Hindawi
Security and Communication Networks
Volume 2018, Article ID 8706940, 14 pages
https://doi.org/10.1155/2018/8706940

WILEY | Hindawi

Research Article

A More Efficient Fully Homomorphic Encryption Scheme Based on GSW and DM Schemes

Xun Wang,[1] **Tao Luo**,[1,2] **and Jianfeng Li**[2]

[1]*Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Tao Luo; tluo@bupt.edu.cn

Received 29 May 2018; Revised 29 October 2018; Accepted 7 November 2018; Published 16 December 2018

Academic Editor: Jiankun Hu

Achieving both simplicity and efficiency in fully homomorphic encryption (FHE) schemes is important for practical applications. In the simple FHE scheme proposed by Ducas and Micciancio (DM), ciphertexts are refreshed after each homomorphic operation. And ciphertext refreshing has become a major bottleneck for the overall efficiency of the scheme. In this paper, we propose a more efficient FHE scheme with fewer ciphertext refreshings. Based on the DM scheme and another simple FHE scheme proposed by Gentry, Sahai, and Waters (GSW), ciphertext matrix operations and ciphertext vector additions are both applied in our scheme. Compared with the DM scheme, one more homomorphic NOT AND (NAND) operation can be performed on ciphertexts before ciphertext refreshing. Results show that, under the same security parameters, the computational cost of our scheme is obviously lower than that of GSW and DM schemes for a depth-2 binary circuit with NAND gates. And the error rate of our scheme is kept at a sufficiently low level.

1. Introduction

With the rapid development of computer networks and big data, the cloud has been playing an important role in storing and processing huge amounts of data [1]. The cloud provides abundant, flexible, and on-demand remote storage and computational resources for network users. However, the cloud is not fully trustable, and users do not have full control power on the data stored in the cloud. Data in the cloud are faced with the risk of leakage, and personal privacy is seriously threatened. In some recent research works, approaches based on network defense have been proposed for guaranteeing cloud security [2–6]. Nevertheless, data encryption provides a more fundamental and universal privacy protection for data in the cloud. In traditional encryption techniques, when the encrypted data are stored in the cloud, they need to be decrypted before computation, and personal privacy is still seriously threatened. Homomorphic encryption allows ciphertext operations to be performed directly; thus an untrusted third party can process the ciphertexts without decrypting them. The decryption of the result of ciphertext operation is equivalent to the result of corresponding plaintext operation. Furthermore, fully homomorphic encryption (FHE) allows arbitrary operations to be performed on ciphertexts. Concretely, let **Enc** and **Dec** denote encryption and decryption algorithms, respectively. And let m_i and c_i denote the plaintexts and corresponding ciphertexts, respectively, where $i = 1, 2, \ldots, l$ and $c_i = \mathbf{Enc}(m_i)$. For a function f_m of plaintexts $m_1, m_2, \ldots, m_l$, and a corresponding function f_c of ciphertexts $c_1, c_2, \ldots, c_l$, FHE schemes satisfy the following property:

$$\mathbf{Dec}\left(f_c\left(c_1, c_2, \ldots, c_l\right)\right) = f_m\left(m_1, m_2, \ldots, m_l\right) \tag{1}$$

This ideal property can be applied to privacy protection in the cloud, where personal data are stored and processed in encrypted form.

In FHE schemes, ciphertexts are generated with a random noise to ensure semantic security. The noise grows as homomorphic operations proceed. When the noise magnitude exceeds a certain threshold, ciphertext will no longer be correctly decrypted. By means of bootstrapping proposed by Gentry [7], ciphertext noise can be reduced and further

homomorphic operations can be performed. However, due to its inherent complexity, bootstrapping has become a major bottleneck for the efficiency of all FHE schemes. Although there are many studies on improving the efficiency of FHE schemes [8–27], they are still not simple and efficient enough to be widely adopted in the real world. Designing a conceptually simple and efficient FHE scheme has become a challenging issue.

In this paper, a new FHE scheme is proposed to achieve both conceptual simplicity and higher efficiency. The scheme is constructed using the ideas of ciphertext matrix operations in the FHE scheme proposed by Gentry, Sahai and Waters (GSW) [19] and ciphertext vector additions in the FHE scheme proposed by Ducas and Micciancio (DM) [21]. Both these schemes are conceptually simpler than most other FHE schemes, while suffering from low efficiency. We have proved that, compared with DM, our scheme allows one more homomorphic operation to be performed before ciphertext refreshing. And the computational cost of our scheme is significantly lower than that of DM and GSW under the same security parameters, with the error rate kept at a sufficiently low level. Our scheme not only inherits the advantage of conceptual simplicity in DM and GSW but is also more efficient.

Assumptions. The assumptions in our scheme are specified as follows: (1) the hardness of the Learning with Errors (LWE) problem [28]; (2) circular security in ciphertext refreshing; that is, one can safely encrypt a secret key under its associated public key [7]; (3) the operations on the binary circuit which are performed parallelly. And the computational cost at each level is represented as that of a specific gate at the level.

Contributions. The main contributions of our scheme are summarized as follows: (1) To the best of our knowledge, our scheme is one of the few FHE schemes which take both simplicity and efficiency into consideration. (2) Our scheme inherits the advantage of conceptual simplicity in DM and GSW, which are conceptually simpler than most other FHE schemes. (3) Our scheme combines the advantage of efficient homomorphic operation in DM with the advantage of moderate growth of noise magnitude in GSW. When compared with DM, it allows one more homomorphic operation to be performed before ciphertext refreshing. Under the same security parameters, the computational cost of our scheme is obviously lower than that of DM and GSW, and the error rate is kept at a sufficiently low level.

Organization. The rest of this paper is organized as follows: the related work is discussed in Section 2; some preliminaries are given in Section 3; a review of GSW and DM is presented in Section 4; our more efficient FHE scheme, along with its correctness, security, and applicability analysis, is presented in Section 5; the comparison of our scheme with DM and GSW in terms of overall efficiency and error rate is given in Section 6; finally, conclusions are drawn in Section 7.

2. Related Work

2.1. Construction of FHE Schemes. Gentry proposed the first FHE scheme in 2009 [7], which marks a milestone in the research of homomorphic encryption. Gentry's FHE scheme is based on ideal lattices, which includes the following major steps: (1) the construction of a somewhat homomorphic encryption scheme (SWHE) which allows limited homomorphic additions and multiplications to be performed on ciphertexts; (2) the squashing step for reducing the complexity of decryption algorithm; (3) the bootstrapping technique for reducing ciphertext noise via re-encryption and homomorphic decryption. Despite its significant contribution, Gentry's scheme sufferes from a rather low efficiency. Following Gentry's work, some other FHE schemes based on ideal lattices have been proposed on improving the efficiency of Gentry's scheme [8–11]. However, the inherent complicated key generation process, along with large key/ciphertext sizes, has made these schemes impractical for real-world applications.

In 2010, Dijk et al. proposed a FHE scheme over the integers [29]. Both the keys and ciphertexts are integers, which are much simpler than previous FHE schemes based on ideal lattices. However, the scheme also suffers from low efficiency due to large key/ciphertext sizes. Although some improved FHE schemes on integers have been proposed [12–15], keys and ciphertexts in these schemes are still too large to be deployed in any practical system.

Recently, most FHE schemes have been constructed based on the LWE problem, which is a computational problem over lattices [28]. LWE has now drawn the attention of more and more cryptographic researchers with its relatively small key/ciphertext sizes and strong security. Brakerski and Vaikuntanathan presented the first LWE-based FHE scheme (BV) in 2011 [30]. The relinearization technique was introduced for controlling ciphertext dimension in homomorphic multiplications. And the dimension-modulus reduction technique was proposed as a new method for simplifying the decryption algorithm to make the scheme bootstrappable, thus fully homomorphic. Compared with the squashing technique proposed by Gentry, the sparse subset-sum assumption was removed in dimension-modulus reduction, making it more natural. Brakerski, Gentry, and Vaikuntanathan proposed a leveled FHE scheme (BGV) in 2012 [16]. The relinearization and dimension-modulus reduction techniques were improved as the key-switching and modulus-switching techniques in BGV, for more efficient control of ciphertext dimension and noise magnitude. Brakerski then introduced a scale-invariant leveled FHE scheme (Bra12) without modulus switching. Compared with previous LWE-based FHE schemes, Bra12 is simpler, and ciphertext noise magnitude grows by a constant multiplicative factor as homomorphic operations proceed, instead of exponentially. However, in all of these schemes, the complex process of key switching (or relinearization) still introduces a huge computational cost, which is unattractive in practice.

In 2013, a new leveled FHE scheme, known as GSW, was proposed by Gentry, Sahai and Waters [19]. GSW is based on approximate eigenvectors of matrices. The ciphertexts in GSW are square matrices, and homomorphic additions and multiplications are just matrix additions and multiplications, respectively. Therefore, ciphertext dimension always keeps constant and key switching is no longer necessary. Scale-invariance can also be achieved in GSW via the flatten

technique; thus modulus switching is also no longer necessary. GSW is simpler and more natural than previous LWE-based FHE schemes. However, matrix multiplication still brings about a high computational cost. Ducas and Miccianico proposed a new FHE scheme with homomorphic NOT AND (NAND) gates [21], which is known as the DM scheme. Homomorphic operations in DM are just ciphertext vector additions, which are very simple operations. However, ciphertexts in DM need to be refreshed after each homomorphic operation, which becomes a bottleneck for the overall efficiency. Although GSW and DM are conceptually simpler than most other FHE schemes, both of them still suffer from efficiency bottlenecks.

Other research works on the construction of LWE-based FHE schemes generally focus on improving the efficiency [22–25] and optimizing the bootstrapping algorithm [26, 27]. In some recent research works, multikey FHE schemes are proposed for secure multiparty computation [31, 32]. However, these schemes involve either key-switching, or ciphertext matrix operations, which are both computationally costing. Some of them are not conceptually simple. Therefore, it is necessary to construct a new FHE scheme with both conceptual simplicity and higher efficiency.

2.2. Applications of Homomorphic Encryption Schemes. As homomorphic encryption supports operations on encrypted data, it is definitely more powerful than traditional encryption techniques and has a vast area of applications. In recent years, with the wide adoption of cloud storage and cloud computation in real-world applications, there have been many applications of homomorphic encryption schemes on privacy protection in the cloud.

Searchable encryption is a basic application of homomorphic encryption, where users can execute secure queries on encrypted data. The query results are obtained through homomorphic operations between the encrypted query and the encrypted data. A lot of researchers have proposed secure information retrieval schemes based on homomorphic encryption [33–36]. Meng Shen et al. proposed a graph encryption scheme which makes use of SWHE and enables approximate Constrained Shortest Distance (CSD) querying over encrypted graph [37]. Another common application of homomorphic encryption schemes is secure e-voting, where the ballots of voters are encrypted and homomorphic operations are performed on these data [38–41]. The property of homomorphic encryption makes it possible to tally all encrypted ballots without accessing the plaintext content of any individual ballot; thus voter's privacy is protected. Recently, with the rapid development of artificial intelligence and machine learning, privacy protection in machine learning has also drawn the attention of many researchers. Many studies on encrypted machine learning have emerged, where homomorphic encryption schemes are adopted for computation on encrypted data. Xiaoqiang Sun et al. implemented three private classification algorithms based on homomorphic encryption [42], which were hyperplane decision-based classification, Naïve Bayes classification, and decision tree classification. M Kim et al. proposed secure logistic regression for biomedical data [43]. There are also lots of research works on secure deep learning based on homomorphic encryption [44–46]. The activation functions in deep learning algorithms are usually approximated as polynomials, which can be homomorphically evaluated by homomorphic encryption schemes. Other recent applications of homomorphic encryption include integrity verification [47, 48], data aggregation [49, 50], and secure multiparty computation [32, 51].

Moreover, homomorphic encryption can be applied in the defense against phishing attack, where user's personal information is encrypted, and the verification is completed via homomorphic operations. Even if personal information is leaked to the phishing server, nothing can be learned from the encrypted data. Longfei Wu et al. proposed a novel automated lightweight antiphishing scheme for mobile platforms, which is highly beneficial for mobile users [52]. Adopting homomorphic encryption in the scheme would provide an even stronger defense against phishing attacks. With the rise of self-awareness of privacy protection and the development of homomorphic encryption, there will be more and more applications of homomorphic encryption in the future.

3. Preliminaries

3.1. Notations. The mathematical symbols in this paper are shown in Table 1.

3.2. The LWE Problem. LWE is a computational problem over lattices, which is proposed by Regev [28]. For security parameter λ, let $n = n(\lambda)$ and $q = q(\lambda)$ denote the dimension and modulus of the vector, respectively, and let $\chi = \chi(\lambda)$ denote the random distribution on $\mathbb{Z}$ for the random errors. The vector $\mathbf{s}$ is generated by sampling $\mathbf{s} \longleftarrow \mathbb{Z}$. For vector $\mathbf{a} \longleftarrow \mathbb{Z}_q^n$ and error $e \longleftarrow \chi$, output the following LWE instance $(\mathbf{a}, b) = (\mathbf{a}, (\mathbf{a} \cdot \mathbf{s} + e) \bmod q) \in \mathbb{Z}_q^{n+1}$. The LWE assumption is that the distribution χ' formed by different LWE instances is computationally indistinguishable from the uniform distribution on $\mathbb{Z}_q^{n+1}$.

3.3. The Cyclotomic Ring. Let N be a power of 2, the $2N$-th cyclotomic polynomial is $\Phi_{2N}(X) = X^N + 1$, and the corresponding polynomial ring is $R = \mathbb{Z}[X]/X^N + 1$. $R_Q = R/QR$ denotes the residue ring of R modulo an integer Q. Each element in R is a polynomial with integer coefficients whose degree is at most $N - 1$, and each element in R_Q is an element in R with all its coefficients modulo Q. For polynomial $r = \sum_{i=0}^{N-1} r_i X^i \in R$, let $\mathbf{CF}(r) = (r_0, \ldots, r_{N-1})$ denote the coefficient vector of the polynomial. And let $\mathbf{ACR}(r)$ denote the following matrix: the first column is $\mathbf{CF}(r)$, and the other columns are the anticyclic rotations of $\mathbf{CF}(r)$ with the cycled entries negated, as shown in

$$\mathbf{ACR}(r) = \begin{bmatrix} r_0 & -r_{N-1} & \cdots & -r_1 \\ r_1 & r_0 & \cdots & -r_2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{N-1} & r_{N-2} & \cdots & r_0 \end{bmatrix} \quad (2)$$

Table 1: List of mathematical symbols with their meanings.

Symbol	Meaning
Regular letters (with possibly superscripts and subscripts), e.g. q, n', B.	Scalars.
Bold lowercase letters (with possibly superscripts and subscripts), e.g. $\mathbf{e}, \mathbf{c}'$.	Vectors.
Bold uppercase letters (with possibly superscripts and subscripts), e.g. $\mathbf{A}, \mathbf{C}'$.	Matrices.
$\mathbb{Z}$	The set of all integers.
$\mathbb{C}$	The set of all complex numbers.
$\mathbb{Z}^+$	The set of all positive integers.
$\mathbb{Z}_q$	The set of integers modulo an integer q, which are reduced to $(-q/2, q/2]$.
$\mathbb{Z}_q^{m\times n}$	The set of $m \times n$ matrices with all coefficients in $\mathbb{Z}_q$.
$\mathbb{Z}[X]$	The set of all polynomials with integer coefficients.
$\mathcal{B}(n, p)$	Binomial distribution with parameters n, p.
$\lfloor x \rceil$	Rounding of x to the nearest integer.
$\langle \mathbf{a}, \mathbf{b} \rangle$ or $\mathbf{a} \cdot \mathbf{b}$	Inner product of vectors $\mathbf{a}, \mathbf{b}$.
$[\mathbf{A}\|\|\mathbf{b}]$	The horizontal concatenation of matrix $\mathbf{A}$ and vector $\mathbf{b}$.
$\|\mathbf{a}\|_\infty$	The infinite norm of vector $\mathbf{a}$, $\|\mathbf{a}\|_\infty = \max_i \|a_i\|$.
$d \longleftarrow D$	If D is a distribution, d is sampled according to D; If D is a set, d is uniformly sampled from D.
$\mathrm{negl}(\lambda)$	A negligible amount: $\mathrm{negl}(\lambda) = o(\lambda^{-c})$ for any constant $c > 0$ as $\lambda \longrightarrow +\infty$.

3.4. BitDecomp and Flatten Techniques. Let $\mathbf{BD}(\cdot)$ denote the BitDecomp operation, and let $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^k$, $l = \lfloor \log q \rfloor + 1$, $N = kl$. The BitDecomp operation is defined as follows:

$$\mathbf{BD}(\mathbf{a}) = \left(a_{1,0}, \ldots, a_{1,l-1}, \ldots, a_{k,0}, \ldots, a_{k,l-1}\right) \tag{3}$$

where $a_{i,j}$ is the j-th bit in a_i's binary representation from the lowest to the highest bit. After BitDecomp, the upper bound of $\mathbf{a}$'s l_1 norm goes down from nq to $n \log q$. Let $\mathbf{BD^{-1}}(\cdot)$ denote the inverse operation of $\mathbf{BD}(\cdot)$; for a vector $\mathbf{a}' = (a_{1,0}, \ldots, a_{1,l-1}, \ldots, a_{k,0}, \ldots, a_{k,l-1}) \in \mathbb{Z}_q^N$, the operation $\mathbf{BD^{-1}}(\cdot)$ is defined as follows:

$$\mathbf{BD^{-1}}\left(\mathbf{a}'\right) = \left(\sum_{j=0}^{l-1} 2^j a_{1,j}, \ldots, \sum_{j=0}^{l-1} 2^j a_{k,j}\right) \in \mathbb{Z}_q^k \tag{4}$$

Let $\mathbf{FL}(\cdot)$ denote the flatten operation; for a vector $\mathbf{a}' \in \mathbb{Z}_q^k$, $\mathbf{FL}(\cdot)$ is defined as follows:

$$\mathbf{FL}\left(\mathbf{a}'\right) = \mathbf{BD}\left(\mathbf{BD^{-1}}\left(\mathbf{a}'\right)\right) \in \{0,1\}^N \tag{5}$$

There is another operation $\mathbf{PowersofTwo}(\cdot)$ which comes hand in hand with $\mathbf{BD}(\cdot)$. Let $\mathbf{PT}(\cdot)$ denote the operation $\mathbf{PowersofTwo}(\cdot)$, which is defined as follows:

$$\mathbf{PT}(\mathbf{b}) = \left(b_1, 2b_1, \ldots, 2^{l-1}b_1, \ldots, b_k, 2b_k, \ldots, 2^{l-1}b_k\right) \in \mathbb{Z}_q^N \tag{6}$$

An obvious property between $\mathbf{BD}(\cdot)$ and $\mathbf{PT}(\cdot)$ is shown as follows:

$$\langle \mathbf{BD}(\mathbf{a}), \mathbf{PT}(\mathbf{b}) \rangle = \langle \mathbf{a}, \mathbf{b} \rangle \tag{7}$$

For a vector $\mathbf{a}' \in \mathbb{Z}_q^N$, the following property also holds:

$$\left\langle \mathbf{a}', \mathbf{PT}(\mathbf{b}) \right\rangle = \left\langle \mathbf{BD^{-1}}\left(\mathbf{a}'\right), \mathbf{b} \right\rangle = \left\langle \mathbf{FL}\left(\mathbf{a}'\right), \mathbf{PT}(\mathbf{b}) \right\rangle \tag{8}$$

It can be observed from (8) that an important advantage of $\mathbf{FL}(\cdot)$ lies in that it makes the coefficients of a vector small, without affecting its inner product with the vector $\mathbf{PT}(\mathbf{b})$. When the above operations are applied to a matrix, they are performed for each row of the matrix.

4. A Review of GSW and DM Schemes

4.1. The GSW Scheme. GSW is constructed based on approximate eigenvectors of matrices. And homomorphic operations in GSW are just ciphertext matrix operations. GSW is more natural and concise than previous LWE-based FHE schemes which require key switching (or relinearization). The main algorithms in GSW are shown as follows:

(i) $\mathbf{GSW.KeyGen}(\lambda, L)$: λ, L denote the security parameter and multiplicative depth, respectively. Ciphertext dimension $n = n(\lambda, L)$, modulus $q = q(\lambda, L)$, and noise distribution $\chi = \chi(\lambda, L)$ are set to guarantee a security level of λ. Let $m = O(n \log q)$, $l = \lfloor \log q \rfloor + 1$, $N = (n+1)l$, and parameter set $params = (n, q, \chi, m)$. Sample $\mathbf{t} \longleftarrow \mathbb{Z}_q^n$, let $\mathbf{s} = (1, -\mathbf{t}) \in \mathbb{Z}_q^{n+1}$, and output secret key $sk = \mathbf{v} = \mathbf{PT}(\mathbf{s})$. Sample $\mathbf{B} \longleftarrow \mathbb{Z}_q^{m\times n}$, $\mathbf{e} \longleftarrow \chi^m$, let $\mathbf{b} = \mathbf{B} \cdot \mathbf{t} + \mathbf{e}$, $\mathbf{A} = [\mathbf{b} \parallel \mathbf{B}]$, and output public key $pk = \mathbf{A}$.

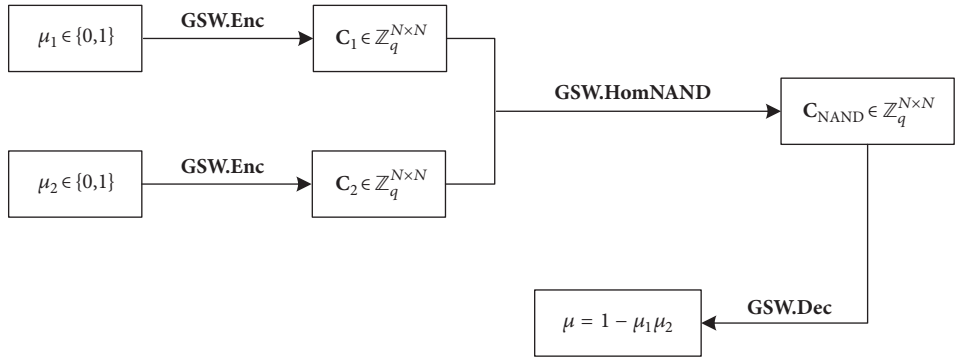

Figure 1: Overall algorithm flow of the GSW scheme.

(ii) **GSW**.**Enc**($params, pk, \mu$): for plaintext message $\mu \in \mathbb{Z}_q$, sample $\mathbf{R} \longleftarrow \{0,1\}^{N\times m}$; output ciphertext:

$$\mathbf{C} = \mathbf{FL}\left(\mu \cdot \mathbf{I}_N + \mathbf{BD}\left(\mathbf{R} \cdot \mathbf{A}\right)\right) \in \mathbb{Z}_q^{N\times N} \tag{9}$$

where $\mathbf{I}_N$ denotes the N-dimensional identity matrix.

(iii) **GSW**.**HomNAND**($\mathbf{C}_1, \mathbf{C}_2$): on input ciphertext pair $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{N\times N}$, output ciphertext

$$\mathbf{C}_{\text{NAND}} = \mathbf{FL}\left(\mathbf{I}_N - \mathbf{C}_1\mathbf{C}_2\right) \tag{10}$$

As a result of the homomorphic NAND operation, $\mathbf{C}_{\text{NAND}}$ satisfies the following property:

$$\mathbf{C}_{\text{NAND}} \cdot \mathbf{v} = \left(1 - \mu_1\mu_2\right)\mathbf{v} - \mu_2\mathbf{e}_1 - \mathbf{C}_1\mathbf{e}_2 \tag{11}$$

where μ_1, μ_2 are the plaintext messages in $\mathbf{C}_1, \mathbf{C}_2$, respectively, and $\mathbf{e}_1, \mathbf{e}_2$ are the corresponding ciphertext noises. Let B_0 denote the upper bound of the noise magnitudes in $\mathbf{C}_1, \mathbf{C}_2$, that is, the upper bound for the l_∞ norms of $\mathbf{e}_1, \mathbf{e}_2$. It is obvious that $\max\{\|\mathbf{e}_1\|_\infty, \|\mathbf{e}_2\|_\infty\} < B_0$. Actually, $\mathbf{C}_1, \mathbf{C}_2 \in \{0,1\}^{N\times N}$ as a result of the flatten operation. As $\mu_2 \in \{0,1\}$, the noise in $\mathbf{C}_{\text{NAND}}$ is upper bounded by $(N+1)B_0$, as shown by (11).

The overall algorithm flow of GSW is shown in Figure 1.

4.2. The DM Scheme. DM is a FHE scheme based on a LWE symmetric encryption scheme. Homomorphic operations in DM correspond to ciphertext vector additions. DM is conceptually simple for its simple homomorphic operation. The main algorithms in DM are shown as follows:

(i) **DM**.**KeyGen**(λ): λ denotes the security parameter. Integer $t \geq 2$ is the plaintext modulus. Ciphertext dimension $n = n(\lambda)$, modulus $q = q(\lambda)$, and ciphertext noise distribution $\chi = \chi(\lambda)$ are set to guarantee a security level of λ. Here $x < q/2t$ for any $x \longleftarrow \chi$. Let $params$ denote the parameter set $params = (n, q, t, \chi)$. The key is uniformly sampled from $\mathbb{Z}_q^n$: $pk/sk \longleftarrow \mathbb{Z}_q^n$.

(ii) **DM**.**Enc**($m, pk, params$): the plaintext and ciphertext spaces are $\mathbb{Z}_t, \mathbb{Z}_q$, respectively. Sample $\mathbf{a} \longleftarrow \mathbb{Z}_q^n$, $e \longleftarrow \chi$, on input plaintext message $m \in \mathbb{Z}_t$, and output ciphertext:

$$\text{LWE}_{\mathbf{s}}^{t/q}(m) = \left(\mathbf{a}, \mathbf{a}\cdot\mathbf{s} + \frac{mq}{t} + e\right) \in \mathbb{Z}_q^{n+1} \tag{12}$$

(iii) **DM**.**HomNAND**($(\mathbf{a}_1, b_1), (\mathbf{a}_2, b_2)$): on input ciphertexts $\mathbf{c}_i = (\mathbf{a}_i, b_i)$, $i \in \{1,2\}$ and $\mathbf{c}_i \in \text{LWE}_{\mathbf{s}}^{4/q}(m_i, q/16)$ encrypts the plaintext message m_i, output $\mathbf{c} = (\mathbf{a}, b) \in \text{LWE}_{\mathbf{s}}^{2/q}(1 - m_1m_2, q/4)$. In particular,

$$(\mathbf{a}, b) = \left(-\mathbf{a}_1 - \mathbf{a}_2, \frac{5}{8}q - b_1 - b_2\right) \tag{13}$$

The ciphertext $(\mathbf{a}, b)$ is a ciphertext of $1 - m_1m_2$ with noise magnitude less than $q/4$, which guarantees correct decryption. Homomorphic NAND operations in DM are completed by a few additions between ciphertext vectors, which are simpler and faster than tensor products or matrix operations in previous schemes. However, ciphertext magnitude would be at least $q/4$ after a further homomorphic operation, then the ciphertext would no longer be correctly decrypted. After each homomorphic operation, ciphertext needs to be refreshed to keep the noise magnitude small.

An efficient ciphertext refreshing algorithm based on Ring-GSW is proposed in DM for reducing ciphertext noise. In the refreshing algorithm, ciphertext $(\mathbf{a}, b) \in \text{LWE}_{\mathbf{s}}^{2/q}(m, q/4)$ and refreshing key $\boldsymbol{K}_{rf}$ are taken as input, and base B_r is used to encode the ciphertext $(\mathbf{a}, b)$. $\boldsymbol{K}_{rf}$ consists of the following ciphertexts:

$$\begin{aligned} &K_{i,c,j} = E\left(cs_iB_r^j \bmod q\right), \\ &\quad c \in \{0, \ldots, B_r - 1\},\ j = 0, \ldots, d_r - 1,\ i = 1, \ldots, n \end{aligned} \tag{14}$$

where $d_r = \lceil \log_{B_r} q \rceil$ and $E(\cdot)$ denotes the encryption algorithm in the ciphertext refreshing algorithm. The ciphertext refreshing algorithm is shown as in Algorithm 1, where **Init**($\cdot$) and **Incr**($\cdot$) denote the initialization and homomorphic addition of the accumulator ACC, respectively. ACC is initialized as an encryption of $b + q/4$. When the main

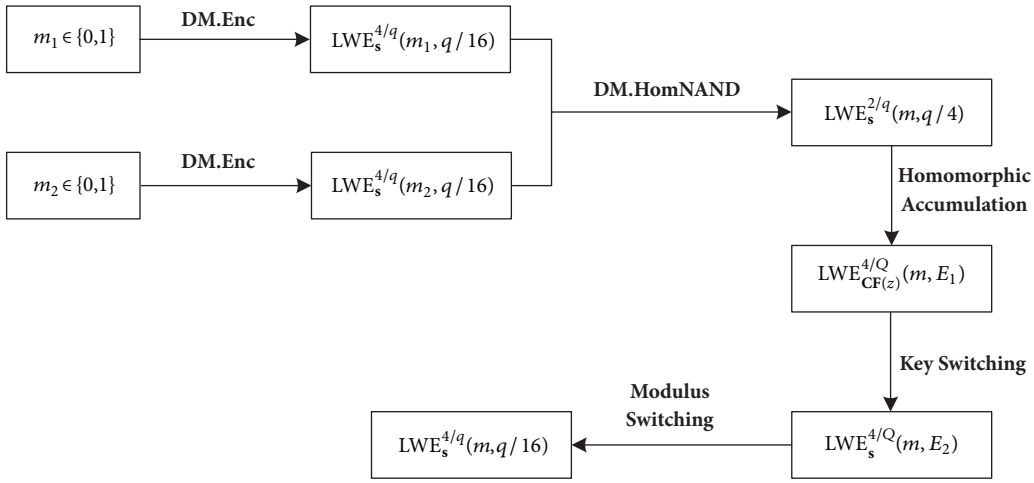

Figure 2: Overall algorithm flow of the DM scheme.

1 **Init**$(\mathrm{ACC}, b + q/4)$
2 **for** $i = 1, \ldots, n$ **do**
3 **Expand** $-a_i$ **as** $-a_i = \sum_j a_{i,j} B_r^j \pmod q$
4 **for** $j = 0, \ldots, d_r - 1$ **do Incr**$(\mathrm{ACC}, K_{i,a_{i,j},j})$
5 **end for**
6 **Output msbExtract**(ACC)

Algorithm 1: **DM**. **Refresh**$((\mathbf{a}, b), \boldsymbol{K}_{rf})$.

1 $[\mathbf{a}^t, \mathbf{b}^t] = [\mathbf{0}^t, \mathbf{t}^t, \mathbf{0}^t, \ldots, \mathbf{0}^t] \cdot \mathbf{ACR}(\mathrm{ACC})$
2 $\mathbf{c} = (\mathbf{a}, b_0 + u) \in \mathrm{LWE}_{\mathbf{CF}(z)}^{t/Q}(\mathrm{msb}(v))$
3 $\mathbf{c}' = \mathbf{KeySwitch}(\mathbf{c}, \boldsymbol{K}_{ks}) \in \mathrm{LWE}_{\mathbf{s}}^{t/Q}(\mathrm{msb}(v))$
4 $\mathbf{c}'' = \mathbf{ModSwitch}(\mathbf{c}') \in \mathrm{LWE}_{\mathbf{s}}^{t/q}(\mathrm{msb}(v))$

Algorithm 2: **msbExtract**$(\mathrm{ACC}, \boldsymbol{K}_{ks}, \mathbf{t})$.

loop in Algorithm 1 ends, the underlying plaintext v of the accumulator satisfies

$$v - \frac{q}{4} = b + \sum_{i,j} a_{i,j} s_i B_r^j = b + \sum_i s_i \sum_j B_r^j a_{i,j} = b - \sum_i a_i s_i = \frac{q}{2} m + e \tag{15}$$

where e is the noise in the input ciphertext $(\mathbf{a}, b)$. As $|e| < q/4$, it is clear that $0 < v < q/2$ when $m = 0$ and $q/2 < v < q$ when $m = 1$. In other words, extracting the most significant bit (msb) in v would yield the plaintext m.

During the **msbExtract** process in Algorithm 1, the accumulator ACC, along with a switching key $\boldsymbol{K}_{ks}$ and a testing vector $\mathbf{t} = -\sum_{i=0}^{q/2-1} \mathbf{CF}(Y^i)$, is taken as input. Here $Y = X^{2N/q}$, and $z \in R$ is the secret key used in the encryption algorithm of the ciphertext refreshing algorithm. The details of **msbExtract** are shown in Algorithm 2.

The ciphertext $\mathbf{c}$ in the 2nd step of Algorithm 2 is

$$\mathbf{c} = (\mathbf{a}, b_0 + u) = (\mathbf{a} \cdot \mathbf{CF}(z) + \mathbf{t} \cdot \mathbf{e} + 2u \cdot \mathrm{msb}(v)) \tag{16}$$

where $\mathbf{a} = \mathbf{t}^t \cdot \mathbf{ACR}(a)$, $[a, b']$ is the 2nd row of ACC and $u = \lceil Q/2t \rceil$ or $\lfloor Q/2t \rfloor$. As $u \approx Q/2t$, $\mathbf{c}$ is an encryption of $\mathrm{msb}(v)$=m. Thus, $\mathbf{c} \in \mathrm{LWE}_{\mathbf{CF}(z)}^{t/Q}(\mathrm{msb}(v))$. After key and modulus switching, $\mathbf{c}$ is transformed to a ciphertext under key $\mathbf{s}$ modulo q. Under an appropriate parameter setting, the noise magnitude of the refreshed ciphertext would be lower than $q/16$, and further homomorphic operations can be performed.

The overall algorithm flow of DM is shown in Figure 2, where $m = 1 - m_1 m_2$.

5. Efficient FHE Scheme Based on GSW and DM Schemes

Aimed at the problem of overly frequent ciphertext refreshings in DM, a new FHE scheme (NHE) is proposed to achieve a higher efficiency. The ciphertext matrix operations in GSW and ciphertext vector additions in DM are both applied in our scheme. And the advantage of conceptual simplicity of both GSW and DM is inherited in our scheme. Moreover, our scheme combines the advantage of efficient homomorphic operation in DM with the advantage of moderate growth of noise magnitude in GSW. The whole scheme is briefly shown here, and some related details will be illustrated later.

(i) **NHE.KeyGen**(λ): here λ denotes the security parameter. Modulus $q = q(\lambda) = 2^k (k \in \mathbb{Z}^+)$, dimension $n = n(\lambda)$, and ciphertext noise distribution $\chi = \chi(\lambda)$ are set to guarantee a security level of λ. Concretely, χ is a discrete Gaussian distribution over integers with zero mean and standard deviation σ. Let *params* denote the parameter set (n, q, χ), and let $l = \log q + 1, N = (n+1)l$. Sample $\mathbf{t} \longleftarrow \mathbb{Z}_q^n$, $\mathbf{s}$=$(1, -\mathbf{t}) \in \mathbb{Z}_q^{n+1}$; output secret key $sk = \mathbf{v} = \mathbf{PT}(\mathbf{s}) \in \mathbb{Z}_q^N$. Sample $\mathbf{B} \longleftarrow$

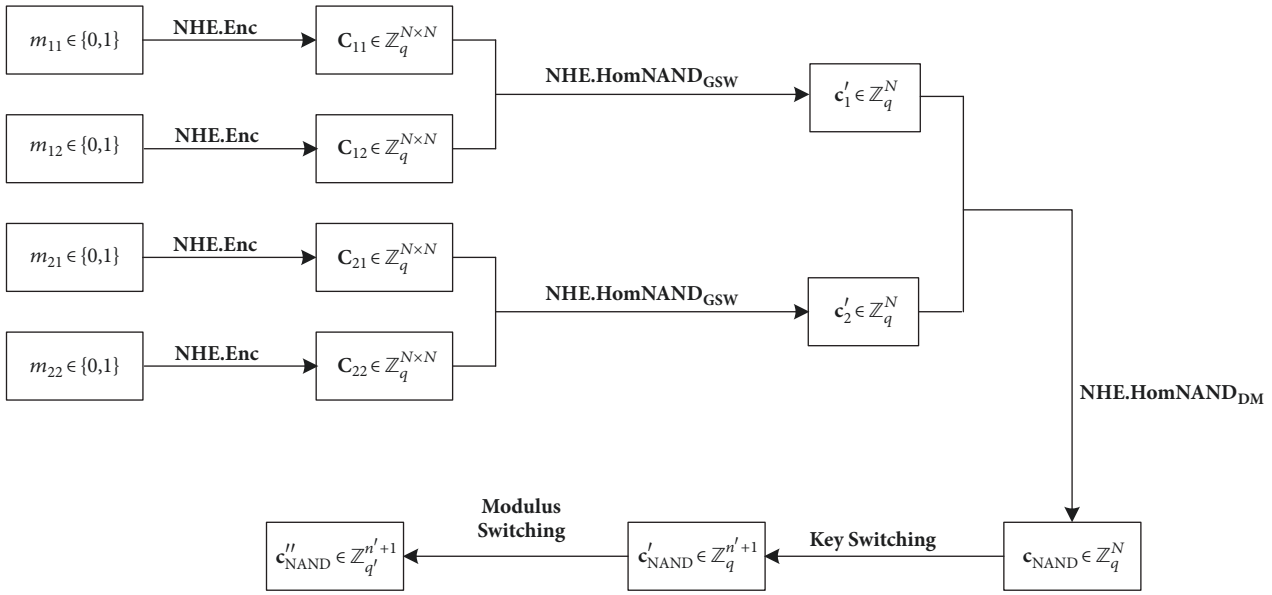

Figure 3: Overall algorithm flow of our scheme.

$\mathbb{Z}_q^{N\times n}, \mathbf{e} \longleftarrow \chi^N$, let $\mathbf{b}=\mathbf{Bt}+\mathbf{e}$, $\mathbf{A} = [\mathbf{b} \parallel \mathbf{B}]$, and output public key $pk = \mathbf{A}$.

(ii) **NHE.Enc**(m, pk): on input plaintext message $m \in \{0,1\}$, output ciphertext

$$\mathbf{C} = \mathbf{FL}\left(m\mathbf{I}_N + \mathbf{BD}\left(\mathbf{A}\right)\right) \in \mathbb{Z}_q^{N\times N} \tag{17}$$

(iii) **NHE.HomNAND**$(\mathbf{C}_1, \mathbf{C}_2)$: the input ciphertexts $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{N\times N}$ correspond to encryptions of m_1, $m_2 \in \{0,1\}$, respectively. Each ciphertext is assumed to have an internal attribute *level* indicating the number of homomorphic operations it has gone through. The *level* of any ciphertext is 0 in the beginning and increases by 1 after each homomorphic operation. For $\mathbf{C}_1, \mathbf{C}_2$ such that $\mathbf{C}_1.level = \mathbf{C}_2.level = 0$, homomorphic NAND operation is performed as follows:

$$\mathbf{C}' = \mathbf{FL}\left(\mathbf{I}_N - \mathbf{C}_1\mathbf{C}_2\right) \tag{18}$$

Then the $(l-2)$-th row is extracted from $\mathbf{C}'$ as the ciphertext $\mathbf{c}' \in \mathbb{Z}_q^N$ for the next homomorphic NAND operation. Clearly, $\mathbf{c}'.level = 1$. For a pair of ciphertexts $\mathbf{c}_1', \mathbf{c}_2' \in \mathbb{Z}_q^N$ such that $\mathbf{c}_1'.level = \mathbf{c}_2'.level = 1$, homomorphic NAND operation is performed as follows:

$$\mathbf{c}_{\mathrm{NAND}} = \mathbf{FL}\left(-\mathbf{c}_1' - \mathbf{c}_2' + \mathbf{c}_0\right) \in \{0,1\}^N \tag{19}$$

where $\mathbf{c}_0$ is an auxiliary vector such that $\mathbf{c}_0 = \mathbf{BD}((5q/8, \mathbf{0})) \in \{0,1\}^N$. The homomorphic operations in (18) and (19) are based on the ideas of ciphertext matrix operations in GSW and ciphertext vector additions in DM, respectively.

(iv) **NHE.KeySwitch**$(\mathbf{c}_{\mathrm{NAND}}, \boldsymbol{K}_{ks})$: the switching key $\boldsymbol{K}_{ks}$ consists of the following ciphertexts: $\mathbf{k}_{i,c} \in \mathrm{LWE}_{\mathbf{s}'}^{q/q}(cv_i), i = 1,\ldots,N, c \in \{0,1\}$, where $\mathbf{s}' \longleftarrow \{0,1\}^{n'}$ is the new secret key. On input ciphertext $\mathbf{c}_{\mathrm{NAND}}$ and the switching key $\boldsymbol{K}_{ks}$, output ciphertext

$$\mathbf{c}'_{\mathrm{NAND}} = \sum_i \mathbf{k}_{i,c_i} \in \mathbb{Z}_q^{n'+1} \tag{20}$$

The above ciphertext $\mathbf{c}'_{\mathrm{NAND}}$ is a ciphertext under the new secret key $\mathbf{s}'$ instead of $\mathbf{v}$.

(v) **NHE.ModSwitch**$(\mathbf{c}'_{\mathrm{NAND}})$: on input ciphertext $\mathbf{c}'_{\mathrm{NAND}}$, output ciphertext

$$\mathbf{c}''_{\mathrm{NAND}} = \left\lfloor \frac{q'}{q}\mathbf{c}'_{\mathrm{NAND}} \right\rceil \in \mathbb{Z}_{q'}^{n'+1} \tag{21}$$

where $q' = 2^{k'}(k' \in \mathbb{Z}^+)$ and $q' < q$. $\mathbf{c}''_{\mathrm{NAND}}$ is the final ciphertext after 2 homomorphic NAND operations. The modulus in $\mathbf{c}''_{\mathrm{NAND}}$ is transformed from q to q'. Moreover, dimension and modulus of $\mathbf{c}''_{\mathrm{NAND}}$ are set to be the same as those of the ciphertexts in DM.

NHE.KeySwitch corresponds to the sum of some $(n'+1)$-dimensional vectors, and **NHE.ModSwitch** corresponds to rounding for each coefficient in a single vector. Both algorithms involve just simple operations, which have no significant effect on the simplicity of our scheme. The overall algorithm flow of our scheme is shown in Figure 3. Here the algorithms **NHE.HomNAND$_{\mathbf{GSW}}$** and **NHE.HomNAND$_{\mathbf{DM}}$** denote the algorithms in (18) and (19), respectively.

5.1. Correctness Analysis. Assuming there are 4 fresh ciphertexts $\mathbf{C}_{11}, \mathbf{C}_{12}, \mathbf{C}_{21}, \mathbf{C}_{22} \in \mathbb{Z}_q^{N\times N}$, as shown in Figure 3.

Each coefficient of the noise vectors in the above ciphertexts follows a discrete Gaussian distribution with zero mean and standard deviation σ. According to the property of discrete Gaussian distribution, the probability of each coefficient being in the interval $[-6\sigma, 6\sigma]$ is p_0, which is very close to 1. The probability of the all the noises in $\mathbf{C}_{11}, \mathbf{C}_{12}, \mathbf{C}_{21}, \mathbf{C}_{22}$ being upper bounded by $B_0 = 6\sigma$ is thus $p_1 = p_0^{4N}$.

It can be learned from (18) that the ciphertext $\mathbf{C}'$ satisfies

$$\mathbf{C}'\mathbf{v} = \left(1 - m_1 m_2\right)\mathbf{v} - m_2\mathbf{e}_1 + \mathbf{C}_1\mathbf{e}_2 \tag{22}$$

where $\mathbf{e}_1, \mathbf{e}_2$ are the noise vectors in $\mathbf{C}_1, \mathbf{C}_2$, respectively. As the first l coefficients in $\mathbf{v}$ are $(v_1, v_2, \ldots, v_l) = (1, 2, \ldots, 2^{l-1})$ and $l = \log q + 1$, it is clear that $v_{l-2} = q/4$. Let $\mathbf{C}'_{l-2}$ denote the $(l-2)$-th row in $\mathbf{C}'$; we have

$$\mathbf{C}'_{l-2}\mathbf{v} = \left(1 - m_1 m_2\right)\frac{q}{4} + e \tag{23}$$

where $e = \mathbf{C}_{1,l-2} \cdot \mathbf{e}_2 - m_2 e_{1,l-2}$. And $e_{1,l-2}, \mathbf{C}_{1,l-2}$ are the $(l-2)$-th coefficient in $\mathbf{e}_1$ and the $(l-2)$-th row in $\mathbf{C}_1$, respectively. e should satisfy the constraint $|e| < q/16$ to guarantee correct decryption after the next homomorphic operation.

Assume the ciphertexts $\mathbf{c}'_1, \mathbf{c}'_2$ in (19) encrypt $m'_1, m'_2 \in \{0, 1\}$, respectively, and the ciphertext noises in $\mathbf{c}'_1, \mathbf{c}'_2$ are e'_1, e'_2, respectively. Clearly, $\mathbf{c}'_i \cdot \mathbf{v} = m'_i(q/4) + e'_i$, $i \in \{1, 2\}$. It is clear that

$$\left(-\mathbf{c}'_1 - \mathbf{c}'_2\right) \cdot \mathbf{v} = -\left(m'_1 + m'_2\right)\frac{q}{4} - \left(e'_1 + e'_2\right) \tag{24}$$

$$\mathbf{c}_0 \cdot \mathbf{v} = \left\langle \left(5q/8, \mathbf{0}\right), (1, -\mathbf{t}) \right\rangle = 5q/8 \tag{25}$$

Let the plaintext spaces of $\mathbf{c}'_i (i = 1, 2)$ and $\mathbf{c}_{\mathrm{NAND}}$ be $\mathbb{Z}_4$ and $\mathbb{Z}_2$, respectively, as in DM. The noise in the ciphertext $\mathbf{c}_{\mathrm{NAND}}$ is

$$\begin{aligned} e_{\mathrm{NAND}} &= \mathbf{c}_{\mathrm{NAND}} \cdot \mathbf{v} - \left(1 - m'_1 m'_2\right)\frac{q}{2} \\ &= -\left(m'_1 + m'_2\right)\frac{q}{4} - \left(e'_1 + e'_2\right) + \frac{5}{8}q \\ &\quad - \left(1 - m'_1 m'_2\right)\frac{q}{2} \\ &= \frac{q}{4}\left[\frac{1}{2} - \left(m'_1 - m'_2\right)^2\right] - \left(e'_1 + e'_2\right) \end{aligned} \tag{26}$$

For each ciphertext $\mathbf{k}_{i,c}$ in the switching key $\boldsymbol{K}_{ks}$, we have $\mathbf{k}_{i,c} = (\mathbf{a}_{ks}, \mathbf{a}_{ks} \cdot \mathbf{s}' + cv_i + e_{ks})$ where $\mathbf{a}_{ks} \longleftarrow \mathbb{Z}_q^{n'}, e_{ks} \longleftarrow \chi$. Thus, the ciphertext $\mathbf{c}'_{\mathrm{NAND}}$ can be further expressed as

$$\begin{aligned} &\mathbf{c}'_{\mathrm{NAND}} \\ &= \left(\sum_{i=1}^{N} \mathbf{a}_{ks,i}, \left(\sum_{i=1}^{N} \mathbf{a}_{ks,i}\right) \cdot \mathbf{s}' + \mathbf{c}_{\mathrm{NAND}} \cdot \mathbf{v} + \sum_{i=1}^{N} e_{ks,i}\right) \\ &\in \mathbb{Z}_q^{n'+1} \end{aligned} \tag{27}$$

where $\mathbf{a}_{ks,i} \longleftarrow \mathbb{Z}_q^{n'}, e_{ks,i} \longleftarrow \chi, i = 1, \ldots, N$. The noise in $\mathbf{c}'_{\mathrm{NAND}}$ is

$$\begin{aligned} e'_{\mathrm{NAND}} &= \mathbf{c}_{\mathrm{NAND}} \cdot \mathbf{v} - \left(1 - m'_1 m'_2\right)\frac{q}{2} + \sum_{i=1}^{N} e_{ks,i} \\ &= e_{\mathrm{NAND}} + \sum_{i=1}^{N} e_{ks,i} \end{aligned} \tag{28}$$

Let $\mathbf{c}_{re}$ denote the error vector brought about by the rounding operation in (21); we have

$$\mathbf{c}_{re} = \left\lfloor \frac{q'}{q}\mathbf{c}'_{\mathrm{NAND}} \right\rceil - \frac{q'}{q}\mathbf{c}'_{\mathrm{NAND}} \in \left(-\frac{1}{2}, \frac{1}{2}\right]^{n'+1} \tag{29}$$

The noise introduced by modulus switching is

$$e_{re} = \mathbf{c}_{re,-(n+1)} \cdot \mathbf{s}' + c_{re,n+1} \tag{30}$$

where $\mathbf{c}_{re,-(n+1)}$ is the subvector extracted from all the coefficients in $\mathbf{c}_{re}$ except the $(n+1)$-th coefficient, and $c_{re,n+1}$ is the $(n+1)$-th coefficient in $\mathbf{c}_{re}$. Then the noise of $\mathbf{c}''_{\mathrm{NAND}}$ can be expressed as

$$e''_{\mathrm{NAND}} = \frac{q'}{q}\left(e_{\mathrm{NAND}} + \sum_{i=1}^{N} e_{ks,i}\right) + e_{re} \tag{31}$$

According to (11), the upper bounds for e'_1, e'_2 are both $(N+1)B_0$, where B_0 is the upper bound of the fresh ciphertexts generated in (17). Under an appropriate parameter setting, we further have

$$\max\left\{\left|e'_1\right|, \left|e'_2\right|\right\} \le (N+1)B_0 < \frac{q}{32} \tag{32}$$

Then it can be learned from (26), (31), and (32) that the noise magnitude in $\mathbf{c}''_{\mathrm{NAND}}$ satisfies

$$\left|e''_{\mathrm{NAND}}\right| < \frac{3q'}{16} + \frac{q'}{q}\left|\sum_{i=1}^{N} e_{ks,i}\right| + \left|e_{re}\right| \tag{33}$$

All the random noises $e_{ks,i}$ $(i = 1, \ldots, N)$ are drawn from an identical discrete Gaussian distribution. The discrete Gaussian distribution can be considered as the corresponding continuous Gaussian distribution with all the instances rounded down to the nearest integer. Assuming N random real numbers are generated from a continuous Gaussian distribution with zero mean and standard deviation σ, their sum also follows a Gaussian distribution with zero mean and standard deviation $\sqrt{N}\sigma$. The probability of the above sum lying in the interval $[-6\sqrt{N}\sigma, 6\sqrt{N}\sigma]$ satisfies $p_2 > 1 - 10^{-8}$. As the downward rounding has an error magnitude of at most N, we have

$$\left|\sum_{i=1}^{N} e_{ks,i}\right| \le 6\sqrt{N}\sigma + N \tag{34}$$

Under an appropriate parameter setting, we have

$$\frac{q'}{q}\left|\sum_{i=1}^{N} e_{ks,i}\right| < \epsilon_1 \tag{35}$$

where ϵ_1 is a very small positive number.

For vectors $\mathbf{a}_{ks,i}(i = 1, \ldots, N)$ which are independently and uniformly sampled from $\mathbb{Z}_q^{n'}$, it is clear that $\sum_{i=1}^{N} \mathbf{a}_{ks,i}$ also follows a uniform distribution over $\mathbb{Z}_q^{n'}$. And each coefficient in $\mathbf{c}_{re,-(n+1)}$ follows the uniform distribution over the following set:

$$S_{re} = \left\{-\frac{1}{2} + \delta, -\frac{1}{2} + 2\delta, \ldots, \frac{1}{2} - \delta, \frac{1}{2}\right\} \tag{36}$$

where $\delta = q'/q = 2^{k'-k}$. The uniform distribution over S_{re} can be considered as the continuous uniform distribution over $[-1/2, 1/2]$ with all the instances rounded up to the nearest element in S_{re}. Let u_{sum} denote the sum of $n_r (0 \leq n_r \leq n')$ independent random real numbers from the uniform distribution over $[0, 1]$. The probability density function of u_{sum} is given by

$$f_{n_r}(x) = \frac{1}{(n_r - 1)!} \sum_{j=0}^{k} (-1)^j C_{n_r}^j (x - j)^{n_r - 1}, \quad x \in [k, k+1] \tag{37}$$

where $k \in \{0, 1, \ldots, n_r - 1\}$ and $C_{n_r}^j$ is the number of combinations when choosing j items from n_r items. Thus, the corresponding cumulative distribution function is

$$F_{n_r}(x) = \frac{1}{n_r!} \sum_{j=0}^{k} (-1)^j C_{n_r}^j (x - j)^{n_r}, \quad x \in [k, k+1] \tag{38}$$

Let p_{3,n_r} denote the probability of u_{sum} lying in the interval $[n_r/2 - B_1, n_r/2 + B_1]$; we have

$$p_{3,n_r} = F_{n_r}\left(\frac{n_r}{2} + B_1\right) - F_{n_r}\left(\frac{n_r}{2} - B_1\right) \tag{39}$$

where B_1 is a positive integer. For n_r independent random real numbers from the uniform distribution over $[-1/2, 1/2]$, p_{3,n_r} is the probability of their sum lying in the interval $[-B_1, B_1]$. Let $p_3 = \min\{p_{3,n_r}\}_{n_r=0,1,\ldots,n'}$ be the lowest probability among $p_{3,n_r}(n_r = 0, 1, \ldots n')$. p_3 would be close to 1 as long as B_1 is sufficiently large. And the absolute value of the above sum can be considered as upper bounded by B_1. When the above n_r independent random real numbers are rounded up to the nearest element in S_{re}, an extra error is introduced. The absolute value of the error is upper bounded by $n_r\delta$. As long as δ is sufficiently small, the following is satisfied:

$$n_r\delta \leq n'\delta < \epsilon_2 \tag{40}$$

where ϵ_2 is another very small positive number. Thus we have

$$|e_{re}| < B_1 + \epsilon_2 + \frac{1}{2} \tag{41}$$

According to the requirement for correct decryption, B_1 should satisfy

$$B_1 \leq \frac{q'}{16} - \left(\epsilon_1 + \epsilon_2 + \frac{1}{2}\right) \tag{42}$$

When q' is sufficiently large, p_3 is still guaranteed to be close to 1 even if B_1 is under the above constraint. According to (33), the upper bound of noise magnitude in the ciphertext $\mathbf{c}''_{\text{NAND}}$ is

$$B_2 = \frac{3q'}{16} + \left(B_1 + \epsilon_1 + \epsilon_2 + \frac{1}{2}\right) \leq \frac{q'}{4} \tag{43}$$

Then we have $|e''_{\text{NAND}}| < B_2 \leq q'/4$, and correct decryption is guaranteed. The ciphertext $\mathbf{c}''_{\text{NAND}}$ can be refreshed using the ciphertext refreshing algorithm in DM, and further homomorphic operations can be performed.

Therefore, the correctness of our scheme lies in that the three incidents corresponding to the probabilities p_1, p_2, p_3 are all true. The error rate of our scheme is $p_{\text{NHE},err} = 1 - p_1 p_2 p_3$.

5.2. Security Analysis. We first give a formal definition for the threat/security model of indistinguishability under chosen plaintext attack (IND-CPA) and then conduct a security analysis for our scheme in line with the model. The IND-CPA threat/security model is defined as the following challenge-guess game between the challenger and the adversary:

(i) **Initialization.** The challenger $\mathscr{C}$ runs the Keygen algorithm to obtain the public and private keys, $(pk, sk) \longleftarrow \textbf{NHE.KeyGen}(\lambda)$, and sends the public key pk to the adversary $\mathscr{A}$.

(ii) **Challenge.** The adversary $\mathscr{A}$ selects a pair of plaintexts m_0, m_1 and sends them to the challenger. The challenger $\mathscr{C}$ randomly selects a plaintext m_b such that $b \longleftarrow \{0, 1\}$, encrypts the plaintext: $c \longleftarrow \textbf{NHE.Enc}(pk, m_b)$, and then sends the ciphertext c to the adversary $\mathscr{A}$.

(iii) **Guess.** The adversary $\mathscr{A}$ guesses the plaintext on receiving ciphertext c and outputs plaintext $m_{b'}(b' \in \{0, 1\})$. If $b' = b$, then the adversary $\mathscr{A}$ wins the game.

Let $\mathscr{A}(c)$ denote the index (0/1) of the adversary's output plaintext on receiving ciphertext c. The adversary's advantage $\text{adv}(\mathscr{A})$ is defined as the difference between the probabilities that the adversary guesses m_b and m_{1-b}, as shown in (44)

$$\text{adv}(\mathscr{A}) = |\Pr[\mathscr{A}(c) = b] - \Pr[\mathscr{A}(c) = 1 - b]| \tag{44}$$

The scheme is IND-CPA secure if for any polynomial time adversary $\mathscr{A}$, the adversary's advantage $\text{adv}(\mathscr{A})$ is negligible: $\text{adv}(\mathscr{A}) = \text{negl}(\lambda)$.

Generally, the ciphertexts in homomorphic encryption schemes are stored outside the local storage. Thus, the storage providers, such as cloud service providers and remote servers, might be the direct potential adversaries. Moreover, there are eavesdroppers who are trying to steal the stored data. And there may be coconspirators with an untrusted storage provider who get the stored data from the untrusted storage provider. They might also be the potential adversaries. In our scheme, both the public key and ciphertexts can be revealed to them.

Thus, it is common for the adversaries to conduct chosen plaintext attacks (CPA). The IND-CPA security of our scheme is analyzed as follows.

It can be learned from (17) that, for the initial ciphertext $\mathbf{C}$, we have $\mathbf{BD}^{-1}(\mathbf{C})=m\mathbf{G}+\mathbf{A}$ where $\mathbf{G}=\mathbf{BD}^{-1}(\mathbf{I}_N)$. As $\mathbf{BD}^{-1}(\mathbf{C})$ can be transformed to $\mathbf{C}$ via BitDecomp, $\mathbf{C}$ is secure if $\mathbf{BD}^{-1}(\mathbf{C})$ effectively hides the plaintext m[19]. The matrix $\mathbf{A} = [\mathbf{b} \parallel \mathbf{B}] \in \mathbb{Z}_q^{N\times(n+1)}$ consists of N independent LWE instances $(\mathbf{B}_i\cdot\mathbf{t}+e_i, \mathbf{B}_i)$, $i = 1, \ldots N$ where $\mathbf{B}_i \longleftarrow \mathbb{Z}_q^n, \mathbf{t} \longleftarrow \mathbb{Z}_q^n, e_i \longleftarrow \chi$.

Suppose a polynomial time adversary $\mathscr{A}$ participates in the challenge-guess game as described above. If $\mathscr{A}$ achieves nonnegligible advantage in the game, then the LWE problem can be solved with equivalent advantage. According to the LWE assumption, no polynomial algorithm can solve the LWE problem with nonnegligible advantage. Thus, the adversary's advantage adv($\mathscr{A}$) should be negligible. Our scheme is IND-CPA secure with respect to the initial ciphertexts. For a final ciphertext $\mathbf{c}''_{\mathrm{NAND}}$, it can be regarded as a ciphertext from a LWE symmetric encryption scheme with secret key $\mathbf{s}'$. In this case, the challenger in the challenge-guess game retains the secret key and performs encryption using the secret key. Following the above analysis, it is easy to show that our scheme is also IND-CPA secure with respect to the final ciphertexts. Therefore, our scheme achieves IND-CPA security under the LWE assumption.

5.3. Applicability Analysis. In general, our scheme supports arbitrary operations on encrypted data; it is universally applicable for privacy-preserving computations in the real world, such as financial and medical data analysis. The underlying plaintexts in each homomorphic NAND operation in our scheme are a pair of bits, which are at the lowest level of data granularity. Thus, our scheme is highly flexible and extensible and can be adjusted to various kinds of computations on encrypted data. As our scheme is conceptually simple, it can be easily implemented, deployed, and maintained in real-world applications. Furthermore, the efficiency of our scheme is relatively high, and the efficient ciphertext refreshing algorithm in DM can be utilized in our scheme for efficient computation on encrypted data in real-world applications.

6. Performance Comparison

In this section, the homomorphic operations in DM, GSW, and our scheme are performed twice on a depth-2 binary circuit with NAND gates. We first present an analysis for the computational costs and error rates of the three schemes. Then based on the above analysis, we present a comparison for the three schemes in terms of computational costs and error rates. To avoid name clashes, the parameters in each scheme are all local to the scheme and apply only to the scheme.

6.1. Computational Cost of DM. For a pair of fresh ciphertexts $\mathbf{c}_0, \mathbf{c}_1 \in \mathbb{Z}_q^{n+1}$, the number of additions needed in the homomorphic operation in (13) is

$$n_{\mathrm{DM},1} = n + 2 \tag{45}$$

It can be learned from **DM**.**Refresh** that **Incr**(ACC,**C**) is performed nd_r times. The operation in **Incr**(ACC,**C**) can be simplified as the multiplication between the 2nd row in ACC and the ciphertext $\mathbf{C}$.

The above multiplication needs 2 inner products between a pair of $2d_g$-dimensional vectors in $R_{\mathrm{Q}}^{2d_g}$. The fast Fourier transform (FFT) of the coefficient vector of each polynomial in R_Q with maximum degree N can be represented as a vector in $\mathbb{C}^{N_2}$ where $N_2 = N/2 + 1$. Inner product between a pair of vectors in $R_{\mathrm{Q}}^{2d_g}$ needs $2N_2d_g$ additions and $2N_2d_g$ multiplications on complex numbers. Each multiplication on complex numbers needs 4 multiplications and 2 additions on real numbers, and each addition on complex numbers needs 2 additions on real numbers. As multiplication generally takes a longer time than addition, each multiplication on complex numbers needs at least 6 additions. Therefore, the number of additions needed in **Incr**(ACC,**C**) is at least

$$n_{\mathrm{DM},2} = 2 \cdot (6 + 2) \cdot 2N_2d_g = 32N_2d_g \tag{46}$$

The key switching in the 3rd step of Algorithm 2 needs Nd_{ks} additions on $(n + 1)$-dimensional vectors. Here $d_{ks} = \lceil \log_{B_{ks}} q \rceil$ and B_{ks} is the base for encoding ciphertexts, as illustrated in DM [21]. Thus the total additions needed in key switching is

$$n_{\mathrm{DM},3} = Nd_{ks}(n + 1) \tag{47}$$

The number of additions needed in the next homomorphic operation is the same as (45). As some other steps are omitted here, a lower bound is obtained for the number of needed operations. According to (45)~(47), the number of additions needed in DM is at least

$$\begin{aligned} n_{\mathrm{DM}} &= 2n_{\mathrm{DM},1} + nd_r \cdot n_{\mathrm{DM},2} + n_{\mathrm{DM},3} \\ &= 32N_2nd_gd_r + (n + 1)\left(Nd_{ks} + 2\right) + 2 \end{aligned} \tag{48}$$

6.2. Computational Cost of Our Scheme. For fresh ciphertexts $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{Z}_q^{N\times N}$, the first homomorphic NAND operation in (18) can be simplified as

$$\mathbf{C}'_{l-2} = \mathbf{FL}\left(\mathbf{I}_{N,l-2} - \mathbf{C}_{1,l-2}\mathbf{C}_2\right) \tag{49}$$

where $\mathbf{I}_{N,l-2}, \mathbf{C}_{1,l-2}$ are the $(l - 2)$-th rows of the matrices $\mathbf{I}_N, \mathbf{C}_1$, respectively. Let $\mathbf{C}_{i,l+1:N}^{l+1:N} \in \mathbb{Z}_q^{nl\times nl}$ denote the submatrix extracted from the $(l+1)$-th to the N-th rows and columns of $\mathbf{C}_i (i = 1, 2)$. It is clear that each coefficient in $\mathbf{C}_{i,l+1:N}^{l+1:N} (i = 1, 2)$ follows a uniform distribution over $\{0, 1\}$. Let $\mathbf{C}_{1,l-2}^{l+1:N} \in \mathbb{Z}_q^{nl}$ denote the subvector extracted from the $(l+1)$-th to the N-th coefficients in $\mathbf{C}_{1,l-2}$. It is clear that each coefficient in $\mathbf{C}_{1,l-2}^{l+1:N}$ also follows a uniform distribution over $\{0, 1\}$.

Let $n_{add,i}$ denote the number of additions needed in the multiplication between $\mathbf{C}_{1,l-2}^{l+1:N}$ and the i-th column in $\mathbf{C}_{2,l+1:N}^{l+1:N}$, $i = 1, \ldots, nl$. For the above operation, we need to add 1 to the intermediate result only when both coefficients being multiplied are nonzero. Thus, the probability of needing 1 addition for the multiplication between each pair of coefficients is 1/4. Thus $n_{add,i}$ follows the binomial distribution $\mathscr{B}(nl, 1/4)$. Let p_4 denote the probability of $n_{add,i}$ being no more than

n_b ($n_b < nl$). And let $p_5 = p_4^{nl}$ denote the probability that $n_{add,i}$ is no more than n_b for each $i = 1, \ldots, nl$. When n_b is sufficiently large, both p_4 and p_5 would be close to 1, the multiplication between $\mathbf{C}_{1,l-2}^{l+1:N}$ and each column in $\mathbf{C}_{2,l+1:N}^{l+1:N}$ can be simplified as at most n_b additions.

For the multiplication between the other coefficients in $\mathbf{C}_{1,l-2}$ and $\mathbf{C}_2$, an upper bound for the number of needed additions can be derived assuming 1 addition for each corresponding coefficient pair:

$$n'_{add} = Nl + l(N-l) = (2N-l)l \tag{50}$$

Considering the substraction between $\mathbf{I}_{N,l-2}$ and $\mathbf{C}_{1,l-2}\mathbf{C}_2$, an upper bound for the number of needed additions in the 1st homomorphic NAND operation is obtained:

$$n_{\mathrm{NHE},1} = n_b nl + n'_{add} + 1 = (n_b + l)(N-l) + Nl + 1 \tag{51}$$

In the following $\mathbf{FL}(\cdot)$ operation, $\mathbf{BD^{-1}}(\cdot)$ is performed followed by a $\mathbf{BD}(\cdot)$ operation. In $\mathbf{BD^{-1}}(\cdot)$, multiplying a power of 2 is just a shift operation, which generally takes less time than addition. Regarding each shift operation as an addition, an upper bound for the amount of computation can be derived. According to (3), $(n+1)(l-1)$ shift operations and $(n+1)(l-1)$ additions are needed in total in the $\mathbf{BD^{-1}}(\cdot)$ operation. In the following $\mathbf{BD}(\cdot)$ operation, 2 shifts and 1 addition are needed for each bit generated from $\mathbf{BD}(\cdot)$. The 2 shifts correspond to shifting right then left by 1 bit each, and the addition corresponds to the subtraction between the original data and the data after the 2 shifts. The amount of computation needed for generating each bit in $\mathbf{BD}(\cdot)$ is upper bounded by 3 additions. Thus the number of additions needed in the flatten operation is upper bounded by the following:

$$n_{\mathrm{NHE},2} = 2(n+1)(l-1) + 3N = (n+1)(5l-2) \tag{52}$$

According to (19), the number of additions in the 2nd homomorphic NAND operation is

$$n_{\mathrm{NHE},3} = 2N \tag{53}$$

And the following $\mathbf{FL}(\cdot)$ operation again needs at most $n_{\mathrm{NHE},2}$ additions. According to (20), the number of additions needed in key switching is

$$n_{\mathrm{NHE},4} = N(n' + 1) \tag{54}$$

As $q'/q = 2^{k'-k}$, modulus switching for each coefficient in the vector is just the process of shifting right for $k - k'$ bits and then adding 1 or 0 to the lowest bit before the binary point. Thus the amount of computation in modulus switching for each coefficient can be represented as 2 additions. The number of additions needed in modulus switching is

$$n_{\mathrm{NHE},5} = 2(n' + 1) \tag{55}$$

Therefore, the number of additions needed in our scheme is upper bounded by

$$\begin{aligned} n_{\mathrm{NHE}} &= \sum_{i=1}^{5} n_{\mathrm{NHE},i} \\ &= (n_b + n' + 2l + 3)N + (10n - n_b - l + 10)l \\ &\quad + (2n' - 4n - 1) \end{aligned} \tag{56}$$

6.3. Computational Cost of GSW. In GSW, as the ciphertext needs to maintain the matrix structure in the 2nd homomorphic operation, the 1st homomorphic operation should be performed as matrix operation. The encryption algorithm of GSW is modified as that in our scheme for a better contrast. The computational cost for the multiplication between each row of a matrix and another matrix is shown in (51). And the computational cost of the 2nd homomorphic operation in GSW is omitted here. A lower bound is then obtained for the computational cost of GSW. From (51), it can be learned that the computational cost of GSW is at least

$$n_{\mathrm{GSW}} = N[(n_b + l)(N-l) + Nl + 1] \tag{57}$$

6.4. Performance Comparison among DM, GSW, and Our Scheme

6.4.1. Parameter Configuration. For convenience, the modulus in each of the above schemes is set to be a power of 2. The parameters in DM are initially set identical to the corresponding simulation parameters in the original DM scheme [21]. The secret key $\mathbf{s}$ in DM is uniformly sampled from $\{0,1\}^n$. For the binary LWE problem, the dimension n should increase to $n \log n$ to ensure equivalent security level as the standard LWE problem [53, 54]. The adversary's advantage is set to $\mathrm{adv} = 2^{-64}$ for all the schemes. For a LWE problem with modulus q, dimension n, and discrete Gaussian noise distribution χ with Gaussian parameter r, the following should be satisfied to guarantee a security level of λ[55]:

$$\begin{aligned} n &\geq \log_2^2\left(\left(\frac{q}{r}\right)\cdot\sqrt{\frac{\ln(1/\mathrm{adv})}{\pi}}\right) \\ &\quad\cdot\frac{\left(\log_2\left(2^\lambda\cdot\mathrm{adv}\right)+110\right)}{7.2q} \end{aligned} \tag{58}$$

According to (58) and the simulation parameters, it can be learned that DM guarantees a security parameter of $\lambda = 58$. When λ changes, the modulus q is kept unchanged, and the dimension n is set as the smallest integer which guarantees a security level λ for the ciphertext. The error rate $p_{\mathrm{DM},biterr}$ for each homomorphic operation in DM can be derived according to the standard deviation β of the Gaussian noise in the refreshed ciphertext [21]. Specifically, $p_{\mathrm{DM},biterr}$ is the probability that the ciphertext noise magnitude does not exceed $q/16$. For depth-2 binary circuit with NAND gates, the error rate of DM is

$$p_{\mathrm{DM},err} = 1 - (1 - p_{\mathrm{DM},biterr})^3 \tag{59}$$

For our scheme, the final modulus q' and dimension n' are set to be the same as those in DM. The standard deviation

Table 2: Computational costs and error rates of DM, GSW, and our scheme.

Security Parameter	58	64	70	76	82	88
$p_{\mathrm{DM},err}(\times 10^{-10})$	0.54	1.95	7.04	22.07	61.43	154.53
$p_{\mathrm{GSW},err}(\times 10^{-5})$	4.27	4.51	4.76	5.00	5.55	5.80
$p_{\mathrm{NHE},err}(\times 10^{-5})$	1.89	2.30	3.18	4.31	6.03	8.52
$n_{\mathrm{DM}}(\times 10^{7})$	5.28	5.61	5.99	6.37	6.75	7.13
$n_{\mathrm{GSW}}(\times 10^{12})$	1.68	1.97	2.31	2.68	3.64	4.16
$n_{\mathrm{NHE}}(\times 10^{7})$	1.66	1.85	2.41	2.65	2.92	3.19

of the Gaussian noise is set to $\sigma = 3$. The modulus and dimension of the initial ciphertext are set according to (32) and (58). Meanwhile, they are set to be as small as possible for efficient operations. n_b is the upper bound for the number of additions needed in the multiplication between $\mathbf{C}_{1,l-2}^{l+1:N}$ and each column in $\mathbf{C}_{2,l+1:N}^{l+1:N}$, as illustrated in Section 6.2. Here n_b is set to be as small as possible under the constraints $p_4 > 1 - 10^{-12}, p_5 > 1 - 10^{-8}$, where p_4, p_5 are the probabilities illustrated in Section 6.2. Thus (51) holds and the computational cost of our scheme is made as low as possible. When the security parameter λ changes, the final modulus q' is kept unchanged, and the final dimension n' is set as the smallest integer which guarantees a security level λ for the final ciphertext. In order to lower the error rate of our scheme, ϵ_1, ϵ_2 are set to be as small as possible under the constraints in (35) and (40). Thus B_1 is made larger under the constraint (42), which promotes the probability p_3. As discussed in the end of Section 5.1, the error rate of our scheme is

$$p_{\mathrm{NHE},err} = 1 - p_1 p_2 p_3 \tag{60}$$

For the GSW scheme, the standard deviation of the Gaussian noise is also set as $\sigma = 3$, as in our scheme. And modulus q and dimension N are set as small as possible under the constraint of correct decryption after 2 homomorphic NAND operations. Meanwhile, the ciphertext should guarantee a security level of λ. As long as the noise of the initial 4 ciphertexts are upper bounded by $B_0 = 6\sigma$, decryption of the final ciphertext would be correct. Therefore, the error rate of GSW is

$$p_{\mathrm{GSW},err} = 1 - p_1 = 1 - p_0^{4N} \tag{61}$$

6.4.2. Performance Comparison. Here a group of security parameters are considered, and the other parameters are set according to the configurations discussed above. $p_{\mathrm{DM},err}, p_{\mathrm{GSW},err}, p_{\mathrm{NHE},err}$ denote the error rates in DM, GSW, and our scheme, respectively, and $n_{\mathrm{DM}}, n_{\mathrm{GSW}}, n_{\mathrm{NHE}}$ denote the corresponding computational costs, as shown in previous discussions. The computational costs and error rates of DM, GSW, and our scheme under different security parameters are shown in Table 2. The error rates are obtained from (59)~(61), and the computational costs are obtained from (48)(56)(57).

From Table 2, it can be learned that the error rates of GSW and our scheme are higher than that of DM. Nevertheless, they can still be considered to be sufficiently low for being lower than 10^{-4}. The main reason is that the noise magnitudes in multiple initial ciphertexts are constrained in a fixed range. Although p_0 is very close to 1, $p_1 = p_0^{4N}$ is much lower than p_0, thus $p_{\mathrm{GSW},err}, p_{\mathrm{NHE},err}$ are both obviously higher than $1 - p_0$. In DM, $p_{\mathrm{DM},biterr}$ is dependent only on the standard deviation of the refreshed ciphertext, which is sufficiently small compared with $q/16$. Thus $p_{\mathrm{DM},biterr}$ is a rather low probability. Moreover, $p_{\mathrm{DM},err}$ depends on the error rates of the 3 homomorphic NAND operations, which is only slightly higher than $p_{\mathrm{DM},biterr}$.

It is also shown in Table 2 that $p_{\mathrm{NHE},err} < p_{\mathrm{GSW},err}$ for small security parameter, and $p_{\mathrm{NHE},err} > p_{\mathrm{GSW},err}$ when the security parameter is sufficiently large. This is because $p_{\mathrm{NHE},err}$ is affected by p_3, the probability of the sum's absolute value being no more than B_1. As the security parameter gets larger, the ciphertext dimension also gets higher, while the upper bound B_1 is kept unchanged. When more random errors are summed up, p_3 would decrease by a certain extent, and the decrease of p_3 is more significant than that of $p_1 = p_0^{4N}$ as the increase of N. With the increase of the security parameter, $p_{\mathrm{NHE},err}$ increases faster than $p_{\mathrm{GSW},err}$.

On the other hand, the overall efficiency of our scheme is significantly higher than that of DM and GSW. Specifically, n_{NHE} is more than 50 percent lower than n_{DM}, and several orders of magnitudes lower than n_{GSW}. The main reason lies in that ciphertext refreshing is removed in our scheme, and the 1st homomorphic operation is simplified as a vector-matrix multiplication. The computational cost is further reduced after considering the uniform randomness on $\{0, 1\}$ for most coefficients. By contrast, the 1st homomorphic operation in GSW is performed as matrix multiplication. And ciphertext dimension in GSW is higher than that of the initial ciphertext in our scheme. In DM, ciphertext refreshing introduces a rather high computational cost.

7. Conclusions

Aiming at the problem of low efficiency caused by overly frequent ciphertext refreshings in DM, we propose a new FHE scheme to achieve a higher efficiency. We utilize ciphertext matrix operations in GSW and ciphertext vector additions in DM to construct our scheme. Furthermore, we combine the advantage of efficient homomorphic operation in DM with the advantage of moderate growth of ciphertext noise magnitude in GSW. Our scheme inherits the conceptual simplicity of DM and GSW and allows 2 homomorphic NAND operations to be performed on ciphertexts before ciphertext refreshing. Results show that under the same security parameters, the computational cost of our scheme

is obviously lower than that in DM and GSW for a depth-2 binary circuit with NAND gates. Thus our scheme is significantly more efficient than DM and GSW. Meanwhile, the error rate of our scheme is kept at a sufficiently low level.

Our work focuses on constructing a simple and efficient FHE scheme based on DM and GSW schemes. We also analyze its correctness, security, and applicability and present a comparison with DM and GSW schemes in terms of computational costs and error rates. Our FHE scheme is intended for universal privacy-preserving computations in the real world. However, our work is limited to the theoretical level. Concrete implementation for our scheme is not considered in our work. And the application of our scheme to real-world algorithms needs to be further explored.

Data Availability

The data for the parameters in the FHE schemes during the current study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China [2016YFF0201003]; the National Natural Science Foundation of China [61571065].

References

[1] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.

[2] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-Trust-a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523–536, 2017.

[3] I. Yaqoob, I. A. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[4] J. Chen and Q. Zhu, "Security as a Service for Cloud-Enabled Internet of Controlled Things under Advanced Persistent Threats: A Contract Design Approach," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2736–2750, 2017.

[5] I. Yaqoob, E. Ahmed, M. H. U. Rehman et al., "The rise of ransomware and emerging security challenges in the Internet of Things," *Computer Networks*, vol. 129, pp. 444–458, 2017.

[6] S. Pokharel, K.-K. R. Choo, and J. Liu, "Mobile cloud security: An adversary model for lightweight browser security," *Computer Standards & Interfaces*, vol. 49, pp. 71–78, 2017.

[7] C. Gentry, *A fully homomorphic encryption scheme*, Stanford University, 2009.

[8] D. Stehlé and R. Steinfeld, "Faster Fully Homomorphic Encryption," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, ryptology - ASIACRYPT 2010*, pp. 377–394, Singapore, 2010.

[9] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography–PKC 2010*, pp. 420–443, Springer, 2010.

[10] N. Ogura, G. Yamamoto, T. Kobayashi, and S. Uchiyama, "An Improvement of Key Generation Algorithm for Gentry's Homomorphic Encryption Scheme," in *Proceedings of the International Workshop on Security, Advances in Information and Computer Security*, pp. 70–83, Kobe, Japan, 2010.

[11] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Proceedings of the International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, pp. 129–148, Springer-Verlag, 2011.

[12] Y. G. Ramaiah and G. V. Kumari, "Towards Practical Homomorphic Encryption with Efficient Public key Generation," *Aceee International Journal on Network Security*, 2012.

[13] C. Jeansébastien, T. Lepoint, and M. Tibouchi, "Scale-Invariant Fully Homomorphic Encryption over the Integers," *Ilar Journal*, vol. 50, no. 4, pp. 361–372, 2014.

[14] J. H. Cheon, J. Kim, M. S. Lee, and A. Yun, "CRT-based fully homomorphic encryption over the integers," *Information Sciences*, vol. 310, pp. 149–162, 2015.

[15] N. Koji and K. Kurosawa, "Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 537–555, Springer, Berlin, Germany, 2015.

[16] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 309–325, ACM, 2012.

[17] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Lecture Notes in Computer Science*, vol. 7417, pp. 868–886, Springer, 2012.

[18] J. Alperin-Sheriff and C. Peikert, "Practical bootstrapping in quasilinear time," in *Advances in Cryptology – CRYPTO*, vol. 8042, pp. 1–20, 2013.

[19] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," *Proceedings of CRYPTO 2013*, vol. 8042, no. 1, pp. 75–92, 2013.

[20] H. Shai and V. Shoup, "Algorithms in HElib," in *Proceedings of the International Cryptology Conference*, pp. 554–571, Springer, Berlin, Germany, 2014.

[21] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 617–640, Springer, Berlin, Germany, 2015.

[22] T. Wu, H. Wang, and Y. P. Liu, "Optimizations of Brakerski's fully homomorphic encryption scheme," in *Proceedings of the 2nd International Conference on Computer Science and Network Technology, ICCSNT 2012*, pp. 2000–2005, December 2012.

[23] X. Zhang, C. Xu, C. Jin, R. Xie, and J. Zhao, "Efficient fully homomorphic encryption from RLWE with an extension to a threshold encryption scheme," *Future Generation Computer Systems*, vol. 36, pp. 180–186, 2014.

[24] C. Ma, J. Li, and G. Du, "A Flexible Fully Homomorphic Encryption," *Wireless Personal Communications*, vol. 95, no. 2, pp. 1–12, 2016.

[25] Z. Li, C. Ma, G. Du, and O. Weiping, "Dual LWE-based fully homomorphic encryption with errorless key switching," in *Proceedings of the 22nd IEEE International Conference on Parallel and Distributed Systems, ICPADS*, pp. 1169–1174, December 2016.

[26] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Proceedings of the International Cryptology Conference*, pp. 297–314, Springer, Berlin, Germany, 2014.

[27] H. Shai and V. Shoup, "Bootstrapping for HElib," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 641–670, Springer, Berlin, Germany, 2015.

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

[29] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in cryptology—EUROCRYPT 2010*, vol. 6110, pp. 24–43, Springer, Berlin, Germany, 2010.

[30] B. Zvika and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE," in *Foundations of Computer Science IEEE*, pp. 97–106, 2011.

[31] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled fhe from learning with errors," in *Proceedings of the Annual Cryptology Conference*, pp. 630–656, Springer, Berlin, Germany, 2015.

[32] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key fhe," in *Advances in Cryptology – EUROCRYPT*, vol. 9666, pp. 735–763, Springer, Berlin, Germany, 2016.

[33] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, "Content-based image retrieval in homomorphic encryption domain," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2015*, pp. 2944–2947, August 2015.

[34] V. Anand and S. C. Satapathy, "Homomorphic encryption for secure information retrieval from the cloud," in *Proceedings of the 1st International Conference on Emerging Trends in Engineering, Technology and Science, ICETETS 2016*, pp. 1–5, February 2016.

[35] M. Nie, P. Ran, and H. Yang, "Efficient Multi-keyword Ranked Search over Outsourced Cloud Data based on Homomorphic Encryption," in *Proceedings of the 2016 8th International Conference on Computer and Automation Engineering, ICCAE 2016*, vol. 56, March 2016.

[36] F. Chen, "Privacy preserving image retrieval method based on binary SIFT and homomorphic encryption," *Transducer Microsystem Technologies*, 2017.

[37] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-Based Approximate Constrained Shortest Distance Queries over Encrypted Graphs with Privacy Protection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 940–953, 2018.

[38] M. A. Will, B. Nicholson, M. Tiehuis, and R. K. L. Ko, "Secure voting in the cloud using homomorphic encryption and mobile agents," in *Proceedings of the 3rd International Conference on Cloud Computing Research and Innovation, ICCCRI 2015*, pp. 173–184, Singapore, October 2015.

[39] S. M. Anggriane, S. M. Nasution, and F. Azmi, "Advanced e-voting system using Paillier homomorphic encryption algorithm," in *Proceedings of the 1st International Conference on Informatics and Computing, ICIC 2016*, pp. 338–342, October 2016.

[40] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han, "A Secure Verifiable Ranked Choice Online Voting System Based on Homomorphic Encryption," *IEEE Access*, pp. 20506–20519, 2018.

[41] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "A Homomorphic LWE Based E-voting Scheme," in *Post-Quantum Cryptography*, pp. 245–265, Springer International Publishing, 2016.

[42] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, "Private machine learning classification based on fully homomorphic encryption," *IEEE Transactions on Emerging Topics in Computing*, 2018.

[43] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang, "Secure Logistic Regression Based on Homomorphic Encryption: Design and Evaluation," *JMIR Medical Informatics*, vol. 2, 2018.

[44] T. P. Le, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," *IEEE Transactions on Information Forensics & Security*, vol. 99, pp. 1-1, 2018.

[45] D. Nathan, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy," *Radio and Wireless Symposium IEEE*, pp. 76–78, 2016.

[46] H. Ehsan, H. Takabi, and M. Ghasemi, *CryptoDL: Deep Neural Networks over Encrypted Data*, 2017.

[47] R. Mukundan, "Efficient integrity verification of replicated data in cloud," *Dissertations Theses - Gradworks*, 2013.

[48] S. Rajat and S. Dey, "Cloud Audit: A Data Integrity Verification Approach for Cloud Computing," *Procedia Computer Science*, vol. 89, pp. 142–151, 2016.

[49] S. Tonyali, K. Akkaya, N. Saputro, and A. S. Uluagac, "A reliable data aggregation mechanism with Homomorphic Encryption in Smart Grid AMI networks," in *Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016*, pp. 550–555, USA, January 2016.

[50] H. Hayouni and M. Hamdi, "A Data Aggregation Security Enhancing Scheme in WSNs Using Homomorphic Encryption," *Intelligent Automation and Soft Computing*, pp. 1–9, 2017.

[51] Y. Yao, J. Wei, J. Liu, and R. Zhang, "Efficiently secure multiparty computation based on homomorphic encryption," in *Proceedings of the 4th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2016*, pp. 343–349, August 2016.

[52] L. Wu, X. Du, and J. Wu, "Effective defense schemes for phishing attacks on mobile computing platforms," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6678–6691, 2016.

[53] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pp. 575–584, 2013.

[54] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with Small Parameters," in *Lecture Notes in Computer Science*, vol. 8042, pp. 21–39, Springer, 2013.

[55] C. Z. Gang, Y. F. Shi, and X. X. Song, "Estimating Concert Security Parameters of Fully Homomorphic Encryption," *Journal of Cryptologic Research*, 2016.