.bitDec (params, sk_i, $\mathbf{C}$, $\mathbf{w}_j$):

(1) Suppose we want to decrypt the bit $\mu_{i,j}$ of row i and column j, so let $\text{sk}_i = \mathbf{s}_i :=$, then define a vector so that the position is, and the other positions are 0, $j \in [t]$.

$$\mathbf{w}_j^T = \left[\underbrace{0, \ldots, \lceil \frac{q}{2} \rceil_j, \ldots, 0}_{t} \middle| \underbrace{0, \ldots, 0}_{n} \right]. \quad (35)$$

(2) For i, j to t, calculate

$$v_{i,j} = \mathbf{s}_i^T\mathbf{C} \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right) (\text{mod}\, q) \in \mathbb{Z}_q. \quad (36)$$

The inner product of $\langle \mathbf{s}_i, \mathbf{C} \rangle$ equals to

$$\mathbf{s}_i^T\mathbf{P}^T\mathbf{R} + \mathbf{s}_i^T\mathbf{M}\mathbf{G} = \mathbf{e}_i^T\mathbf{R} + \mathbf{s}_i^T\mathbf{M}\mathbf{G}\,(\text{mod}\, q)\mathbb{Z}_q^{1\times N}. \quad (37)$$

(3) Output a message $\mu_{i,j} = \| \lfloor \left(\mathbf{V}_{i,j}/(q/2) \right) \rceil \| \in \{0, 1\}$, in which $\lfloor \cdot \rceil$ represents the operation that rounds to the nearest integer. Therefore the value belongs to $\{0, 1\}$. 4. Finally, by repeating it t^2 times, the entire message can be recovered. The bitDec algorithm here is similar to the algorithm in [2], which is achieved by recovering each element separately.

Note 6. It should be noted here that due to the structural characteristics of the public key in our scheme, accurate decryption is achieved by dynamically adjusting the position of $\lceil (q/2) \rceil$ in the vector $\mathbf{w}$. That is, dot-multiply $\mathbf{s}_i^T\mathbf{C}$ and $\mathbf{G}^{-1}(\mathbf{w}_j)$ to obtain the bits of the row and column of the plaintext matrix.

We can get all the bits of the message by using the bitDec decryption algorithm and appropriate private key.

Note 7. It can be seen that our message-encapsulation GSW scheme is to implement $t \times t$-bit homomorphic addition. However, since the (i, j) element of $\mathbf{U}_1 \times \mathbf{U}_2$ is not a product of $\mu_{1_{i,j}} \times \mu_{2_{i,j}}$, only t-bit homomorphic multiplication is supported.

4.2. Correctness Analysis. Next, we analyze the correctness of the MFHE scheme.

Definition 11. We call the message matrix $\mathbf{U} \in \mathbb{Z}_q^{t\times t}$ which is obtained by decrypting the ciphertext under t different private keys $\mathbf{s}_i, i \in [t]$ (see (2)). The noise of a single-bit message is as follows:

$$\text{noise}_{(\mathbf{s}_i,\mathbf{M})} = \mathbf{s}_i^T\mathbf{C} - \mathbf{s}_i^T\mathbf{M}\mathbf{G} = \mathbf{s}_i^T\mathbf{P}^T\mathbf{R} = \mathbf{e}_i^T\mathbf{R}. \quad (38)$$

For flexible single-bit decryption algorithm bitDec, we represent the noise vector as noise $\in \mathbb{Z}_q^{1\times N}$. For simplicity, we abbreviate $\text{noise}_{(\mathbf{s}_i,\mathbf{M})}(\mathbf{C})$ to $\text{noise}_{\mathbf{s}_i}$ when $\mathbf{M}$ and $\mathbf{C}$ do not affect the contextual understanding.

Note that, in our setup, due to the structure of the new public key, $\text{noise}_{\mathbf{s}_i}$ is the noise of row i of the plaintext matrix $\mathbf{U}$, not the single-bit noise.

Lemma 4. *Obviously, using Definition 4.1, for convenience, for a decryption algorithm Dec, if the noise meets*

$$\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = \mathbf{S}^T \cdot \mathbf{P}^T \cdot \mathbf{R} = \begin{pmatrix} \text{noise}_{\mathbf{s}_1} \\ \vdots \\ \text{noise}_{\mathbf{s}_t} \end{pmatrix} (\text{mod} q), \quad (39)$$

where $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_t]$ *is a one-time private key matrix, we can represent the entire noise matrix as*

$$\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C}) = \left(\text{noise}_{\mathbf{s}_1}, \ldots, \text{noise}_{\mathbf{s}_t}\right)^T \in \mathbb{Z}_q^{t\times N}. \quad (40)$$

For convenience, we will abbreviate $\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C})$ *as* $\text{Noise}_{\mathbf{S}}$ *when* $\mathbf{M}$ *and* $\mathbf{C}$ *do not affect the contextual understanding.*

In order to analyze the correctness, for convenience, we first define the following noise ciphertext concept.

Definition 12 (E-Noise Ciphertext). A ciphertext matrix $\mathbf{C} \in \mathbb{Z}_q^{(m+1)\times N}$ with E noise, which makes in a private key $\mathbf{s}_i \in \mathbb{Z}_q^{(n+t)\times 1}$, for a corresponding message $\mathbf{M}$, $\langle \mathbf{s}_i, \mathbf{C} \rangle = \mathbf{s}_i^T \cdot \mathbf{M} \cdot G + \mathbf{e}_i^T \cdot \mathbf{R}$. Then, let the norm of $\text{noise}_{\mathbf{s}_i}$ be

$$\left\|\text{noise}_{\mathbf{s}_i}\right\| \leq \left\|\mathbf{e}_i^T\mathbf{R}\right\| \leq \left\|\mathbf{e}_i^T\right\|_2 \cdot \|\mathbf{R}\|_\infty \leq \sqrt{N} \cdot 2\sqrt{m}B \leq E. \quad (41)$$

Lemma 5. *For a one-time private key matrix* $\mathbf{S} \in \mathbb{Z}_q^{(n+t)\times t}$, *we can get* $\text{Noise}_{\mathbf{S}} = [\mathbf{e}_1, \ldots, \mathbf{e}_t]^T \cdot \mathbf{R}$ *when we run the Dec algorithm. So, in this case, we get*

$$\left\|\text{Noise}_{\mathbf{S}}\right\| \leq t \cdot \left\|\text{noise}_{\mathbf{s}_i}\right\| \leq t \cdot E. \quad (42)$$

Lemma 6. *For a plaintext matrix* $\mathbf{U}$ *(a combination of* $\mathbf{M}$*) and a private key* $\mathbf{s}_i, i \in [t]$, *the noise vector of the ciphertext* $\mathbf{C}$ *meets*

$$t\left\|\text{noise}_{\mathbf{s}_i}\right\| = \left\|\text{Noise}_{\mathbf{S}}\right\|. \quad (43)$$

In the following, we analyze the correctness of the decryption.

Lemma 7. *Let* $\mathbf{C}$ *be an* E *noise encryption of* $\mathbf{M}$*. If we can recover* $\mu_{i,j}$ *(an element of* $\mathbf{U}$*) from the ciphertext* $\mathbf{C}$ *under the private key* $\mathbf{s}_i$*, then there is*

$$\mu_{i,j} := \langle \mathbf{s}_i, \mathbf{C}\rangle \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right) = \left(\text{noise}_{\mathbf{s}_i} + \mathbf{s}_i^T \mathbf{MG}\right) \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right), \tag{44}$$

so that

$$\left\|\text{noise}_{\mathbf{s}_i} \cdot \mathbf{G}^{-1}\left(\mathbf{w}_j^T\right)\right\|_\infty \leq \left\|\text{noise}_{\mathbf{s}_i}\right\| \cdot \left\|\mathbf{G}^{-1}\left(\mathbf{w}_j^T\right)\right\| \leq N \cdot E < \frac{q}{8}. \tag{45}$$

Proof. Obviously, by using Lemma 4.2 we can simply prove Lemma 4.7, and we will not go into details here. □

Lemma 8. *Let* $\mathbf{C}$ *be an* E *noise encryption in* $\mathbf{M}$*. If we can recover all* $\mathbf{U}$ *from the ciphertext* $\mathbf{C}$*, then there is a private key matrix* $\mathbf{S}$ *such that*

$$\mathbf{V} = \langle \mathbf{S}, \mathbf{C}\rangle \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right) = \left(\text{Noise}_{\mathbf{S}} + \mathbf{S}^T \mathbf{MG}\right) \cdot \mathbf{G}^{-1}\left(\mathbf{W}^T\right), \tag{46}$$

where $\|Noise_{SS} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \leq N \cdot tE < (q/8)$.

Proof. This proof can be obtained directly from Lemma 4.2 and Lemma 4.7. Now, we know that as long as $\|\text{Noise}_{\mathbf{S}} \cdot \mathbf{G}^{-1}(\mathbf{W}^T)\|_\infty \leq (q/8)$, the decryption runs correctly, i.e., $E < (q/4tN)$. Therefore, we call the value $E = (q/4tN)$ as the bound of noise.

The analysis of the homomorphic operation is given in the following. Before introducing the boundary of noise, the following notes are given. □

Note 8. For the convenience of reading, let $\Upsilon_{\mathbf{C}_1} := \text{Noise}_{(\mathbf{S},\mathbf{M}_1)}(\mathbf{C}_1)$ and $\Upsilon_{\mathbf{C}_2} := \text{Noise}_{(SS,\mathbf{M}_2)}(\mathbf{C}_2)$.

Lemma 9 *(See [8]). The boundary of the noise of homomorphic addition, homomorphic multiplication, and homomorphic negative is as follows:*

Addition: for $\mathbf{M}_1, \mathbf{M}_2 \in \{0,1\}^{(n+t)\times(n+t)}$*, the following condition is met:*

$$\left\|\text{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)}(\mathbf{C}_1 + \mathbf{C}_2)\right\| \leq \left\|\Upsilon_{\mathbf{C}_1}\right\| + \left\|\Upsilon_{\mathbf{C}_2}\right\|. \tag{47}$$

Multiplication: for $\mathbf{M}_1, \mathbf{M}_2$*, the following condition is met:*

$$\begin{aligned}\left\|\text{Noise}_{(\mathbf{S},(\mathbf{M}_1\cdot\mathbf{M}_2))}\left(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)\right)\right\| &\leq \|\mathbf{U}_1\|_2 \\ &\cdot \left\|\Upsilon_{\mathbf{C}_2}\right\|_\infty + \left\|\mathbf{G}^{-1}(\mathbf{C}_2)\right\|_\infty \cdot \left\|\Upsilon_{\mathbf{C}_1}\right\|_\infty.\end{aligned} \tag{48}$$

NAND: for $\mathbf{M}$*, the following condition is met:*

$$\left\|\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{G} - \mathbf{C})\right\| = \left\|\text{Noise}_{(\mathbf{S},\mathbf{M})}(\mathbf{C})\right\|. \tag{49}$$

Proof. Let $\mathbf{S} \in \mathbb{Z}^{(n+t)\times t}$ be a private key matrix. Let $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{(m+1)\times N}$ be the ciphertext of the encrypted message $\mathbf{M}_1, \mathbf{M}_2 \in \{0,1\}^{(n+t)\times(n+t)}$ separately. Then,

Homomorphic addition, that is, add ciphertext and ciphertext $\mathbf{C}^{\text{Add}} = \mathbf{C}_1 + \mathbf{C}_2 \pmod q$, so that

$$\langle \mathbf{S}, \mathbf{C}^{\text{Add}}\rangle = \text{Noise}_{(SS,\mathbf{M}_1+\mathbf{M}_2)} + \mathbf{S}^T \cdot \mathbf{M}^{\text{Add}} \cdot \mathbf{G}. \tag{50}$$

Where $\mathbf{M}^{\text{Add}} = \mathbf{M}_1 + \mathbf{M}_2$ and the noise is

$$\text{Noise}_{(\mathbf{S},\mathbf{M}_1+\mathbf{M}_2)} = \text{Noise}_{(\mathbf{S},\mathbf{M}_1)} + \text{Noise}_{(\mathbf{S},\mathbf{M}_2)}. \tag{51}$$

Obviously, the noise is $t \cdot (E_1 + E_2)$.

Homomorphic multiplication: that is, multiply the ciphertext and ciphertext $\mathbf{C}^{\text{Mult}} = \mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2) \in \mathbb{Z}_q^{(n+t)\times N}$, so that

$$\mathbf{C}^{\text{Mult}} = \mathbf{M}_1\mathbf{M}_2\mathbf{G} + \left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right). \tag{52}$$

Then, we have $\langle \mathbf{S}, \mathbf{C}^{\text{Mult}}\rangle$ which equals to

$$\mathbf{S}^T\left(\mathbf{M}_1\mathbf{M}_2\mathbf{G} + \left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right)\right). \tag{53}$$

For convenience, we first set the noise to

$$\text{Noise}_{(\mathbf{S},\mathbf{M}_1\mathbf{M}_2)} = \mathbf{S}^T\left(\mathbf{P}^T\mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + \mathbf{M}_1\mathbf{P}^T\mathbf{R}_2\right). \tag{54}$$

Obviously, according to Lemma 4.2, there is

$$\left\|\Upsilon_{\mathbf{C}_1}\right\| = \left\|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1\right\| \leq \left\|[\mathbf{e}_1, \ldots, \mathbf{e}_t]^T \cdot \mathbf{R}_1\right\| \leq tE_1, \tag{55}$$

and $\mathbf{C}_2$ is a $(n+t)\times N$ binary matrix ($\mathbf{G}^{-1} \in \mathbb{Z}_q^{N\times(n+t)}$). Therefore, in this case,

$$\left\|\mathbf{S}^T\mathbf{P}^T\mathbf{R}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2)\right\| \leq tE_2 \cdot \left\|\mathbf{G}^{-1}(\mathbf{C}_2)\right\| \leq N \cdot tE_2 \tag{56}$$

exists. Also, pay attention to that

$$\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\right) = \begin{pmatrix} \left(u_i\mathbf{b}_1^T - \mathbf{t}_i^T\mathbf{B}^T\right) \\ \vdots \\ \left(u_i\mathbf{b}_t^T - \mathbf{t}_i^T\mathbf{B}^T\right) \end{pmatrix}. \tag{57}$$

The boundary of $(u_i \cdot \mathbf{b}_i^T - \mathbf{t}_i^T \cdot \mathbf{B}^T)$ is $|\mathbf{e}_i^T|$. Therefore,

$$\left\|\mathbf{S}^T \cdot \left(\mathbf{M}_1\mathbf{P}^T\right)\right\| \leq \left\|\left[\mathbf{e}_1^T, \ldots, \mathbf{e}_t^T\right]^T\right\| \leq \max_i\left\|\mathbf{e}_i^T\right\|. \tag{58}$$

In this case, we can easily get the boundary $\|\Upsilon_{\mathbf{C}_2}\| := \|\mathbf{S}^T \cdot (\mathbf{M}_1\mathbf{P}^T\mathbf{R}_2)\| \leq \|\mathbf{e}_i^T\mathbf{R}\| \leq E_2$. In other words, $\|\mathbf{U}_1\|_2 \cdot \|\Upsilon_{\mathbf{C}_2}\|_\infty \leq \sqrt{t} \cdot E_2$. Therefore, we have $\|\text{Noise}_{(SS,(\mathbf{M}_1\cdot\mathbf{M}_2))}(\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2))\| \leq NtE_2 + \sqrt{t}E_2$, and the ciphertext $\mathbf{C}^{\text{Mult}}$ is $((Nt + \sqrt{r}) \cdot E)$ noisy.

Table 1: Comparison of the related works.

Underlying	HAO	Ours-MFHE	LMDO[
Assumption	LWE	LWE	dual-LWE
$\|msg\|$	t	t	t
$\|\mathbf{pk}\|$	$\mathcal{O}(mn\log q)$	$\mathcal{O}(nm\log q)$	$\mathcal{O}(mn\log q)$
$\|\mathbf{sk}\|$	$\mathcal{O}(nt\log q)$	$\mathcal{O}(nt\log q)$	$\mathcal{O}(nt\log q)$
$\|ct\|$	$\mathcal{O}(nN\log q)$	$\mathcal{O}(nN\log q)$	$\mathcal{O}(nN'\log q)$
flexibledec	√	√	√
one – timedec	×	√	√
_$\|msg\|$: message length		_N: $(n+t)l$	
_$\|ct\|$: cipher text length		_N': $(m+t)l$	

NAND gate: the same operation is true for the NAND gate, and output matrix product is $\mathbf{G}-\mathbf{C}_1\mathbf{G}^{-1}(\mathbf{C}_2)$. Consider a Boolean circuit whose computational depth is L while containing NAND gates. It takes the new ciphertext as input, that is, the E noise ciphertext, the noise multiplied by a factor which is at most $(Nt+\sqrt{t})$ at each level, that is, the norm of the error element increases by a factor which is, at most, $(Nt+\sqrt{t})$. Therefore, the wrong element norm of the final ciphertext is bounded as $E_{\text{final}}=(Nt+\sqrt{t})^L\cdot E$.

In order to ensure the correctness of the decryption, $E_{\text{final}}\le(\lfloor(q/2)\rfloor/4)$ needs to be true. That is to say, the inequality $(Nt+\sqrt{t})^L\cdot E\le(\lfloor(q/2)\rfloor/4)$ must be true, which is guaranteed by the parameters we choose. The proof is completed. □

4.3. IND – CPA Security Analysis. In the following, we use Theorem 4.1 to prove that the message-encapsulation GSW scheme based on the LWE assumption that it is IND – CPA safe and that the scheme is indistinguishable from the original GSW scheme [4].

Theorem 2. *Let $m>n\in\mathbb{N},q\in\mathbb{N}$ and χ be a discrete Gaussian distribution on $\mathbb{Z}$, which makes the (n,q,χ,m) – LWE problem difficult. Let t be an integer that makes $t=O(\log(n))$ true. Define two distributions $\mathcal{X}$ and $\mathcal{Y}$ as follows:*

$\mathcal{X}$ is a distribution on the $m\times(t+n)$ matrix $[\mathbf{b}_1|\cdots|\mathbf{b}_t|\mathbf{B}]$. Among them, $\mathbf{B}\in\mathbb{Z}_q^{m\times n}$ is uniformly selected, for all $1\le i\le t$, $\mathbf{b}_i=\mathbf{B}\mathbf{t}_i+\mathbf{e}_i(\text{mod}q)$, in which $\mathbf{t}_i$ are uniformly selected from $\mathbb{Z}_q^n$, and $\mathbf{e}_i$ is selected from a discrete Gaussian distribution χ.

$\mathcal{Y}$ is evenly distributed on $\mathbb{Z}_q^{m\times(t+n)}$.

Therefore, the distribution $\mathcal{X}$ and $\mathcal{Y}$ is computational indistinguishable.

Theorem 3. *Let params $=(n,q,\chi,m,t)$ so that the assumption $LWE_{n,q,\chi,m}$ is true and $m=O(n\log q)$. Then, the MFHE scheme is IND – CPA safe.*

Proof. The proof of security contains two steps:

First, we use Theorem 4.11 to prove that, under the LWE assumption, the matrix $\mathbf{P}=[\mathbf{b}_1,\ldots,\mathbf{b}_t,\mathbf{B}]\in\mathbb{Z}_q^{m\times(n+t)}$ and the randomly chosen matrix are computationally indistinguishable

Then, using the Left-over Hash Lemma, a uniform random value $\mathbf{C}'$ can be used to replace the ciphertext $\mathbf{C}=\mathbf{MG}+\mathbf{P}^T\mathbf{R}$, that is, $\mathbf{P}^T\cdot\mathbf{R}$ is indistinguishable from the uniform distribution

The brief proof is over. See more details in [4]. □

5. Conclusions

In this paper, we construct an efficient message-encapsulation FHE scheme. The scheme can achieve the decryption at one time and can also flexibly decrypt bit-by-bit. In Table 1, we give a comparison of the parameters of this scheme with the existing schemes. It can be seen from the comparison that compared with the previous ones, the scheme keeps the key length substantially, and this scheme is based on more conventional assumptions and, meanwhile, reduces the ciphertext length to some extent. The proposal of this scheme makes the full homomorphic encryption take a big step from theoretical research to large-scale application. It is conducive to greatly improving the efficiency of encrypted data processing (such as retrieval and operation) in the Internet of things, saving the energy consumption of nodes in the Internet of Things, and ensuring that the data are not statistically analyzed, which has a better application scenario [29–31].

In addition, there are many interesting open issues that may be resolved in the future. For example, our thinking has certain reference value for enhancing big data security and constructing a message-encapsulated casual transmission protocol, but it also has certain challenges.

Data Availability

No data were used in this study.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Proceedings of the Public-Key Cryptography—PKC 2013—16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 1–13, Nara, Japan, February 2013.

[2] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," in *Proceedings of the Public-Key Cryptography—PKC 2015—18th IACR International Conference on Practice and Theory in Public-Key Cryptography*, pp. 699–715, Gaithersburg, MD, USA, March 2015.

[3] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Proceedings of the Advances in Cryptology—CRYPTO 2012—32nd Annual Cryptology Conference*, pp. 868–886, Santa Barbara, CA, USA, August 2012.

[4] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2013—33rd Annual Cryptology Conference*, pp. 75–92, Santa Barbara, CA, USA, August 2013.

[5] Z. Li, C. Ma, E. Morais, and G. Du, "Multi-bit leveled homomorphic encryption via dual LWE-based," in *Proceedings of the Revised Selected Papers Information Security and Cryptology—12th International Conference*, pp. 221–242, Beijing, China, November 2016.

[6] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1–34, 2009.

[7] Z. Li, S. D. Galbraith, and C. Ma, "Preventing adaptive key recovery attacks on the gentry-sahai-waters leveled homomorphic encryption scheme," *IACR Cryptology ePrint Archive*, p. 1146, 2016.

[8] Z. Brakerski and R. Perlman, "Lattice-based fully dynamic multi-key FHE with short ciphertexts," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2016—36th Annual International Cryptology Conference*, pp. 190–213, Santa Barbara, CA, USA, August 2016.

[9] Z. Li, C. Ma, and D. Wang, "Leakage resilient leveled FHE on multiple bit message," *IEEE Transactions on Big Data*, 2017.

[10] Z. Li, C. Ma, and D. Wang, "Towards multi-hop homomorphic identity-based proxy re-encryption via branching program," *IEEE Access*, vol. 5, pp. 16214–16228, 2017.

[11] Z. Li, C. Ma, and D. Wang, "Achieving multi-hop PRE via branching program," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 45–58, 2020.

[12] Z. Li, C. Ma, and H. Zhou, "Multi-key FHE for multi-bit messages," *Sciece China Information Sciences*, vol. 61, no. 2, Article ID 029101, 2018.

[13] Z. Li, C. Xiang, and C. Wang, "Oblivious transfer via lossy encryption from lattice-based cryptography," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5973285, 11 pages, 2018.

[14] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proceedings of the Topics in Cryptology—CT-RSA 2011—the Cryptographers' Track at the RSA Conference 2011*, pp. 319–339, San Francisco, CA, USA, February 2011.

[15] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Proceedings of the 28th Annual International Cryptology Conference Advances in Cryptology—CRYPTO 2008*, pp. 554–571, Santa Barbara, CA, USA, August 2008.

[16] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudorandom generation from one-way functions (extended abstracts)," in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pp. 12–24, Seattle, Washigton, USA, May 1989.

[17] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *Proceedings of the STOC 2009 41st Annual ACM Symposium on Theory of Computing*, pp. 333–342, Bethesda, MD, USA, May 2009.

[18] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 187–196, British Columbia, Canada, May 2008.

[19] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Proceedings of the Part II Advances in Cryptology—EUROCRYPT 2016—35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 735–763, Vienna, Austria, May 2016.

[20] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proceedings of the Advances in Cryptology—EUROCRYPT 2012— 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 738–755, Cambridge, UK, April 2012.

[21] J. Katz, A. Thiruvengadam, and H. Zhou, "Feasibility and infeasibility of adaptively secure fully homomorphic encryption," in *Proceedings of the Public-Key Cryptography—PKC 2013—16th International Conference on Practice and Theory in Public-Key Cryptography*, pp. 14–31, Nara, Japan, February 2013.

[22] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, CRC Press, Boca Raton, FL, USA, Second edition, 2014.

[23] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the FOCS 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 97–106, Palm Springs, CA, USA, October 2011.

[24] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled FHE from learning with errors," in *Proceedings of the Part II Advances in Cryptology—CRYPTO 2015—35th Annual Cryptology Conference*, pp. 630–656, Santa Barbara, CA, USA, August 2015.

[25] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *Proceedings of the ITCS'14 Innovations in Theoretical Computer Science*, pp. 1–12, Princeton, NJ, USA, January 2014.

[26] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of the Advances in Cryptology—EUROCRYPT 2012—31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 700–718, Cambridge, UK, April 2012.

[27] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Proceedings of the Part I Advances in Cryptology—CRYPTO 2014—34th Annual Cryptology Conference*, pp. 297–314, Santa Barbara, CA, USA, August 2014.

[28] L. Ducas and D. Micciancio, "FHEW: bootstrapping homomorphic encryption in less than a second," in *Proceedings of the Part I Advances in Cryptology—EUROCRYPT 2015—34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 617–640, Sofia, Bulgaria, April 2015.

[29] Z. Li, V. Sharma, C. Ma, C. Ge, and W. Susilo, "Ciphertext-policy attribute-based proxy re-encryption via constrained PRFS," *Science China Information Sciences*, vol. 64, no. 6, 2020.

[30] Z. Li, J. Wang, C. Choi, and W. Zhang, "Multi-factor password-authenticated key exchange via pythia PRF service," *Computers, Materials & Continua*, vol. 63, no. 2, pp. 663–674, 2020.

[31] V. Sharma, D. N. K. Jayakody, and M. Qaraqe, "Osmotic computing-based service migration and resource scheduling in mobile augmented reality networks (MARN)," *Future Generation Computer Systems*, vol. 102, pp. 723–737, 2020.