Research Article

Efficient GSW-Style Fully Homomorphic Encryption over the Integers

Jianan Zhao (✉), Ruwei Huang (✉), and Bo Yang (✉)

School of Computer and Electronic Information, Guangxi University, Nanning 530004, China

Correspondence should be addressed to Jianan Zhao; 526957608@qq.com and Ruwei Huang; ruweih@126.com

Received 24 June 2020; Revised 9 February 2021; Accepted 20 February 2021; Published 5 March 2021

Academic Editor: Angel M. Del Rey

Copyright © 2021 Jianan Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a GSW-style fully homomorphic encryption scheme over the integers (FHE-OI) that is more efficient than the prior work by Benarroch et al. (PKC 2017). To reduce the expansion of ciphertexts, our scheme consists of two types of ciphertexts: integers and vectors. Moreover, the computational efficiency in the homomorphic evaluation can be improved by hybrid homomorphic operations between integers and vectors. In particular, when performing vector-integer multiplications, the evaluation has the computational complexity of $O(c \log c)$ and thus outperforms all prior FHE-OI schemes. To slow down the noise growth in homomorphic multiplications, we introduce a new noise management method called sequentialization; therefore, the noise in the resulting ciphertext increases by a factor of $l \cdot \text{poly}(\lambda)$ rather than $\text{poly}(\lambda)^l$ in general multiplications, where $l$ is the number of multiplications. As a result, the circuit with larger multiplicative depth can be evaluated under the same parameter settings. Finally, to further reduce the size of ciphertexts, we apply ciphertext truncation and obtain the integer ciphertext of size $O(\lambda \log \lambda)$, thus additionally reducing the size of the vector ciphertext in Benarroch’s scheme from $O(\lambda^4)$ to $O(\lambda^2 \log^2 \lambda)$.

1. Introduction

Fully homomorphic encryption (FHE) allows us to evaluate any computable function on plaintext $m$ but only manipulate the ciphertexts; moreover, this procedure will not reveal any private information. With this property, FHE can be applied to outsourcing computation, particularly in the scenario that the remote server is an untrusted third party.

FHE has received major attention in cryptography since the first breakthrough by Gentry [1]. In Gentry’s blueprint, the first step is to construct a somewhat homomorphic encryption (SWHE) scheme based on ideal lattices, which can only evaluate a circuit of low depth since the ciphertext contains a noise component that grows exponentially after homomorphic multiplications. If the noise magnitude passes a predefined threshold, the correctness of decryption will no longer remain. Then, by a key procedure called bootstrapping, the noise magnitude in the ciphertext can be reduced to the same level as that in the original ciphertext, and therefore, the SWHE scheme can be converted into an FHE scheme to evaluate a circuit of arbitrary depth.

However, the costly bootstrapping procedure becomes a bottleneck in making this FHE scheme practical, and the complicated operations on ideal lattices become incomprehensible. Very shortly afterward, van Dijk, Gentry, Halevi, and Vaikuntanathan (DGHV) [2] proposed an alternative FHE scheme over the integers (FHE-OI) that is conceptually simpler than [1] since the homomorphic operations only consist of arithmetic operations on integers instead of ideal lattices. In addition, the scheme is based on a different hardness assumption, namely, the approximate greatest common divisor (AGCD) assumption [3]. Unfortunately, this scheme is still inefficient since it follows Gentry’s blueprint, which has a costly bootstrapping procedure. Moreover, the large size of public keys results in an increased storage requirement.

In order to improve the efficiency and avoid bootstrapping, a new noise management technique called modulus switching was proposed by Brakerski et al. [4]. The authors first constructed an SWHE scheme based on the learning with errors (LWE) assumption. Then, by applying modulus switching, the noise component in the ciphertext
grows only linearly after homomorphic multiplications. As a result, they obtained a leveled FHE scheme that can evaluate any circuit with polynomial depth $L$ by carefully calibrating $L$ ladders of moduli. Later, this technique was adapted to the FHE-OI scheme [5]; however, it leads to a greater storage requirement since the circuit with multiplicative depth $L$ needs to be evaluated via a procedure of key switching, and this requires storing $L$ public keys. The larger size of public keys is an inherent property in AGCD-based FHE schemes compared to LWE-based ones; hence, the public keys of $L$ times larger size yield a huge storage requirement. Afterward, a scale-invariant LWE-based FHE scheme was proposed by Brakerski [6]. In this scheme, the message $m$ is encrypted to a higher bit and the same modulus is used throughout the homomorphic evaluation process. By applying the bit decomposition technique on ciphertexts and the encryptions of secret keys, the author obtained a leveled LWE-based FHE scheme whose noise growth was still linear after homomorphic multiplications. The scale-invariant technique was subsequently applied to the FHE-OI scheme and shown to be more efficient than [5]. At PKC 2014, Coron et al. [7] proposed a scale-invariant FHE-OI scheme (CLT14) where only an extra multiplication key needed to be stored in the original public keys, which significantly reduced the storage requirement compared with [5]. To ensure the security of the secret key, they encrypted it into a subset sum as the multiplication key and made a circular-secure assumption. At Eurocrypt 2015, Cheon et al. [8] proposed another scale-invariant FHE-OI scheme (CS15), which introduced a dimension-reduction technique to homomorphic multiplications; they also used invalid encryptions of the bits of the secret key as the multiplication key instead of using subset sum, but they still needed a circular-secure assumption.

At Crypto 2013, Gentry, Sahai, and Waters (GSW) [9] proposed a conceptually simpler LWE-based leveled FHE scheme without using modulus switching and evaluation keys; however, the noise growth was still linear with the multiplicative depth. In this scheme, the plaintext bit $m$ is extended into a plaintext vector by multiplying a gadget vector, and then each entry in the plaintext vector is encrypted to a vector ciphertext. Finally, the output ciphertext has the form of a matrix instead of a vector as in prior LWE-based FHE schemes. The homomorphic multiplication procedure is the general multiplicative operation between a ciphertext matrix and a decomposed binary ciphertext matrix. The main contribution of this scheme is the simple and natural homomorphic multiplication procedure other than complicated modulus switching; however, it must generate a matrix ciphertext, which requires larger storage. This new FHE scheme was studied by Benarroch et al. [10], and they proposed a complete GSW-style FHE-OI scheme (BBL17). In this scheme, there is no need to encrypt the secret key as the multiplication key and make the circular-secure assumption; however, like [9], it must expand the ciphertext from integer to vector, and therefore the size of the vector ciphertext becomes a bottleneck to this scheme’s efficiency.

Note that the above FHE schemes are the encryptions of the plaintext $m$ of a single bit, and the message space in these schemes is $m \in \{0, 1\}$. There have been some SIMD approaches to make the FHE-OI scheme more efficient in parallel. In [11], the authors proposed a new technique to encrypt a vector of messages in a single ciphertext based on the Chinese Reminder theorem (CRT), thereby extending the message space from $m \in \{0, 1\}$ to $m \in \{0, 1\}^2$. This technique has been improved by subsequent studies [7, 10]. In [12], the authors proposed an optimized bootstrapping procedure for larger message spaces of a single bit from $m \in \{0, 1\}$ to $m \in \mathbb{Z}_Q$, where $Q$ is the arbitrary prime larger than two. Subsequently, some improvements have been proposed to accelerate this technique and apply it to the homomorphic evaluation procedure [13–19].

From what has been discussed above, we can conclude that current FHE-OI schemes have significantly improved efficiency by adapting new noise management techniques from LWE-based schemes. While CLT14 [7] and CS15 [8] adapt the scale-invariant technique to obtain a leveled FHE-OI scheme and avoid costly bootstrapping procedures, the homomorphic evaluation procedure is still not efficient enough to be practical. Moreover, these schemes need to encrypt the secret key as the multiplication key and make a circular-secure assumption. Meanwhile, BBL17 [10] constructs a GSW-style FHE-OI scheme whose homomorphic multiplication procedure is a general multiplicative operation between vector and matrix, and the multiplication key is not needed. Unfortunately, it must expand the ciphertext from integer to vector, and this expansion renders difficult attempts to make this scheme more efficient. Therefore, how to improve the computational efficiency in homomorphic evaluation and reduce the ciphertext expansion have remained a challenging and open problem.

**Our Contribution.** In this paper, we propose a GSW-style FHE-OI scheme that is similar to BBL17 [10] but has a better spatial and computational efficiency. The main contribution of our scheme is three-fold.

Firstly, our scheme reduces the expansion of ciphertexts by generating two types of ciphertexts (vector and integer ciphertexts), and not only the vector ciphertexts as in BBL17 [10]. Besides, our scheme supports some hybrid homomorphic operations between integer and vector ciphertexts in the homomorphic evaluation procedure, such as vector-vector, vector-integer, and integer-integer operations. Since the vector ciphertext has a size of $O(y^2)$, where $y$ times larger than the integer ciphertext, it is easy to see that the vector-vector operations are more costly than the integer-integer and integer-vector operations. As a result, when performing integer-integer and integer-vector homomorphic operations, the computational efficiency in our scheme is better than that in BBL17 [10]. In particular, when performing vector-vector multiplications, the asymptotic computational complexity in our scheme is identical to the homomorphic multiplications in prior leveled FHE-OI schemes [7, 8, 10], i.e., there is a complexity of $O(y^3 \log y)$. However, when performing vector-vector multiplications, the asymptotic computational complexity in our scheme is $O(y \log y)$, which is better than that of all prior leveled FHE-OI schemes.
Secondly, our scheme displays better noise management since it performs the homomorphic multiplications in sequence. The method of sequentialization in homomorphic multiplications was first proposed in [20, 21] and derived from the observation that the noise growth is asymmetric after the homomorphic multiplications in GSW-style FHE schemes based on LWE. We find that the noise terms in our scheme also have an asymmetric form such as $r_1 + r_2$ after homomorphic multiplications. Hence, if we perform homomorphic multiplications between $L$ ciphertexts with the same noise magnitude $r$ in the sequence, the resulting ciphertext will have a noise magnitude of $O(L \cdot y \cdot r)$ rather than $O(y^2 \cdot r)$ after general multiplications. Therefore, the circuit with a multiplicative depth larger than $L$ can be evaluated under the same parameter settings as general multiplications.

Finally, the size of ciphertexts in our scheme can be further reduced by applying a technique of ciphertext compression called truncation. This technique was first proposed in CS15 [8], and the authors obtained an integer ciphertext of size $O(y - \rho)$. However, they only made a simple and abstract description of their technique and had not constructed a complete scheme. Moreover, they set $y - \rho$ to be small to obtain better efficiency, but this has been proved to be insecure in subsequent works [22]. We adapt this technique to our scheme and generalize it to the vector ciphertext. In addition, we construct a complete scheme and make the tight analysis of noise growth. In order to thwart the optimized OL attack proposed in [22], we set $y - \rho = \lambda \log \lambda + \Omega(L \log \lambda)$, which significantly reduces the size of ciphertexts. Finally, we obtain an integer ciphertext of size $O(\lambda + L)$ and vector ciphertext of size $O(\lambda^2 + L^2)$.

2. Preliminaries

2.1. Notation. We denote the parameters in our scheme by Greek letters (such as $\rho$, $\eta$, $\tau$, $\lambda$). Scalars are denoted by lowercase Latin characters (such as $p$, $q$, $x$, $y$, and $r$). Vectors are denoted by lowercase bold English letters (such as $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{R}$). Matrices are denoted by uppercase English letters (such as $P$, $Q$, $X$, $Y$, and $R$). For two integers $z$ and $p$, we denote by $[z]_p$ the result of $(-p/2, p/2)$ after computing $\mathbf{z} \cdot \mathbf{P}$. For an $n$-dimensional vector $\mathbf{z}$ and an integer $p$, we denote by $[z]_p$ the result of $(-p/2, p/2)^n$ after computing $\mathbf{z} \cdot \mathbf{P}$. For a vector $\mathbf{z}$, we denote the norm $\|z\|$ to be the infinity norm of $z$, i.e., $\|z\|_{\infty} = \max_{i \in \{1, \ldots, n\}} |z_i|$. For some real number $r$, we denote by $\lfloor r \rfloor$ the nearest integer rounded to $r$, and we have $r = r + \delta$, where $\delta \in [-(1/2), (1/2)) \cap \mathbb{R}$. We denote a uniform distribution on a finite set $A$ by $U(A)$. All logarithms in this paper are base two.

2.2. Gadget Vector and Bit Decomposition Function. For some integer $n$, define an $n$-dimensional gadget vector $\mathbf{g} = (2^0, 2^1, \ldots, 2^{n-1})$. Define a decomposition function $\mathbf{g}^{-1} : [0, 2^n] \rightarrow \{0, 1\}^n$ that can decompose an $n$-bit-length integer to its binary representation and output an $n$-dimensional binary vector. For some integer $z$, we have $\mathbf{g}^{-1}(z) = (b_0, b_1, \ldots, b_{n-1})^T$, where $b_i$ is the $i$-th least significant bit of $z$, and $z = \langle \mathbf{g}, \mathbf{g}^{-1}(z) \rangle$. Define an augmented decomposition function $\mathbf{G}^{-1} : [0, 2^n] \rightarrow \{0, 1\}^{n\eta}$ which can decompose an $l$-dimensional vector $\mathbf{z} = (z_0, z_1, \ldots, z_{l-1})$ to its binary representation and output an $(n \times l)$-dimensional binary matrix such that $\mathbf{G}^{-1}(\mathbf{z}) = (\mathbf{g}^{-1}(z_0), \mathbf{g}^{-1}(z_1), \ldots, \mathbf{g}^{-1}(z_{l-1}))$ and $\mathbf{z} = \mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{z})$.

Lemma 1 (leftover hash lemma [2]). Set $x_1, \ldots, x_m \in \mathbb{Z}_M$ uniformly and independently, set $s_1, \ldots, s_m \leftarrow \{0, 1\}$, and set $y = \sum_{i=1}^m s_i x_i \mod M$. Then $(x_1, \ldots, x_m, y)$ is $1/2 \sqrt{M/2^m}$-uniform over $\mathbb{Z}_M^m$.

We can write the above leftover hash lemma as the following version, and we can see the obvious equivalence between the two lemmas.

Lemma 2. Set $\mathbf{x} = (x_1, \ldots, x_m) \leftarrow \mathbb{Z}_M^m$ uniformly and independently, set $\mathbf{s} \leftarrow \{0, 1\}^m$, and let $y = \langle \mathbf{x}, \mathbf{s} \rangle \mod M$. Then $(\mathbf{x}, y)$ is $1/2 \sqrt{M/2^m}$-uniform over $\mathbb{Z}_M^m$.

Lemma 3 (see [10]). Set $\mathbf{x} = (x_1, \ldots, x_m) \leftarrow \mathbb{Z}_M^m$ uniformly and independently, and set $\mathbf{s} \leftarrow \{0, 1\}^{mn}$ for some $n$; and let $y = \mathbf{x} \cdot \mathbf{s} \mod M$. Then $(\mathbf{x}, y)$ is $1/2 \sqrt{M/2^m}$-uniform over $\mathbb{Z}_M^{mn}$.

2.3. Approximate GCD. Since the first FHE scheme based on the approximate greatest common divisor (AGCD) problem was proposed in DGHV [2], there have been some AGCD variants proposed. The first noise-free AGCD variant was proposed in [23]. It generates a special public key element $x_0 = pq_0$ without noise component $r_0$ and sets $q_0$ to be a random square free $2^l$-rough integer to thwart some attacks based on factorization. Afterward, the first decisional and noise-free AGCD variant was proposed in [11]. In homomorphic evaluation, a noise-free $x_0$ will not introduce an extra noise component and will simplify the noise analysis; however, this could make the scheme vulnerable to thwart quantum attacks. Therefore, to improve the security, some decisional-AGCD variants with a noise component in $x_0$ were proposed in [7, 8, 10]. In the present work, we consider the decisional noisy variant proposed in BBL17 [10], and formally define it as follows.

Definition 1. Let $\rho, \eta, \tau, p$ be some integers satisfying $\eta > \tau > \rho > 0$, and $p$ is an $\eta$-bit odd integer. Define the distribution $\mathcal{D}_{\eta, \rho}(p)$ supported over $[0, 2^\rho)$ as follows:

$$\mathcal{D}_{\eta, \rho}(p) = \{\text{Sample } q \leftarrow \mathbb{Z} \cap [0, 2^\rho/p), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : x = p \cdot q + r\}.$$ (1)

Definition 2. $(\rho, \eta, \gamma)$-AGCD. The $(\rho, \eta, \gamma)$-AGCD problem is to find $p$ when given polynomially many samples from $\mathcal{D}_{\eta, \rho}(p)$. The $(\rho, \eta, \gamma)$-decisional-AGCD problem is to distinguish between $\mathcal{D}_{\eta, \rho}(p)$ and the uniform distribution
Let \( X \) be a distribution supported over \( \mathbb{Z} \), and let \( k \in \mathbb{Z} \). The distribution \( X^{\leq k} \) is the distribution \( X \) conditioned on \( X \leq k \). If \( \Pr[X \leq k] = 0 \), then \( X^{\leq k} \) is undefined. Analogously, we can define \( X^{\geq k} \).

**Lemma 4.** Let \( x_0 > 2^{r-1} \). Then, the distribution \( \mathcal{D}_{\eta \rho}(p)^{<x_0} \) and \( U(\{0, 2^r\} \cap \mathbb{Z}) \) are computationally indistinguishable under the \((\rho, \eta, \gamma)\)-decisional-AGCD assumption.

**Proof.** First, we sample \( x_i \leftarrow \mathcal{D}_{\eta \rho}(p)^{<x_0} \) via rejection sampling. It is easy to see that the rejection probability when sampling each \( x_i \) is at most \( 1/2 \) since \( x_0 > 2^{r-1} \). Therefore, the distribution \( \mathcal{D}_{\eta \rho}(p)^{<x_0} \) is efficiently samplable up to a negligible statistical distance with \( \mathcal{D}_{\eta \rho}(p) \) since \( \Pr[(\mathcal{D}_{\eta \rho}(p)^{<x_0}) < x_0] \) is noticeable. Second, we replace the samplings of the \( x_i \)'s by \( x_i \leftarrow U(\{0, x_0\} \cap \mathbb{Z}) \). Similar to the above procedure, we can deduce that the statistical distance between \( U(\{0, x_0\} \cap \mathbb{Z}) \) and \( U(\{0, 2^r\} \cap \mathbb{Z}) \) is negligible. Since the distribution \( \mathcal{D}_{\eta \rho}(p) \) and \( U(\{0, x_0\} \cap \mathbb{Z}) \) are computationally indistinguishable under the \((\rho, \eta, \gamma)\)-decisional-AGCD assumption, we finally conclude that the distribution \( \mathcal{D}_{\eta \rho}(p)^{<x_0} \) and \( U(\{0, x_0\} \cap \mathbb{Z}) \) are computationally indistinguishable.

### 3. Our Basic Scheme

In this section, we first describe the construction of our GSW-style FHE scheme. Then, we analyze the correctness and noise growth, and finally prove its underlying security.

#### 3.1. Construction

Recall that for a specific \( \eta \)-bit odd integer \( p \), we define the distribution as follows:

\[
\mathcal{D}_{\eta \rho}(p) = \{ Sample \ q = Z \cap \{0, 2^\eta \}, \ r = Z \cap (-2^\eta, 2^\eta) : \text{Output } x = p \cdot q + r \}.
\]  

(i) HE.KeyGen(1^λ): Given a security parameter \( \lambda \), choose the parameters \((\rho, \eta, \gamma, \lambda)\) according to \( \lambda \) and some constraints to ensure the correctness and security (see analysis below). Sample uniformly an \( \eta \)-bit odd integer \( p \). Sample an integer \( x_0 \leftarrow 0 \leftarrow \mathcal{D}_{\eta \rho}(p)^{<x_0} \) via rejection sampling. For \( 1 \leq i \leq \tau \), sample \( x_i \leftarrow \mathcal{D}_{\eta \rho}(p)^{<x_0} \) via rejection sampling, and define \( x = (x_0, x_1, \ldots, x_\tau) \). Sample an integer \( y \leftarrow \mathcal{D}_{\eta \rho}(p)^{<x_0} \) and resample until \( y/p \) is odd, write \( y = p'q' + r' \). Output the secret key \( sk = p \) and the public key \( pk = (x_0, x_1, \ldots, x_\tau, y) \).

(ii) HE.EncVec(pk, m): Given a message \( m \in \{0, 1\} \), uniformly sample a matrix \( S \leftarrow \{0, 1\}^\gamma \) and output a \( \eta \)-dimensional vector \( c \):

\[
c = [m \cdot g + xS]_{x_0},
\]  

(iii) HE.EncInt(pk, m): Given a message \( m \in \{0, 1\} \), uniformly sample a vector \( s \leftarrow \{0, 1\}^\gamma \) and output an integer \( c \):

\[
c = \left[m \cdot \frac{\eta}{2} + \langle x, s \rangle \right]_{x_0}.
\]  

(iv) HE.Add(c_i, c_i): Given two ciphertexts \( c_i \) or \( c_i \), for \( i \in \{1, 2\} \), output the following:

Integer-vector addition:

\[
c_{\text{add}} = [c_1 + c_2]_{x_0}.
\]  

Integer-total addition:

\[
c_{\text{add}} = [c_1 + c_2 + g^{-1} \left( \frac{\eta}{2} \right)]_{x_0}.
\]  

(v) BBL.Dec(sk, c): Compute \( f = c \cdot g^{-1} (p/2) \pmod{p} \), and output the following:

\[
m \leftarrow \begin{cases} 1, & \text{if } |f| \geq p/4, \\ 0, & \text{if } |f| < p/4. \end{cases}
\]
(v) HE.Mult($c_i, c_j$): Given two ciphertexts $c_i$ or $c_j$ for $i \in [1, 2]$, output the following:

Vector-vector multiplication:

$$
\mathbf{c}_{\text{mul}} = \left[ c_i \cdot G^{-1}(c_j) \right]_{x_0}.
$$

(12)

Vector-integer multiplication:

$$
\mathbf{c}_{\text{mul}} = \left[ c_i \cdot g^{-1}(c_j) \right]_{x_0}.
$$

(13)

(vi) HE.DecVec($sk, c$): Given a $y$-dimensional vector ciphertext $c$, first compute $c = c \cdot G^{-1}(p/2)$, and then output the following:

$$
m = \left[ \frac{2c}{P} \right]_2.
$$

(14)

(vii) HE.DecInt($sk, c$): Given an integer ciphertext $c$, output the following:

$$
m = \left[ \frac{2c}{P} \right]_2.
$$

(15)

3.2. Analysis of Correctness and Noise Growth. We first prove the correctness of our scheme and analyze the noise growth in encryption, decryption, and homomorphic operations procedures. Then we convert our scheme into a leveled FHE-OI scheme by analyzing the multiplicative circuit depth L. Finally, we show that when performing homomorphic multiplications in sequence, the resulting ciphertext can have a smaller noise growth.

Lemma 5 (Encryption noise of integer ciphertext). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \rightarrow HE.KeyGen(1^n))$ and $c \rightarrow HE.EncInt(pk, m)$ for a message $m \in [0, 1]$, then

$$
c = \left\lfloor \frac{p}{2} \right\rfloor m + r \mod p,
$$

(16)

where $|r| \leq (2\tau + 1)2^\rho$.

Proof. For a public key element vector $x = (x_1, \ldots, x_r)$ and a random vector $s \rightarrow [0, 1]^r$, it is easy to see that $x, s = \sum_{i=0}^{r} x_i s_{x_i}$, where $x_i = pq_i + r_i$, and $s \subseteq \{1, 2, \ldots, r\}$. Denote $\left(\frac{y}{2}\right) = (pq'/2) + (r'/2) + \delta$, where $\delta \in [-1/(2\tau), 1/(2\tau)] \cap \mathbb{R}$; hence, we can write

$$
c = \left\lfloor \frac{p}{2} \right\rfloor m + \langle x, s \rangle \mod p.
$$

(17)

Since $q'$ is odd, we have modulo $p$ and

$$
c = \left\lfloor \frac{p}{2} \right\rfloor m + \sum_{i \in S} r_i - kr_0 + \left(\frac{r'}{2} + \delta\right)m \mod p,
$$

(18)

for some $k \in [0, r] \cap \mathbb{Z}$. As a result, we have $r = \sum_{i \in S} r_i - kr_0 + (r'/2 + \delta)m$, where $|r| \leq 2^\rho + 2^\rho + 2^\rho - 1 + (1/2) \leq (2\tau + 1)2^\rho$.

Lemma 6 (encryption noise of vector ciphertext). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \rightarrow HE.KeyGen(1^n))$ and $c \rightarrow HE.EncVec(pk, m)$ for a message $m \in [0, 1]$, then

$$
c = mg + pq + r,
$$

(19)

where $\|r\| \leq 2\tau \cdot 2^\rho$.

Proof. For a public key element $x_i = pq_i + r_i$ where $i \in [0, r] \cap \mathbb{Z}$, write $x = (x_1, \ldots, x_r) =pq' + r'$, with $q' = (q_1, \ldots, q_r)$ and $r' = (r_1, \ldots, r_r)$. Then, we have

$$
c = [m \cdot g + xS]_{x_0} = m \cdot g + p \cdot q'S + r'S - k(pq_0 + r_0).
$$

(20)

Define $q = q'S - kr_0$ and $r = r'S - kr_0$, and then we have $c = mg + pq + r$. Since $S \in [0, 1]^{\tau \times 1}$ and $k \in [0, r]^{\tau \times 1}$, we have $\|r'S\| \leq 2\tau^2$ and $\|kr_0\| \leq 2\tau^2$. Therefore, $\|r\| = \|r'S - kr_0\| \leq \|r'S\| + \|kr_0\| \leq 2\cdot 2\tau^2$.

Lemma 7 (noise of integer-integer addition). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \rightarrow HE.KeyGen(1^n))$ and $c_1 \rightarrow HE.EncInt(pk, m_1)$ with $c_1 = p/2m_1 + r \mod p$ for $i \in [1, 2]$, then

$$
c_{\text{add}} = \left\lfloor \frac{p}{2} \right\rfloor (m_1 + m_2) + r \mod p,
$$

(21)

where $|r| \leq |r_1| + |r_2| + 2\tau^2$.

Proof. Since $c_1 = p/2m_1 + r \mod p$, we have

$$
c_{\text{add}} = [c_1 + c_2]_{x_0} = \left\lfloor \frac{p}{2} \right\rfloor (m_1 + m_2) + r_1 + r_2 - kr_0 \mod p,
$$

(22)

where $k \in [-1, 1] \cap \mathbb{Z}$. Hence, we have $|r| = |r_1 + r_2 - kr_0| \leq |r_1| + |r_2| + 2\tau^2$.

Lemma 8 (noise of integer-vector addition). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \rightarrow HE.KeyGen(1^n))$, $c_1 \rightarrow HE.EncInt(pk, m_1)$ with $c_1 = p/2m_1 + r \mod p$ and $c_2 \rightarrow HE.EncVec(pk, m_2)$ with $c_2 = m_2 g + pq_2 + r_2$. Then,

$$
c_{\text{add}} = \left\lfloor \frac{p}{2} \right\rfloor (m_1 + m_2) + r \mod p,
$$

(23)

where $|r| \leq |r_1| + y \|r_2\| + 2\tau^2$.

Proof. Write $c_1 = p/2m_1 + r \mod p$, $c_2 = m_2 g + pq_2 + r_2$, and define

$$
c_2 = c_2 \cdot g^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right) = (m_2 g + pq_2 + r_2) g^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right)
$$

$$
= m_2 gg^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right) + pq_2 g^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right) + r_2\ g^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right)
$$

$$
= \left\lfloor \frac{p}{2} \right\rfloor m_2 + \left\lfloor \frac{y}{2} \right\rfloor + r_2\ \ g^{-1}\left(\left\lfloor \frac{y}{2} \right\rfloor\right)
$$

$$
= \left\lfloor \frac{p}{2} \right\rfloor m_2 + \left\lfloor \frac{y}{2} \right\rfloor + \left\lfloor \frac{r'}{2} + \delta\right\rfloor m_2 \mod p,
$$

(24)

$$
= \left\lfloor \frac{p}{2} \right\rfloor m_2 + \left\lfloor \frac{y}{2} \right\rfloor + r_2 \mod p.
$$

(25)
where $|r_2| = |r_2g^{-1}(y/2) + ((r'/2) + \delta)m_2| \leq \gamma \|r_2\| + 2^{\rho - 1} + (1/2) \leq \gamma\|r_2\| + 2^\rho$. Hence, we can write

$$c_{\text{add}} = \left[ c_1 + c_2 \cdot g^{-1}\left(\frac{y}{2}\right)\right]_{x_0} = [c_1 + c_2]_{x_0} \quad (25)$$

$$= \left\lfloor \frac{p}{2}\right\rfloor (m_1 + m_2) + r_1 + r_2 - kr_0 \pmod{p},$$

for some $|k| \in [0, (\gamma + 2)/2] \cap \mathbb{Z}$. As a result, we have $|r| = |r_1 + r_2 - kr_0| \leq |r_1| + \gamma\|r_2\| + 2^\rho + (\gamma + 2)2^{\rho - 1} \leq |r_1| + \gamma\|r_2\| + 2^\rho$.

**Lemma 9** (noise of vector-vector addition). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \leftarrow \text{HE.KeyGen}(1^\lambda))$ and $c_i \leftarrow \text{HE.EncVec}(pk, m_i)$ with $c_i = m_i g + p q_i + r_i$ for $i \in \{1, 2\}$ and $m_i \in \{0, 1\}$. Then,

$$c_{\text{add}} = (m_1 + m_2)g + p q + r,$$  

where $\|r\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Proof.** $c_{\text{add}} = [c_1 + c_2]_{x_0} = (m_1 + m_2)g + p (q_1 + q_2 - kq_0) + r_1 + r_2 - kr_0$, where $k \in [-1, 1] \cap \mathbb{Z}$. Define $q = q_1 + q_2 - kq_0$, and then we have $c_{\text{add}} = (m_1 + m_2)g + p q + r$, where $r = r_1 + r_2 - kr_0$. Hence, we have $\|r\| \leq \|r_1\| + \|r_2\| + \|kr_0\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Lemma 10** (noise of vector-vector multiplication). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \leftarrow \text{HE.KeyGen}(1^\lambda))$ and $c_i \leftarrow \text{HE.EncVec}(pk, m_i)$ with $c_i = m_i g + p q_i + r_i$ for $i \in \{1, 2\}$ and $m_i \in \{0, 1\}$. Then,

$$c_{\text{mult}} = m_1 m_2 g + p q + r,$$  

where $\|r\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Proof.** Write $c_1 = m_1 g + p q_1 + r_1$ and $c_2 = m_2 g + p q_2 + r_2$, and then we have

$$c_{\text{mult}} = m_1 m_2 g + p q + r,$$  

where $\|r\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Proof.** Write $c_1 = m_1 g + p q_1 + r_1$ and $c_2 = m_2 g + p q_2 + r_2$, and then we have

$$c_{\text{mult}} = m_1 m_2 g + p q + r,$$  

where $\|r\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Lemma 11** (noise of vector-integer multiplication). Let $(sk = p, pk = (x_0, x_1, \ldots, x_r, y) \leftarrow \text{HE.KeyGen}(1^\lambda))$ $m_i \in \{0, 1\}, c_i \leftarrow \text{HE.EncVec}(pk, m_i)$ with $c_i = m_i g + p q_i + r_i$ and $r_i \leftarrow \text{HE.EncInt}(pk, m_i)$ with $m_i = p/2m_2 + r_2 \pmod{p}$. Then,

$$c_{\text{mult}} = \left\lfloor \frac{p}{2}\right\rfloor m_1 m_2 + r \pmod{p},$$  

where $\|r\| \leq \|r_1\| + \|r_2\| + 2^\rho$.

**Proof.** Write $c_2 = p/2m_2 + p q_2 + r_2 \pmod{p}$, and then we have

$$c_{\text{mult}} = \left\lfloor \frac{p}{2}\right\rfloor m_1 m_2 + r \pmod{p},$$  

where $r = r_1 g^{-1}(c_2) + m_1 r_2 - kr_0$ and $|k| \in [0, (\gamma + 1)/2] \cap \mathbb{Z}$. As a result, we have $|r| \leq \|r_1\| + \|r_2\| + 2^\rho$.
Lemma 12 (decryption noise of integer ciphertext). Let sk = p HE.KeyGen(1^l) and an integer ciphertext c with c = p/2m + r mod p for a message m ∈ [0, 1]. Then,
\[ \text{HE.DecInt}(sk, c) = \left[ \frac{2c}{p} \right]_2 = m, \]
if |r| < (p/4) − (1/2).

Proof. Write c = p/2m + pq + r mod p. Since p is odd, we have
\[ \left\lfloor \frac{2c}{p} \right\rfloor = \frac{2p + 1/2 \cdot m + pq + r}{p} = m + 2q + \left\lfloor \frac{2r + m}{p} \right\rfloor, \]
which is correctly decrypted to m when 2|r| + 1/p < (1/2). Hence, we have |r| < (p/4) − (1/2).

Lemma 13 (decryption noise of vector ciphertext). Let sk = p HE.KeyGen(1^l) and a vector ciphertext c with c = mg + pq + r for a message m ∈ [0, 1], then,
\[ \text{HE.DecVec}(sk, c) = \left[ \frac{2\left( \mathbf{c} \cdot \mathbf{g}^{-1}(p/2) \right)}{p} \right]_2 = m, \]
if ||r|| < (1/η)((p/4) − (1/2)).

Proof. For a ciphertext c = mg + pq + r, first compute
\[ c = \mathbf{c} \cdot \mathbf{g}^{-1} \left( \left[ \frac{p}{2} \right] \right) = (mg + pq + r) \mathbf{g}^{-1} \left( \left[ \frac{p}{2} \right] \right) \]
\[ = m\mathbf{g} \mathbf{g}^{-1} \left( \left[ \frac{p}{2} \right] \right) + pq \mathbf{g}^{-1} \left( \left[ \frac{p}{2} \right] \right) + r \mathbf{g}^{-1} \left( \left[ \frac{p}{2} \right] \right) \]
\[ = \frac{p}{2} m + pq + r, \]
where q = \mathbf{q} \mathbf{g}^{-1} (p/2) and r = \mathbf{r} \mathbf{g}^{-1} (p/2) with |r| ≤ η||r||. By Lemma 12, the ciphertext c is correctly decrypted to m when |r| < ((p/4) − (1/2)). Hence, we have ||r|| < (1/η) ((p/4) − (1/2)).

The following lemma shows that our HE scheme can be converted to a leveled FHE-OI scheme to evaluate any binary circuit with depth L if the parameters satisfy some constraints.

Lemma 14. Given any binary circuit C with depth L, the HE scheme can be converted to a leveled FHE-OI scheme to evaluate this circuit if
\[ η − \rho ≥ L \log (y + 1) + \log (2τ + 1) + \log η + 2. \]

Proof. Assume that the given circuit C only consists of multiplication gates; then, a ciphertext will undergo L homomorphic multiplications. We first consider the vector-vector multiplications. For i ∈ [1, L], let \( c_i = m_i \mathbf{g} + pq_i + r_i \) be the ciphertext after the i-th homomorphic multiplication, and \( \delta_i \) be a fresh ciphertext. Let \( R_i \) be the upper bound on the noise magnitude ||r_i||. First, we have \( R_0 = 2r_0^2 \) by Lemma 6; then, after the i-th homomorphic multiplication, we have
\[ R_i = (y + 1)R_{i-1} + γ2^ζ \] by Lemma 10. By solving the recurrence equation, we obtain
\[ R_L = (y + 1)^2 (2τ + 1)2^ζ - 2^ζ, \]
which should be smaller than (1/η)((p/4) − (1/2)) for decryption correctness by Lemma 13. Hence, we have η − ρ ≥ L log (y + 1) + log (2τ + 1) + log η + 2.

Note that vector-integer multiplications have a similar noise growth than vector-vector multiplications, and hence, we omit vector-integer multiplication in the analysis of vector-vector multiplication.

Lemma 15 (sequentialization of homomorphic multiplication). Let (sk = p, pk = (x_0, x_1, ..., x_n, y) HE. KeyGen(1^l)) and \( c_i \) HE.EncVec(pk, m_i) with \( \mathbf{c}_i = m_i \mathbf{g} + pq_i + r_i \) for i ∈ [1, k] ∩ Z and \( m_i \in [0, 1] \), where k is an arbitrary positive integer larger than 1. Then, for any fixed multiplication sequence
\[ \mathbf{c} = \left[ \left[ \mathbf{c}_1 \mathbf{G}^{-1}(\left[ \left[ \mathbf{c}_2 \mathbf{G}^{-1}(\cdots (\left[ \left[ \mathbf{c}_{k-1} \mathbf{G}^{-1}(\mathbf{c}_k) \right]_{x_k} \right]_{x_{k-1}} \right]_{x_{k-2}} \right]_{x_2} \right]_{x_1}) \right]_{x_0} \right], \]
where the noise component \( r \) in \( c \) satisfies ||r|| ≤ η||r_0|| + γ||r_1|| + γ||r_2|| + ... + γ||r_{k-1}|| + ||r_k|| + (k − 1)γ2^ζ ≤ (k − 1)(2τ + 1)2^ζ + 2^ζ + 1.

Proof. Define \( \mathbf{c}' = [\mathbf{c}_1 \mathbf{G}^{-1}(\mathbf{c}_2)]_{x_0} \); hence, ||r'|| ≤ γ||r_0|| + ||r_1|| + γ2^ζ by Lemma 13. Define \( c'' = [\mathbf{c}_k \mathbf{G}^{-1}(\mathbf{c}')]_{x_0} \). Then we have ||r''|| ≤ γ||r_0|| + ||r'|| + γ2^ζ. By recursively performing the above procedure, we finally obtain a ciphertext \( \mathbf{c} \) with the noise component \( r \) satisfying ||r|| ≤ γ||r_0|| + γ||r_1|| + γ||r_2|| + ... + γ||r_{k-1}|| + ||r_k|| + (k − 1)γ2^ζ ≤ (k − 1)(2τ + 1)2^ζ + 2^ζ + 1.

The above lemma implies that when we perform homomorphic multiplications between L+1 fresh ciphertexts and undergo at most L multiplications, if we operate these ciphertexts by a fixed multiplication sequence in Lemma 15, then the resulting ciphertext will have a noise growth of \( L(2τ + 1)2^ζ + 2^ζ + 1 \). This growth is in contrast to the noise blowup in general homomorphic multiplications, i.e., \( (γ + 1)^2 (2τ + 1)2^ζ - 2^ζ \) in Lemma 14. This is a unique property in the GSW-style FHE scheme since the noise terms in ciphertexts grow asymmetrically. We can realize better noise management and evaluate a circuit with a multiplicative depth larger than L by applying this multiplication sequence in homomorphic evaluation.

Note that the method of sequentialization in homomorphic multiplication can also be applied to vector-integer multiplication since it has the same asymmetrical property as vector-vector multiplication and the noise growth is similar. Here we omit the analysis of vector-integer multiplication.

3.3. Parameters. We present some constraints in choosing parameters to ensure the correctness and security against known attacks. Let \( L \) be the circuit depth in homomorphic evaluation and let \( \lambda \) be the security parameter.
\(\rho\) is the bit-length of the noise \(r\) in public key elements, and it should satisfy \(\rho \geq \lambda\) to protect against brute force attacks on the noise [3, 5, 24, 25].

\(\eta\) is the bit-length of the secret key \(p\), and it should satisfy \(\eta \geq \rho + \log (\gamma + 1) + \log (2\tau + 1) + \log \eta + 2\) to ensure decryption correctness (see Lemma 14).

\(\gamma\) is the bit-length of the public key elements \(x_o, y\) and it should satisfy \(\gamma \geq \Omega((\log\lambda)(\eta - \rho))\) to thwart different lattice reduction attacks on the AGCD (as studied in [8]), such as orthogonal lattice attacks [2, 26], the simultaneous Diophantine approximation attack [27], and the multivariate polynomial attack [28].

\(\tau\) is the number of integers in the public keys, and it should satisfy \(\tau = \gamma + \Omega(\lambda)\) to be able to use the leftover hash lemma in the security proof.

To satisfy the above constraints, we can take

\[
\rho = \lambda, \\
\eta = \lambda + \Omega(\log\lambda), \\
\gamma = \Omega(L^2\log\lambda), \\
\tau = \gamma + \Omega(\lambda).
\]

(38)

3.4. Semantic Security. We show that the security of our HE scheme can be based on the decisional-AGCD problem; the main proof process is to show that it is difficult to distinguish the ciphertext from a uniform integer module \(x_o\).

Theorem 1. The HE scheme is IND-CPA secure under the \((\rho, \eta, \gamma)\)-decisional-AGCD assumption.

Proof. We will use a three-step hybrid argument to prove that both the integer and vector ciphertexts in our HE scheme have computational indistinguishability and there is no such a probabilistic polynomial-time adversary \(\mathcal{A}\) that can distinguish these ciphertexts.

Hybrid \(H_0\): Let \(pk = (x_o, x, y) \leftarrow HE.KeyGen(1^\lambda)\), \(c \leftarrow HE.Enc(pk, m) = [m \cdot g + xS]_{x_o}\) and \(c' \leftarrow HE.EncInt(pk, m) = [m \cdot y/2 + (x, s)]_{x_o}\) for a message \(m \in \{0, 1\}\). By the leftover hash lemma in Lemmas 2 and 3, the tuple \((x, xS \mod x_o), (x, s) \mod x_o)\) is within an exponentially small statistical distance by \((1/2)\sqrt{x_o/\tau} = \Omega(\lambda)\).

As a result, the tuple \((x, xS \mod x_o), (x, s) \mod x_o, c, c')\) is within an exponentially small statistical distance by \(\Omega(\lambda)\), independent of the underlying plaintext. Hence, we have

\[
Pr_{H_0}[\mathcal{A}(1^\lambda, pk) = 1] - Pr_{H_0}[\mathcal{A}(1^\lambda, pk, c, c') = 1] \leq \Omega(\lambda).
\]

(39)

Hybrid \(H_1\): Let \(U\) be the uniform distribution over \([0, 2^\tau)\) \(\cap Z\). In this hybrid, we replace the \(x_i\) in the public keys by sampling from \(U(0, 2^\tau) \cap Z\). We first sample \(x'_o \leftarrow U(0, 2^\tau) \cap Z\) via rejection sampling. Then, we sample \(x_i \leftarrow U(0, 2^\tau) \cap Z\) via rejection sampling for \(i \in [1, \tau] \cap Z\), write \(x' = (x'_o, \ldots, x'_i)\), and output \(pk' = (x'_o, x', y)\). The resulting public key distribution is computationally indistinguishable from the genuine public key distribution in \(H_0\) by Lemma 4. Hence, we have

\[
Pr_{H_1}[\mathcal{A}(1^\lambda, pk) = 1] - Pr_{H_1}[\mathcal{A}(1^\lambda, pk', c', c') = 1] \leq \Omega(\lambda).
\]

(40)

Hybrid \(H_2\): Define \(c' = [m \cdot g + x'S]_{x'_o}\) and \(c' = [m \cdot y/2 + x', s]_{x'_o}\) for \(m \in \{0, 1\}\). By the leftover hash lemma in Lemmas 2 and 3, the tuple \((x', x'S \mod x'_o), (x', s) \mod x'_o, c, c')\) is within an exponentially small statistical distance by \((1/2)\sqrt{x'_o/\tau} = \Omega(\lambda)\).

As a result, the tuple \((x', x'S \mod x'_o, (x', s) \mod x'_o, c, c')\) is within an exponentially small statistical distance by \(\Omega(\lambda)\), independent of the underlying plaintext. Hence, we have

\[
Pr_{H_1}[\mathcal{A}(1^\lambda, pk) = 1] - Pr_{H_1}[\mathcal{A}(1^\lambda, pk', c', c') = 1] \leq \Omega(\lambda).
\]

(41)

Finally, we conclude that

\[
Pr_{H_1}[\mathcal{A}(1^\lambda, pk, c, c) = 1] - Pr_{H_1}[\mathcal{A}(1^\lambda, pk', c', c') = 1] \leq \Omega(\lambda),
\]

(42)

which proves the IND-CPA security of the HE scheme.

4. Ciphertext Compression

The above HE scheme reduces the ciphertext expansion in BBL17 [10] by generating two types of ciphertexts. Moreover, the computational efficiency in homomorphic evaluation can be improved by hybrid homomorphic operations between integers and vectors. However, the size of the vector ciphertext \(O(\gamma^3)\) is still a bottleneck to making our scheme more efficient.

In this section, we will use a technique of ciphertext compression to reduce the size of both integer and vector ciphertexts. The technique was first proposed in CS15 [8], and we observe that it can also be applied to our HE scheme.

Note that the plaintext bit \(m\) is embedded into the most significant bit of the integer ciphertext \(c\) modulo \(p\), i.e., \(c = p/2m + r\mod p\). For vector ciphertext \(c\), first compute \(c' = c \cdot g^{-1} (p/2)\), then, \(c'\) has the same form as integer ciphertext \(c\) (see the proof in Lemma 13). In this case, some least significant bits of ciphertexts \(c\) and \(c'\) are irrelevant for decryption correctness, and therefore, we can truncate these bits to obtain a ciphertext of smaller size. However, the number of bits we can truncate must be limited to an upper bound since the decryption failure probability becomes overwhelming if we truncate more bits. According to CS15 [8], truncating a ciphertext by \(\rho\) bits can result in optimal performance. We follow this approach and describe the truncation procedure in the HE scheme below.

Define \(N = 2^\rho\). For integer ciphertext \(c\), define \(\hat{c} = c/N\) so that \(|N\hat{c} - c| \leq N/2\). For vector ciphertext \(c\), define \(\hat{c} = (c_i/N)^{\gamma_i} \mod p\) for \(1 \leq i \leq \gamma\), where \(c_i\) is the \(i\)-th coordinate in vector \(c\).
In the following, we formally present the truncated version of the HE scheme (we call this truncated scheme TLE).

(i) [THE Encryption]

(1) For encryption of the integer ciphertext, let \( \text{sk} = p, pk = (x_0, x_1, \ldots, x_r, y) \) — HE.KeyGen(1) — and \( c \leftarrow \text{HE.EncInt}(pk, m) \) with \( c = p\cdot 2m + pq + r \mod p \) for a message \( m \in [0, 1] \). Then,

\[
\tilde{c} = \left\lfloor \frac{c}{N} \right\rfloor = \left\lfloor \frac{p\cdot m}{2N} + \frac{p}{N} q + \frac{r}{N} \right\rfloor = \left\lfloor \frac{p}{2} \right\rfloor \frac{m}{N} + \frac{p}{N} q + \frac{r}{N} + \delta,
\]

(43)

where \( \delta \in [-1/2, 1/2] \cap \mathbb{R} \), and the noise component is \( \| (r/N) + \delta \| \leq (2r + 1)/(2\rho/N) + (1/2) \) by Lemma 5.

(2) For encryption of the vector ciphertext, let \( \text{sk} = p, pk = (x_0, x_1, \ldots, x_r, y) \) — HE.KeyGen(1) — and \( c \leftarrow \text{HE.EncVec}(pk, m) \) with \( c = mg + pq + r \) for a message \( m \in [0, 1] \). Write \( c = (c_j)_{j \in [0, 1]} \), and then we have \( c_j = m \cdot 2^j + pq_j + r_j \). Then,

\[
\tilde{c} = \left\lfloor \frac{c_j}{N} \right\rfloor_{p \leq i \leq y-1} = \left\lfloor \left( m \cdot 2^j \cdot N + \frac{p}{N} q_j + \frac{r_j}{N} \right) \right\rfloor_{p \leq i \leq y-1} = \left( m \cdot 2^j \cdot N + \frac{p}{N} q_j + \frac{r_j}{N} \right)_{p \leq i \leq y-1} \]

(44)

where \( g' = (2^0, 2^1, \ldots, 2^{r-1}) \), \( \delta' = (\delta_j)_{p \leq i \leq y-1} \), \( \delta_j \in [-1/2, 1/2] \cap \mathbb{R} \), \( q' = (q_j)_{p \leq i \leq y-1} \), \( r' = (r_j)_{p \leq i \leq y-1} \), and the noise component is \( \| (r'/N) + \delta' \| \leq (1/N) \| \delta' \| + \| \delta' \| \leq 2r \cdot (2^r/N) + (1/2) \) by Lemma 6.

(ii) [THE Addition]

(1) For integer-integer addition,

\[
\tilde{c}_{\text{add}} = \left[ \tilde{c}_1 + \tilde{c}_2 \right]_{\rho_0} = \left[ \frac{p}{2} \right] \frac{m_1}{N} + \frac{p}{N} q_{10} + \frac{r_1}{N} + \delta_1 + \left[ \frac{p}{2} \right] \frac{m_2}{N} + \frac{p}{N} q_{20} + \frac{r_2}{N} + \delta_2 - k\left( \frac{p}{N} q_0 + \frac{r_0}{N} + \delta_0 \right)
\]

(45)

where \( q = q_1 + q_2 - kq_0 \), \( r = r_1 + r_2 - kr_0 \), \( \delta = \delta_1 + \delta_2 - k\delta_0 \), \( \delta_0, \delta_1, \delta_2 \in [-1/2, 1/2] \cap \mathbb{R} \), and \( k \in [-1, 1] \cap \mathbb{Z} \). The noise component is \( \| (r/N) + \delta \| \leq (|r_1| + |r_2| + 2^r/N) + |\delta_1| + |\delta_2| + (1/2) \).

(2) For vector-vector addition,

\[
\tilde{c}_{\text{add}} = \left[ \tilde{c}_1 + \tilde{c}_2 \right]_{\rho_0} = \left[ m_1 g' + \frac{p}{N} q' + \frac{r_1}{N} + \delta_1' + m_2 g' + \frac{p}{N} q' + \frac{r_2}{N} + \delta_2' - k'\left( \frac{p}{N} q_0 + \frac{r_0}{N} + \delta_0 \right) \right]
\]

(46)

where \( g' = (2^0, 2^1, \ldots, 2^{r-1}) \), \( \delta' = (\delta_j')_{p \leq i \leq y-1} \), \( \delta_j' \in [-1/2, 1/2] \cap \mathbb{R} \), \( q' = (q_j')_{p \leq i \leq y-1} \), \( r' = (r_j')_{p \leq i \leq y-1} \), and the noise component is \( \| (r'/N) + \delta' \| \leq (1/N) \| \delta' \| + \| \delta' \| \leq 2r \cdot (2^r/N) + (1/2) \) by Lemma 6.
where $q' = q_1' + q_2' - k'q_0$, $r' = r_1' + r_2' - k'r_0$, 
$\delta' = \delta_1' + \delta_2' - k'\delta_0$, and $k' \in [-1, 1]^r \cap \mathbb{Z}^r$.

The noise component is $\|(r'/N) + \delta'\| \leq (\|r_1\| + \|r_2\| + 2r'/N) + \|\delta_1\| + \|\delta_2\| + (1/2)$.

(3) For integer-vector addition, first compute

$$
\tilde{c}_2 = \tilde{c}_2 \cdot g^{-1}(\frac{y}{2N})
$$

$$
= m_2 g'^{-1}(\frac{y}{2N}) + p\frac{q_1'}{N}g'^{-1}(\frac{y}{2N}) + \frac{r_1'g^{-1}(y/2N)}{N} + \delta_2'g^{-1}(\frac{y}{2N})
$$

$$
= \left(\frac{pq' + r'}{2N} + \frac{p}{N}\frac{q_1'}{q_2}g^{-1}(\frac{y}{2N}) + \frac{r_1'g^{-1}(y/2N)}{N} + \delta_2'g^{-1}(\frac{y}{2N})\right) \mod m_2 + \frac{p}{N}q_2 + \frac{r_2}{N} + \delta_2,
$$

where $q_2 = q_1'g^{-1}(y/2N)$, $r_2 = r_1'g^{-1}(y/2N)$ + $((r'2) + N\delta) m_2$, $\delta' \in [-\frac{1}{2}, \frac{1}{2}] \cap \mathbb{R}$ and $\delta_2 = \delta_2'g^{-1}(y/2N)$. We have $\|r_2\| \leq (y - \rho)$

$$
\|r_2\| + 2r' + (N/2) \leq (y - \rho)\|r_2\| + 2r', \quad \|\delta_2\| \leq (y - \rho)\|\delta_2\|. \text{ Then, compute}
$$

$$
\tilde{c}_{add} = \left[\tilde{c}_1 + \tilde{c}_2\right]_{\delta_0}
$$

$$
= \left[\frac{p}{2}\right]\left(\frac{m_1 + m_2}{N}\right) + \frac{p}{N}(q_1 + q_2 - k_0) + \frac{r_1 + r_2 - kr_0}{N} + \delta_1 + \delta_2 - k\delta_0
$$

$$
= \left[\frac{p}{2}\right]\left(\frac{m_1 + m_2}{N}\right) + \frac{p}{N}q + \frac{r}{N} + \delta,
$$

where $q = q_1 + q_2 - k_0$, $r = r_1 + r_2 - kr_0$, $\delta = \delta_1 + \delta_2 - k\delta_0$, and $|k| \in \mathbb{Z}$, $\{y - \rho + 2/2\} \cap \mathbb{Z}$ with $|r| \leq |r_1| + |r_2| + (y - \rho + 2)2r' + |\delta| \leq |\delta_1| + |\delta_2| + (y - \rho + 4)$. The noise component is

$$
\|r_2\| + \|\delta\| \leq (\|r_2\| + (y - \rho)2r'/N) + |\delta| + (y - \rho)\|\delta_2\| + (y - \rho/2).
$$

(iii) [THE Multiplication]

(1) For vector-vector multiplication,

$$
\tilde{c}_{mult} = \left[\tilde{c}_1 \cdot G^{-1}(\tilde{c}_2)\right]_{\delta_0}
$$

$$
= m_1 g'G^{-1}(\tilde{c}_2) + \frac{p}{N}q_1'g^{-1}(\tilde{c}_2') + \frac{r_1'g^{-1}(\tilde{c}_2)}{N} + \delta_1'G^{-1}(\tilde{c}_2) - k'\left(\frac{p}{N}q_0 + \frac{r_0}{N} + \delta_0\right)
$$

$$
= m_1\left(m_2 g' + \frac{p}{N}q_2 + \frac{r_2'}{N} + \delta_2'\right) + \frac{p}{N}(q_1'g^{-1}(\tilde{c}_2') - k'q_0) + \frac{r_1'g^{-1}(\tilde{c}_2') - k'r_0}{N} + \delta_1'G^{-1}(\tilde{c}_2') - k'\delta_0
$$

$$
= m_1 m_2 g' + \frac{p}{N}(q_1'g^{-1}(\tilde{c}_2') + m_1 q_2' - k'q_0) + \frac{r_1'g^{-1}(\tilde{c}_2) + m_1 r_2' - k'r_0}{N} + \delta_1'G^{-1}(\tilde{c}_2) + m_1 \delta_2' - k' \delta_0
$$

$$
= m_1 m_2 g' + \frac{p}{N}q' + \frac{r'}{N} + \delta',
$$

where $\delta_1 = \delta_1' + \delta_2' - k'\delta_0$.
where \( q' = q_i(G^{-1}(c_2)) + m_iq'_4 - k'q_0, \quad r' = r_iG^{-1}(c_2) + m_ir_2 - k'r_0, \quad \delta' = \delta_iG^{-1}(c_2) + m_i\delta_2 - k\delta_0, \) and \( \|k\| \in [0, ((\gamma - \rho + 1)/2)]^{r'N} \cap \mathbb{Z}^{r'N}. \) The noise component is \( \|r'/N\| + \delta' \leq ((\gamma - \rho)\|r'_N\| + \|r'\| + (\gamma - \rho)\|\delta'\| + (\gamma - \rho)/2). \) (2) For vector-integer multiplication,

\[
\tilde{c}_{\text{mult}} = [\tilde{c}_1, \tilde{g}^{-1}(c_2)]_{q_0} = \left(m_1g' + \frac{p}{N}q'_4 + \frac{r'_i}{N} + \delta'_i\right)g^{-1}(c_2) - k\tilde{c}_0
\]

\[
= m_1g'g^{-1}(c_2) + \frac{p}{N}m_1q'_4g^{-1}(c_2) + \frac{r'_i}{N}g^{-1}(c_2) - k\left(\frac{p}{N}q_0 + q_0 + \delta'_0\right)
\]

\[
= m_1\left(\frac{p}{2}\right)\left(m_1 + \frac{p}{N}q_2 + r_2 + \delta_2\right) + \frac{p}{N}\left(q'_4g^{-1}(c_2) - kq_0\right) + \frac{r'_i}{N}g^{-1}(c_2) - k\delta_0
\]

\[
= \left(\frac{p}{2}\right)m_1m_2 + \frac{p}{N}q_2 - m_1q_2 - kq_0 + \frac{r'_i}{N}g^{-1}(c_2) + m_1r_2 - kr_0 + \frac{r'_i}{N}g^{-1}(c_2) - k\delta_0
\]

\[
= \left(\frac{p}{2}\right)m_1m_2 + \frac{p}{N}q + r + \delta,
\]

where \( q = q'_4g^{-1}(c_2) + m_1q_2 - kq_0, r = r'_ig^{-1}(c_2) + m_1r_2 - kr_0, \quad \delta = \delta'_ig^{-1}(c_2) + m_1\delta_2 - k\delta_0, \) and \( |k| \in [0, ((\gamma - \rho + 1)/2)] \cap \mathbb{Z}. \) The noise component is \( |(r'/N) + \delta| \leq ((\gamma - \rho)\|r'_N\| + |r_2| + (\gamma - \rho)2^\|\tilde{m}\|/4N + (\gamma - \rho)\|\delta_2\| + (\gamma - \rho)/2). \)

(iv) [THE Decryption]

(1) For decryption of the integer ciphertext,

\[
\left[\begin{array}{c}
\tilde{c} \\
\frac{p}{N}
\end{array}\right]_2 = \left[\begin{array}{c}
\tilde{c} \\
\frac{p}{N}
\end{array}\right]_2 = \left[\begin{array}{c}
m + 2q + \left[m + 2r + 2N\tilde{m}\right]_2 \\
pN
\end{array}\right],
\]

which is correctly decrypted to \( m \) when \( |m + 2r + 2N\tilde{m}| < (1/2). \) Hence, we have \( |r/N| + \delta < (p/4N) - (1/2N). \)

(2) For decryption of the vector ciphertext, first compute

\[
\tilde{c} = \tilde{c} \cdot g^{-1}\left(\frac{p}{2N}\right) = \left(mg' + \frac{p}{N}q'_4 + \frac{r'}{N} + \delta'\right)g^{-1}\left(\frac{p}{2N}\right)
\]

\[
= m\left(\frac{p}{2}\right)g^{-1}\left(\frac{p}{2N}\right) + \frac{p}{N}q'_4g^{-1}\left(\frac{p}{2N}\right) + \frac{r'}{N}g^{-1}(p/2N)
\]

\[
+ \delta'g^{-1}\left(\frac{p}{2N}\right)
\]

\[
= \left(\frac{p}{2N}\right)m + \frac{p}{N}q'_4 + \frac{r'}{N} + \delta,
\]

where \( q = q'_4g^{-1}(p/2N), r = r'_ig^{-1}(p/2N), \) and \( \delta = \delta'_ig^{-1}(p/2N) \) with \( |r| \leq (\eta - \rho)\|r'\| \) and \( |\delta| \leq (\eta - \rho)\|\delta\|. \) Then, compute

\[
\gamma = \Omega\left(L^2\lambda\log\lambda\right),
\]

\[
\eta = \gamma - \lambda\log\lambda,
\]

\[
\rho = \eta - \Omega\left(L\log\lambda\right).
\]
Table 1: Ciphertext size comparison with across various schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Size of ciphertext</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLT14 [7]</td>
<td>$\mathcal{O}(L^2\lambda + \lambda^2)$</td>
</tr>
<tr>
<td>CS15 [8]</td>
<td>Basic scheme</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(L^2\lambda \log \lambda)$</td>
</tr>
<tr>
<td></td>
<td>Truncated scheme</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(\lambda^4 + L^4)$</td>
</tr>
<tr>
<td>BBL17 [10]</td>
<td>Scalar ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(L^3\lambda \log (2N^2Bb))$</td>
</tr>
<tr>
<td></td>
<td>Vector ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(L^3\lambda^2 \log^2 \lambda)$</td>
</tr>
<tr>
<td></td>
<td>Integer ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(L^2\lambda \log \lambda)$</td>
</tr>
<tr>
<td></td>
<td>Vector ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(L^3\lambda^2 \log^2 \lambda)$</td>
</tr>
<tr>
<td>Our HE</td>
<td>Integer ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\lambda \log \lambda + \Omega(L\log \lambda)$</td>
</tr>
<tr>
<td>Our THE</td>
<td>Vector ciphertext</td>
</tr>
<tr>
<td></td>
<td>$\lambda^3 \log^2 \lambda + \Omega(L^2\log^2 \lambda)$</td>
</tr>
</tbody>
</table>

5. Comparison with Prior Works

In this section, we make a comparison between our scheme and prior schemes, such as CLT14 [7], CS15 [8], BBL17 [10], and Per20b [19]. We show that our scheme has more advantages in terms of both spatial and computational efficiency. Our HE scheme reduces the ciphertext expansion by generating integer and vector ciphertexts instead of only generating vector ciphertexts as in BBL17 [10]. Additionally, the size of both ciphertexts can be further reduced by applying the THE scheme. Compared to CLT14 [7], the vector ciphertexts in the HE scheme have a larger size since the CLT14 [7] scheme only consists of integer ciphertexts; moreover, by applying the THE scheme, the vector ciphertexts are similar to the integer ciphertexts in CLT14 [7]. However, we cannot compare our schemes with Per20b [19] directly since Per20b used a variant called RAGCD and it has a larger message space such that $m \in \mathbb{Z}_m$. However, we can convert Per20b into a general AGCD scheme and set message space to $m \in \{0, 1\}$ by restricting the parameters to some concrete values. Roughly, we can take $N = 1, B = 1, l = y, b = 2$ and set the size of scalar and vector ciphertexts in Per20b equal to $\Omega(L^3\lambda \log \lambda)$ and $y_1 \Omega(L^3\lambda^2 \log^2 \lambda)$. In this way, we can see that our schemes have similar performance as Per20b. Table 1 shows the concrete parameters of the size of ciphertexts in our schemes and prior schemes.

In homomorphic evaluation, since the multiplication procedure takes up most of the running time, it determines the final computational efficiency. In the following, we show that the homomorphic multiplication in our scheme has more advantages than prior schemes.

Table 2: Comparison of multiplication complexity across various schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Multiplication complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLT14 [7]</td>
<td>$\mathcal{O}(\gamma \log \gamma)$</td>
</tr>
<tr>
<td>CS15 [8]</td>
<td>Basic scheme</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(\gamma \log \gamma)$</td>
</tr>
<tr>
<td></td>
<td>Truncated scheme</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}((\gamma - \rho) \log (\gamma - \rho))$</td>
</tr>
<tr>
<td>BBL17 [10]</td>
<td>Vector-Scalar</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(\gamma \log \gamma)$</td>
</tr>
<tr>
<td>Per20b [19]</td>
<td>Vector-Vector</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(\gamma \log \gamma)$</td>
</tr>
<tr>
<td>Our HE</td>
<td>Vector-Integer</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}(\gamma \log \gamma)$</td>
</tr>
<tr>
<td>Our THE</td>
<td>Vector-Vector</td>
</tr>
<tr>
<td></td>
<td>$\mathcal{O}((\gamma - \rho) \log (\gamma - \rho))$</td>
</tr>
</tbody>
</table>

6. Conclusion

In this work, we present a GSW-style FHE-OI scheme and reduce the ciphertext expansion by generating integer and vector ciphertexts. The computational efficiency in homomorphic evaluation is improved by hybrid homomorphic operations between integers and vectors, especially when performing vector-integer multiplications. Our computational complexity is better than those of all prior leveled FHE-OI schemes. By applying sequentialization in homomorphic multiplications, the noise growth in the resulting ciphertext is smaller than that produced through general multiplications, and therefore, our scheme can evaluate a circuit with a larger multiplicative depth under the same parameter settings as general multiplications. Finally, the efficiency in our scheme can be further improved by a technique of ciphertext compression called truncation, and the optimized scheme is as secure as before when it comes to thwarting known attacks. Note that our scheme can also be generalized to SIMD operations such as the batch technique based on Chinese remainder theorem and larger prime message space.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

Ruwei Huang and Bo Yang are contributed equally to this work.

Acknowledgments

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

References


