Efficient Fully Homomorphic Encryption with Circularly Secure Key Switching Process

Zhou Tanping*, Yang Xiaoyuan, Zhang Wei and Wu Liqiang

Key Laboratory of Network & Information Security under the Chinese Armed Police Force,Electronic Department, Engineering College of the Armed Police Force, Xi'an, 710086, China
Email: 850301775@qq.com
Email: xyyangwj@126.com
Email: zhaangweei@yeah.net
Email: 1191724329@qq.com

Abstract: Fully homomorphic encryption (FHE) has important applications in cloud computing. However, almost all fully homomorphic encryption schemes share two common flaws that they all use large-scale secret keys and some operations inefficient. In this paper, the "special b" variant of the Learning With Errors problem (bLWE) is presented, and helps us construct the first circularly secure key switching process which can replace the key switching process and similar re-linearization process used by the existing FHE schemes. Then, we present an efficient FHE. Compared with Brakerski's scheme, our scheme reduces L secret keys to one and is more efficient. Finally, we prove the chosen-plaintext attack (CPA) security of the fully homomorphic scheme and the circular security of key switching process in standard model under the learning with errors problem (LWE) assumption.

Keywords: circular security; fully homomorphic encryption; LWE problem

Biographical notes: Yang Xiaoyuan, born in 1959. PhD supervisor and master. His main research interests include information security and cryptology.

Zhou Tanping, born in 1989. Master candidate. His main research interests include fully homomorphic encryption, encryption scheme based on lattice.

Zhang Wei, born in 1976. PhD and assistant professor. Her main research interests include fully homomorphic encryption, encryption scheme based on lattice.

Wu Liqiang, born in 1986. Master. His main research interests include encryption scheme based on lattice, security proof.

1 Introduction

Recently, cryptography obtained the swift and violent development [1-4]. The idea of homomorphic encryption can be retrospect to 1978 [5], it means that an entity can implement computations on encrypted data without decryption. This character of encryption scheme sounds appealing in network services. The most important application of homomorphic encryption is the outsourcing of data and computation on clouds. Besides these, there are some other interesting applications including database encryption delegate computation, private information retrieval (PIR), electronic voting, and secure multiparty computation [6].

If an encryption scheme can compute any function of the ciphertexts, then it is called a Fully Homomorphic Encryption scheme. Otherwise, if it can only evaluate a limited set of circuits about ciphertexts, then it is called a Somewhat Homomorphic Encryption(SHE) scheme. The substantial progress was achieved by Gentry's first FHE scheme based on ideal lattices in STOC'2009 [7], following many improvements with higher efficiency and better performance. In Eurocrypt 2010, Dijk, Gentry and Halevi et al. [8] promoted another succinct construction of FHE scheme on in-

This work was supported by the National Natural Science Foundation of China (Grant No. 61272492. 61103231. 61103230) .

tegers other than on ideal lattice, called DGHV. In 2011, Brakerski et al. [9] proposed two schemes based on Learning with Errors problem over Rings (RLWE) and two important techniques called re-linearization and dimension-modulus reduction to control noise and the length of encrypted data. In 2011, Brakerski, Gentry and Vaikuntanathan [10] presented a novel approach to FHE that dramatically improves performance and bases security on weaker assumptions, by a much more effective approach for managing the noise level, called BGV. In 2012, Halevi and Shoup programmed BGV by C++ and NTL math kernel library [11]. In Cryptology–CRYPTO 2013, Gentry et al. [12] proposed a scheme, called GSW, based on Learning with Errors problem and a new technique called the approximate eigenvector method. What's more, Gentry et al. constructed the first identity-based FHE scheme and attribute-based FHE scheme.

Considering the efficiency, we will focus on BGV. BGV consists of the scheme based on RLWE and the scheme based on LWE. Based on the Ring LWE assumption, it can reduce the per-gate computation of the bootstrapped version from $\widetilde{\Omega}\left(\lambda^{3.5}\right)$ to $\tilde{O}(\lambda)$, comparing with Previous FHE schemes. Although the scheme based on RLWE is the most effective FHE scheme, up to now, the security is controversial. The scheme based on LWE is not so effective, but it has a more reliable security. This paper improves BGV based on LWE. Compared with Brakerski et.al's scheme based on LWE, our scheme has advantages of a smaller secret key size, easier and faster computation in Inputs Normalization and an efficient FHE.Add [6].

2 Preliminaries

2.1 Basic Notation

Definition 2.1 [13] (B-bounded distributions). A distribution ensemble $\{\chi_n\}_{\mathrm{n}\in\mathbb{N}}$, supported over the integers, is called B-bounded if $\Pr_{e\xleftarrow{S}\chi_n}[|e|>B]\le 2^{-\tilde{\Omega}(n)}$.

Definition 2.2 [14] (Leveled Fully Homomorphic Encryption). We say that a family of homomorphic encryption schemes $\{\varepsilon^{(L)}: L\in\mathbb{Z}^+\}$ is leveled fully homomorphic if, for all $L\in\mathbb{Z}^+$, they all use the same decryption circuit, $\varepsilon^{(L)}$ compactly evaluates all circuits of depth at most L (that use some specified complete set of gates), and the computational complexity of $\varepsilon^{(L)}$'s algorithms is polynomial (the same polynomial for all L) in the security parameter, L, and (in the case of the evaluation algorithm) the size of the circuit.

Definition 2.3 [15] (LWE). For security parameter λ, let $n=n(\lambda)$ be an integer dimension, $q=q(\lambda)\ge 2$ be an integer, and $\chi=\chi(\lambda)$ be a distribution over $\mathbb{Z}$. The $LWE_{n,\mathrm{q},\chi}$ problem is to distinguish the following two distributions: In the first distribution, one samples $(\mathbf{a}_i,\mathrm{b}_i)$ uniformly from $\mathbb{Z}_q^{n+1}$. In the second distribution, one first draws $\mathbf{s}\leftarrow\mathbb{Z}_q^n$ uniformly and then samples $(\mathbf{a}_i,\mathrm{b}_i)\in\mathbb{Z}_q^{n+1}$ by sampling $\mathbf{a}_i\leftarrow\mathbb{Z}_q^n$ uniformly, $e_i\leftarrow\chi$, and setting $b_i=<\mathbf{a}_i,s>+e_i$. The $LWE_{n,\mathrm{q},\chi}$ assumption is that the $LWE_{n,\mathrm{q},\chi}$ problem is infeasible.

Lemma 2.4 [16] (Average-case to Worst-case) Let $n,q\ge 1$ be some integers and χ be some distribution on $\mathbb{Z}_q$. Assume that we have access to a distinguisher W that distinguishes $A_{\mathbf{s},\chi}$ from U for a non-negligible fraction of all possible $\mathbf{s}$. Then there exists an efficient algorithm W' that for all $\mathbf{s}$ accepts with probability exponentially close to 1 on inputs from $A_{\mathbf{s},\chi}$ and rejects with probability exponentially close to 1 on inputs from U.

We let $\Gamma_{d,q}^n$ be the distribution of $\mathbf{a}\in\mathbb{Z}_q^n$ where $(\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1])\xleftarrow{U}\mathbb{Z}_q^{n-1}$, $\mathbf{a}[\mathrm{n}]=d-\sum_{i=1}^{i=n-1}\mathbf{a}[\mathrm{i}](\mathrm{mod}\,q)$.

Definition 2.5(bLWE: LWE with a special b): For security parameter λ, let $n=n(\lambda)$ be an integer dimension, let $q=q(\lambda)\ge 2$ be an integer, $d\ge 1$ be an integer, and let $\chi=\chi(\lambda)$ be a distribution over $\mathbb{Z}$. The $bLWE_{n,q,d,\chi}$ problem is to distinguish the following two distributions: In the first distribution, one samples $((\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1]),\mathrm{b})\in\mathbb{Z}_q^n$ by sampling $(\mathbf{a},\mathrm{b})$ uniformly from $\mathbb{Z}_q^n$. In the second distribution, called $bA_{q,\mathbf{s},\chi}$, one first draws $\mathbf{s}\leftarrow\mathbb{Z}_q^n$ uniformly and then samples $((\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1]),\mathrm{b})\in\mathbb{Z}_q^n$ by sampling $\boldsymbol{a}\leftarrow\Gamma_{d,q}^n$, $e\leftarrow\chi$, and setting $\mathrm{b}=\sum_{i=1}^n\mathbf{a}[\mathrm{i}]\mathbf{s}[\mathrm{i}]+2e$. The $bLWE_{n,q,d,\chi}$ assumption is that the $bLWE_{n,q,d,\chi}$ problem is infeasible (When $d=1$ we represent $bLWE_{n,q,\chi}$ as $bLWE_{n,q,1,\chi}$.)

In order to follow the description of BGV based on LWE, we represent R as $\mathbb{Z}$ denote set of the integers.

2.2 Basic Encryption Scheme based on bLWE

Set $\|\mathbf{a}\|_1$ to be $\sum_{i=1}^n\mathbf{a}[\mathrm{i}]\,\mathrm{mod}\,q$, set $\boldsymbol{a}^{-1}$ to be $\|\boldsymbol{a}\|_1^{-1}\,\mathrm{mod}\,q$.

E.Setup($1^\lambda, 1^\mu$): Choose a μ-bit modulus q and choose the other parameters $(n = n(\lambda,\mu), N = \lceil (2n+1)\log q \rceil, \chi = \chi(\lambda,\mu), d = 1)$ appropriately to ensure that the scheme is based on a LWE instance that achieves 2^λ security against known attacks. Let $params = (q, n, N, \chi)$.

E.SecretKeyGen($params$): Draw $\mathbf{s}' \leftarrow \chi^n$. Set $sk = \mathbf{s} \leftarrow (1, \mathbf{s}'[1], ..., \mathbf{s}'[n]) \in R_q^{n+1}$ .32

E.PublicKeyGen($params, sk$): Takes as input a secret key $sk = \mathbf{s} = (1, \mathbf{s}')$ with $\mathbf{s}[0] = 1$, $\mathbf{s}' \in R_q^n$ and the $params$. Draw $\mathbf{e} \leftarrow \chi^N$. Generate matrix $\mathbf{A}'' \in R_q^{N\times(n-1)}$, matrix $\mathbf{A}' \in R_q^{N\times n}$ and a vector $\mathbf{b}$. Set $\mathbf{a}_i'[j]$ to be the element of i th row and j th column of $\mathbf{A}'$ and $\mathbf{a}_i'$ to be the i th row of $\mathbf{A}'$. Sample $\mathbf{a}_i' \leftarrow \Gamma_{1,q}^n$ for all the $1 \le i \le N$, notice that $\|\boldsymbol{a}_i'\|_1 = 1$. Set $\mathbf{A}'' \in R_q^{N\times(n-1)}$ to be the matrix consisting of the frontal $n-1$ columns of $\mathbf{A}'$. Set $\mathbf{b}_i = <\boldsymbol{a}_i', \boldsymbol{s}> + 2e_i$ to be each element of $\mathbf{b}$. Set $\mathbf{A} = (\mathbf{b} \mid -\mathbf{A}') \in R_q^{N\times(n+1)}$ and $\bar{\mathbf{A}} = (\mathbf{b} \mid -\mathbf{A}'') \in R_q^{N\times n}$ (Observe: $\mathbf{A}\cdot\mathbf{s}$=2$\mathbf{e}$.) It's easy to convert $\bar{\mathbf{A}}$ into $\mathbf{A}$.

Set the public key $pk = \bar{\mathbf{A}}$.

E.Enc($params, pk, m$): To encrypt a message $m \in R_2$, set $\mathbf{m} \leftarrow (m, 0, ..., 0) \in R_q^{n+1}$, sample $\mathbf{r} \leftarrow R_2^N$, find out $\mathbf{A}$ according to $pk = \bar{\mathbf{A}}$ and output the ciphertext $\mathbf{c} \leftarrow \mathbf{m} + \mathbf{A}^T\mathbf{r} \in R_q^{n+1}$.

E.Dec($params, sk, \mathbf{c}$): Output $m \leftarrow \left\lfloor \left[\langle \mathbf{c}, \mathbf{s} \rangle \right]_q \right\rfloor_2$.

Notice that the decryption equation for a ciphertext $\mathbf{c}$ that encrypts m under key $\mathbf{s}$ can be written as $m = \left\lfloor \left[L_{\mathbf{c}}(\mathbf{s}) \right]_q \right\rfloor_2$ where $L_{\mathbf{c}}(\mathbf{x})$ is a ciphertext-dependent linear equation over the coefficients of $\mathbf{x}$ given by $L_{\mathbf{c}}(\mathbf{x}) = \langle \mathbf{c}, \mathbf{x} \rangle$.

Suppose that we have two ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$, encrypting m_1 and m_2 respectively under the same secret key $\mathbf{s}$. The way homomorphic multiplication is accomplished in [13] is to consider the quadratic equation $Q_{\mathbf{c}_1,\mathbf{c}_2}(\mathbf{x}) = L_{\mathbf{c}_1}(\mathbf{x}) L_{\mathbf{c}_2}(\mathbf{x})$. Assuming the noises of the initial ciphertexts are small enough, we obtain $m_1 m_2 = \left\lfloor \left[Q_{\mathbf{c}_1,\mathbf{c}_2}(\mathbf{s}) \right]_q \right\rfloor_2$, as desired. If one wishes, one can view $Q_{\mathbf{c}_1,\mathbf{c}_2}(\mathbf{x})$ as a linear equation $L_{\mathbf{c}_1,\mathbf{c}_2}^{long}(\mathbf{x}\otimes\mathbf{x})$ over the coefficients of $\mathbf{x}\otimes\mathbf{x}$ – that is, the tensor of $\mathbf{x}$ with itself – where $\mathbf{x}\otimes\mathbf{x}$'s dimension is roughly the square of $\mathbf{x}$'s.

2.3 Key Switching

Key Switching stands for a process, which can convert a ciphertext $\mathbf{c}_1$ under the secret key $\mathbf{s}_1$ into a ciphertext $\mathbf{c}_2$ under a different secret key $\mathbf{s}_2$.

BitDecomp($\mathbf{x} \in R_q^n, q$) [10] decomposes $\mathbf{x}$ into its bit representation. Namely, write $\mathbf{x} = \sum_{j=0}^{\lfloor \log q \rfloor} 2^j \cdot \mathbf{u}_j$, where all of the vectors $\mathbf{u}_j$ are in R_2^n, and output $(\mathbf{u_0}, ...\mathbf{u}_{\lceil \log q \rceil}) \in R_2^{n\lceil \log q \rceil}$.

Powersof2($\mathbf{x} \in R_q^n, q$) [10] outputs the vector $(\mathbf{x}, 2\mathbf{x}, ..., 2^{\lfloor \log q \rfloor}\mathbf{x}) \in R_q^{n\lceil \log q \rceil}$.

SwitchKeyGen($\mathbf{s}_1 \in R_q^{n_1}, \mathbf{s}_2 \in R_q^{n_2}$) **:**

1. Run $\bar{\mathbf{A}}_2 \leftarrow$ E.PublicKeyGen($\mathbf{s}_2$, N) for $N = n_1 \lceil \log q \rceil$.

2. Find out $\mathbf{A}_2$ according to $\bar{\mathbf{A}}_2$ and set $\mathbf{B} \leftarrow \mathbf{A}_2 + \text{Powersof2}(\mathbf{s_1}) = (\mathbf{b}_2 + \text{Powersof2}(\mathbf{s_1}) \mid -\mathbf{A}_2')$ (Add Powersof2($\mathbf{s_1}$) $\in R_q^N$ to $\mathbf{A}$'s first column.) Output $\tau_{\mathbf{s}_1\to\mathbf{s}_2} = \mathbf{B}$.

SwitchKey($\tau_{\mathbf{s}_1\to\mathbf{s}_2}, \mathbf{c_1}$) **:** Output $\mathbf{c}_2 = \text{BitDecomp}(\mathbf{c}_1)^T \mathbf{B} \in R_q^{n_2}$.

Lemma 2.6: Let $\mathbf{s}_1, \mathbf{s}_2, q, n_1, n_2, \mathbf{A}_2, \mathbf{B} = \tau_{\mathbf{s}_1\to\mathbf{s}_2}$ be as in SwitchKeyGen($\mathbf{s}_1, \mathbf{s}_2$), and let $\mathbf{A}_2\mathbf{s_2} = 2\mathbf{e}_2 \in R_q^N$. Let $\mathbf{c}_1 \in R_q^{n_1}$ and $\mathbf{c_2} \leftarrow \text{SwitchKey}(\tau_{\mathbf{s}_1\to\mathbf{s}_2}, \mathbf{c_1})$. Then, $\langle \mathbf{c}_2, \mathbf{s_2} \rangle = 2\langle \text{BitDecomp}(\mathbf{c_1}), \mathbf{e_2} \rangle + \langle \mathbf{c}_1, \mathbf{s}_1 \rangle \bmod q$

2.4 Modulus Switching

Modulus Switching stands for a process, which does not use the secret key $\mathbf{s}$ but a bound on its length, can transform a ciphertext $\mathbf{c}$ modulo q into a different ciphertext modulo p while preserving correctness -namely, $\left[\langle \mathbf{c}', \mathbf{s} \rangle \right]_p = \left[\langle \mathbf{c}, \mathbf{s} \rangle \right]_q \bmod 2$.

For integer vector $\mathbf{x}$ and integers $q > p > m$, we define $\mathbf{x}' = \text{Scale}(\mathbf{x}, q, p, r)$ to be the R-vector closest to $(p/q)\cdot\mathbf{x}$ that satisfies $\mathbf{x}' = \mathbf{x} \bmod r$.

Lemma 2.7: Let q and p be two odd moduli, and let $\mathbf{c}$ be an integer vector. Define $\mathbf{c}'$ to be the integer vector

closest to $(p/q)\mathbf{c}$ such that $\mathbf{c}' = \mathbf{c} \bmod 2$. Then, for any $\mathbf{s}$ with $\left|\left[\langle \mathbf{c}, \mathbf{s}\rangle\right]_q\right| < q/2 - (q/p)\cdot \ell_1(\mathbf{s})$, we have $\left[\langle \mathbf{c}', \mathbf{s}\rangle\right]_p = \left[\langle \mathbf{c}, \mathbf{s}\rangle\right]_q \bmod 2$ and $\left|\left[\langle \mathbf{c}', \mathbf{s}\rangle\right]_p\right| < (q/p)\left|\left[\langle \mathbf{c}, \mathbf{s}\rangle\right]_q\right| + \ell_1(\mathbf{s})$, where $\ell_1(\mathbf{s})$ is the ℓ_1-norm of $\mathbf{s}$.

3 A Fully Homomorphic Encryption Scheme based on bLWE

FHE.Setup$(1^\lambda, 1^L)$: Takes as input the security parameter λ, a number of levels L. Let $\mu=\mu(\lambda, L)=\theta(\log\lambda + \log L)$ be a parameter about moduli. For $j = L$ (input level of circuit) to 0 (output level), run $params_j \leftarrow \text{E.Setup}(1^\lambda, 1^{(j+1)\mu})$ to obtain a ladder of decreasing moduli from $q_L\left((L+1)\mu\text{bits}\right)$ down to $q_0(\mu\text{bits})$. For $j = L-1$ to 0, replace the distribution χ_L with $\chi=\chi_L$. (That is, the noise distribution do not depend on the circuit level.)

FHE.KeyGen($\{\text{params}_j\}$): Set $\mathbf{s}_L \leftarrow \text{E.SecretKeyGen}(params_L)$. Generate matrix $\mathbf{A}'_L \leftarrow R_{q_L}^{N\times n}$ uniformly, a vector $\mathbf{e}_L \leftarrow \chi^N$, Set $\mathbf{b}_L \leftarrow \mathbf{A}'_L\mathbf{s}'_L + 2\mathbf{e}_L$ and $\mathbf{A}_L = (\mathbf{b}_L \mid -\mathbf{A}'_L)$.

For $j = L$ down to 0, do the following:

1. Run $\mathbf{s}_j = \mathbf{s}_L \bmod q_j$ and $\mathbf{A}_j \leftarrow \text{E.PublicKey Gen}(params_j, \mathbf{s}_j)$. Notice that $\mathbf{s}_j$ and $\mathbf{s}_L$ are the same, essentially, since $\mathbf{s}_j$ is the state of $\mathbf{s}_L$ under the different modulus. (Omit this step when $j = L$.)

2. Set $\mathbf{s}'_j \leftarrow \mathbf{s}_j \otimes \mathbf{s}_j \in R_{q_j}^{\binom{n_j+1}{2}}$. That is, $\mathbf{s}'_j$ is a tensor of $\mathbf{s}_j$ with itself whose coefficients are each the product of two coefficients of $\mathbf{s}_j$ in R_{q_j}.

3. Run $\tau_{\mathbf{s}'_{j+1}\to\mathbf{s}_j} \leftarrow \text{SwitchKeyGen}(\mathbf{s}'_{j+1}, \mathbf{s}_j)$. (Omit this step when $j = L$.)

Output: The secret key $sk = \mathbf{s}_L$ and the public key $pk = (A_j, \tau_{\mathbf{s}'_{j+1}\to\mathbf{s}_j})$ (For $j = L$ down to 0.)

FHE.Enc$(params, pk, m)$: Take a message in R_2. Output: E.Enc$(params_L, A_L, m)$.

FHE.Dec$(params, pk, \mathbf{c})$: Suppose the ciphertext is under level of j. Output: E.Dec$(params_j, s_j, \mathbf{c})$. (The ciphertext could be augmented with an index indicating which level it belongs to.)

FHE.Add$(pk, \mathbf{c}_1, \mathbf{c}_2)$: Apply Inputs Normalization: suppose the two ciphertexts are $\mathbf{c}_1$ modulo $q_{\mathbf{c}_1}$ and $\mathbf{c}_2$ modulo $q_{\mathbf{c}_2}$. If $q_{\mathbf{c}_2} = q_{\mathbf{c}_1}$, abandon. If $q_{\mathbf{c}_2} > q_{\mathbf{c}_1}$, set $\mathbf{c}_2 = \text{Scale}(\mathbf{c}_2, q_{\mathbf{c}_2}, q_{\mathbf{c}_1}, 2)$. If $q_{\mathbf{c}_1} > q_{\mathbf{c}_2}$, set $\mathbf{c}_1 = \text{Scale}(\mathbf{c}_1, q_{\mathbf{c}_1}, q_{\mathbf{c}_2}, 2)$ (Notice that it takes $\left|q_{\mathbf{c}_2} - q_{\mathbf{c}_1}\right|$ times of FHE.Refresh layer-by-layer for BGV, thus inefficient). Set $q_j = \max\left\{q_{\mathbf{c}_2}, q_{\mathbf{c}_1}\right\}$.

Output: $\mathbf{c}_4 \leftarrow \mathbf{c}_1 + \mathbf{c}_2 \bmod q_j$

FHE.Mult$(pk, \mathbf{c}_1, \mathbf{c}_2)$: Apply Inputs Normalization. First, multiply: the new ciphertext, under the secret key $\mathbf{s}'_j = \mathbf{s}_j \otimes \mathbf{s}_j$, is the coefficient vector $\mathbf{c}_3$ of the linear equation $L^{long}_{\mathbf{c}_1,\mathbf{c}_2}(\mathbf{x}\otimes\mathbf{x})$. Then, output:

$\mathbf{c}_4 \leftarrow \text{FHE.Refresh}(\mathbf{c}_3, \tau_{\mathbf{s}'_j\to\mathbf{s}_{j-1}}, q_j, q_{j-1})$

FHE.Refresh$(\mathbf{c}, \tau_{\mathbf{s}'_j\to\mathbf{s}_{j-1}}, q_j, q_{j-1})$: Takes a ciphertext encrypted under $\mathbf{s}'_j$, the auxiliary information $\tau_{\mathbf{s}'_j\to\mathbf{s}_{j-1}}$ to facilitate key switching, and the current and next moduli q_j and q_{j-1}. Do the following:

1. Switch Keys: Set $\mathbf{c}_1 \leftarrow \text{SwitchKey}(\tau_{\mathbf{s}'_j\to\mathbf{s}_{j-1}}, \mathbf{c}, q_j)$, a ciphertext under the key $\mathbf{s}_{j-1}$ for modulus q_j.

2. Switch Moduli: Output $\mathbf{c}_2 \leftarrow \text{Scale}(\mathbf{c}_1, q_j, q_{j-1}, 2)$, a ciphertext under the key $\mathbf{s}_{j-1}$ for modulus q_{j-1}.

3.1 Correctness and Performance

There are two distinctions between the FHE scheme of BGV based on LWE with this paper. Firstly, we improve the process of FHE.Add. Secondly, we replaces the sample $(\text{A}, \mathbf{b})$ in the BGV based on LWE with the sample $(\bar{\text{A}}, \mathbf{b})$. Notice that the second part doesn't interfere with the correctness and efficiency of BGV. The first part is easy after improvement. The proof of theorem is similar to the proof of BGV. So we omit the concrete proof here.

Theorem 3.1 For some $\mu=\theta(\log\lambda + \log L)$, FHE is a correct L-leveled FHE scheme – specifically, it correctly evaluates circuits of depth L with Add and Mult gates over R_2. The per-gate computation of Mult gates is $\tilde{O}\left(\lambda^3 \cdot L^5\right)$ and Add gates is $\tilde{O}(\lambda)$.

3.2 Security

In this section, we will focus on the security of our FHE scheme and prove the CPA security of the FHE scheme and the circular security of key switching process in standard model.

Lemma 3.2 For any $n \ge 2, q \ge 1, d \ge 1$, and error distribution χ, there is an efficient (transformation) reduction from $LWE_{n-1,q,\chi}$ to the $bLWE_{n,q,\chi}$ that reduces the advantage by at most 2^{-n}.

Proof. Sample $(\mathbf{a}', b') \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ from the given oracle,

and output $(\mathbf{a}', b) = (\mathbf{a}', b' + (d - \|\mathbf{a}'\|_1)\mathbf{s}[\mathrm{n}])$. Set $\mathbf{a} = (\mathbf{a}' \mid (d - \|\mathbf{a}'\|_1)$ with the vertical bar denoting concatenation.

(1) Given an uniform sample $(\mathbf{a}', b') \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$, the reduction outputs an uniform sample $(\mathbf{a}', b) = (\mathbf{a}', b' + (d - \|\mathbf{a}'\|_1)\mathbf{s}[\mathrm{n}])$, up to statistical distance 2^{-n} (since d is a constant, b' and $\mathbf{s}[n]$ are uniform in $\mathbb{Z}_q$.)

(2) Given a sample $(\mathbf{a}', b') \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ from $A_{q,\mathbf{s},\chi}$, the reduction outputs a sample $(\mathbf{a}', b) = (\mathbf{a}', b' + (d - \|\mathbf{a}'\|_1)\mathbf{s}[\mathrm{n}])$ from $bA_{q,\mathbf{s},\chi}$, where $b = b' + (d - \|\mathbf{a}'\|_1)\mathbf{s}[\mathrm{n}] = <\mathbf{a}, \mathbf{s}> + 2\mathrm{e}$, up to statistical distance 2^{-n}.

Therefore, if the $LWE_{n-1,\mathrm{q},\chi}$ problem is infeasible, then $bLWE_{n,\mathrm{q},\chi}$ problem is infeasible, completing the proof. It means that $bA_{q,\mathbf{s},\chi}$ is indistinguishable from uniform.

A public-key encryption system is circular-secure when it remains secure even encrypting some messages that depend on the secret keys **[6]**.

Theorem 3.3. Let q, n, N, χ be the parameters associated to FHE.SwitchKey. There is an efficient (transformation) reduction from FHE.SwitchKey to the $bLWE_{n,\mathrm{q},\chi}$.

We can get the result by proofing that each row of $\tau_{\mathbf{s}_1 \to \mathbf{s}_2} = \mathbf{B}$ doesn't leak any message of $\boldsymbol{s}$, namely, the lows are indistinguishable from uniform.

Set $\mathrm{g}_i(\boldsymbol{s})$=Powersof2($\boldsymbol{s}$)[$i$] and $\boldsymbol{t} = \boldsymbol{s} + (\mathrm{g}_i(\boldsymbol{s}),\ \mathrm{g}_i(\boldsymbol{s}),..., \mathrm{g}_i(\boldsymbol{s})) \in \mathbb{Z}_q^n$. The following equation is correct, since we have $\boldsymbol{a}^{-1} = 1$ in the process of SwitchKeyGen.

$$
\begin{aligned}
b &= <\boldsymbol{a}, \boldsymbol{s}> + 2e + \mathrm{g}_i(\boldsymbol{s}) \\
&= <\boldsymbol{a}, \boldsymbol{s}> + 2e + \|\boldsymbol{a}\|_1 \cdot \mathrm{g}_i(\boldsymbol{s}) \\
&= <\boldsymbol{a}, \boldsymbol{s}> + 2e + <\boldsymbol{a}, (\mathrm{g}_i(\boldsymbol{s}),\ \mathrm{g}_i(\boldsymbol{s}),...,\ \mathrm{g}_i(\boldsymbol{s}))> \\
&= <\boldsymbol{a}, \boldsymbol{s} + (\mathrm{g}_i(\boldsymbol{s}),\ \mathrm{g}_i(\boldsymbol{s}),...,\ \mathrm{g}_i(\boldsymbol{s}))> + 2e \\
&= <\boldsymbol{a}, \boldsymbol{t}> + 2e
\end{aligned}
$$

Notice that $\mathbf{t}$ is a vector based on $\boldsymbol{s}$ and $\mathbf{s}' \leftarrow \chi^n$, only. We view $(\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1], <\boldsymbol{a}, \boldsymbol{t}> + 2e) \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ as a sample of $bA_{q,\mathbf{t},\chi}$, up to statistical distance 2^{-n}. If there is an adversary can distinguish $bA_{q,\mathbf{t},\chi}$ from uniform. Then there is an adversary can distinguish $A_{q,\mathbf{t},\chi}$ from uniform. Then according to the average-case to worst-case lemma, there is an adversary can distinguish $A_{q,\mathbf{s},\chi}$ from uniform which is infeasible. As a result, $(\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1], <\boldsymbol{a}, \boldsymbol{t}> + 2e) \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ is indistinguishable from uniform.

Sample $(\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1], b) \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ from the $bLWE_{n,\mathrm{q},\chi}$ oracle. Given an uniform sample, the reduction outputs an uniform sample. Given samples from $bA_{q,\mathbf{t},\chi}$, the reduction outputs a sample of the row of switching key, completing the proof. It means that we have constructed the first circularly secure key switching process, what's more, it can replace the key switching process and similar re-linearization process used by the existing FHE schemes.

According to lemma 3.2, $(\mathbf{a}[1],...,\mathbf{a}[\mathrm{n}-1], <\boldsymbol{a}, \boldsymbol{s}> + 2e + \mathrm{Powersof2}(\boldsymbol{s})[i]) \in \mathbb{Z}_q^{n-1} \times \mathbb{Z}_q$ is indistinguishable from uniform.

We let $Adv_{NSH}^{CPA}(C)$ be the success probability of attacker C in our scheme, $Adv_{BGV}(D)$ be the success probability of attacker D in the FHE scheme of BGV based on LWE, $Adv_{LWE_n}(A)$ be the success probability of attacker A in the LWE with n dimensions, $Adv_{bLWE_n}(B)$ be the success probability of attacker B in the bLWE with n dimensions.

There are two distinctions between the FHE scheme of BGV based on LWE with this paper. The first distinction between the BGV based on LWE with this paper is that we improve the process of FHE.Add. The second is that we replaces the sample $(\mathrm{A}, \mathbf{b})$ in the BGV based on LWE with the sample $(\bar{\mathrm{A}}, \mathbf{b})$. We will focus on the second part, since the first part don't lower the secure.

The following inequations are correct, since each row of $(\bar{\mathrm{A}}, \mathbf{b})$ is indistinguishable from the sample of bLWE.

$$
\begin{aligned}
&Adv_{NSH}^{CPA}(C) \le Adv_{BGV}(D) + |\, Adv_{LWE_n}(A) - Adv_{aLWE_n}(B)\,| \\
&\le Adv_{BGV}(D) + |\, Adv_{LWE_n}(A) - Adv_{LWE_{n-1}}(A) + Adv_{LWE_{n-1}}(A) \\
&- Adv_{bLWE_n}(B)\,| \le Adv_{BGV}(D) + |\, Adv_{LWE_n}(A) - Adv_{LWE_{n-1}}(A)\,| \\
&+ |\, Adv_{LWE_{n-1}}(A) - Adv_{bLWE_n}(B)\,|.
\end{aligned}
$$

Notice that $Adv_{BGV}(D)$, $|\, Adv_{LWE_n}(A) - Adv_{LWE_{n-1}}(A)\,|$ and $|\, Adv_{LWE_{N-1}}(A) - Adv_{bLWE_N}(B)\,|$ are negligible. It means that the success probability of attacker C in our scheme is negligible and our scheme is CPA secure.

3.3 Performance

This paper improves the BGV based on LWE. Compared with BGV based on LWE, our scheme has advantages of a smaller secret key size , easier and faster computation in Inputs Normalization and an efficient FHE.Add.

1. Secret key size. There are L secret keys in BGV, since each level needs a secret key $\mathbf{s}_j \in R_{q_j}^{n+1}$ to decryption. Our scheme needs only one secret key $\mathbf{s}_L$.

2. Inputs Normalization. BGV needs to apply FHE.Refresh layer-by-layer to modify the input ciphertexts of different levels. Our scheme needs only one Scale operation.

3. FHE.Add operation. BGV uses FHE.Refresh operation which contains SwitchKey operation and Scale operation in FHE.Add. It can be removed for efficiency.

Table 1 Efficiency Comparison of BV11b [9], BGV [10] and Our Scheme

Schemes	Public Key Size	Secret Key Size	Circular Security	CPA Security
BV11b [9]	$[m\times(n+1)]\mathrm{lb}\,q$	$L+1$	No	Yes
BGV [10]	$[N(n+1)]\cdot(L+1)\cdot\mathrm{lb}\,q$	L	No	Yes
Our Scheme	$[N(n+1)]\cdot(L+1)\cdot\mathrm{lb}\,q$	1	Yes	Yes

As a result, our scheme is more efficient then BGV, DGHV and BV11b based on LWE, since BGV is more efficient than these schemes.

4 Summary and Future Directions

This paper improves the BGV based on LWE. Our scheme has a smaller secret key size, faster computation in Inputs Normalization and an efficient FHE.Add comparing with BGV based on LWE. An important remaining problem is to develop the circular security key switching process to base on RLWE and apply it to other scheme, such as GSW, BV11b. Another interesting problem is to improve the performance of our system.

This work was supported by the National Natural Science Foundation of China (Grant No. 61272492. 61103231. 61103230) .

1 Li J, Song Y Q. DLB: a novel real-time QoS control mechanism for multimedia transmission[J]. International journal of high performance computing and networking, 2009, 6(1): 4-14.

2 Swankoski E J, Vijaykrishnan N, Brooks R, et al. Symmetric encryption in reconfigurable and custom hardware[J]. International Journal of Embedded Systems, 2005, 1(3): 205-217.

3 Pande A, Zambreno J. Reconfigurable hardware implementation of a modified chaotic filter bank scheme[J]. International Journal of Embedded Systems, 2010, 4(3): 248-258.

4 Abdellatif K M, Chotin-Avot R, Mehrez H. Low cost solutions for secure remote reconfiguration of FPGAs[J]. International Journal of Embedded Systems, 2014, 6(2): 257-265.

5 R.L.Rivest, L. Adleman, M.L.Dertouzos. On Data Banks and Privacy Homomorphisms. Foundations of Secure Computation, pp.169-177, 1978.

6 Zhang Wei, Liu Shuguang, Yang Xiaoyuan, Multi-bit homomorphic encryption based on learning with errors over rings[J]. IACR Cryptology ePrint Archive, 2013, pp.138-138.

7 C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169-178, 2009.

8 M. Dijk, C.Gentry, S.Halevi, V.Vaikuntanathan, Fully Homomorphic Encryption over the Integers. Advancesin Cryptology-EUROCRYPT 2010, pp.24-43, 2010. 1, 7, 8.

9 Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. Advances in Cryptology-CRYPTO2011, pp.505-524, 2011. 1, 9, 13

10 Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping[C]//Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ACM, 2012: 309-325.

11 Halevi S, Shoup V. Design and Implementation of a Homomorphic-Encryption Library[EB/OL]. [2012-04-09]. http://eprint.iacr.org/2012/181 .

12 Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[M]//Advances in Cryptology–CRYPTO 2013. Springer Berlin Heidelberg, 2013: 75-92.

13 Z.Brakerski, V.Vaikuntanathany. Efficient Fully Homomorphic Encryption from (Standard) LWE. In: Electronic Colloquium on Computational Complexity ECCC, Vol. 18 (2011) , p. 109-138.

14 Craig Gentry. A fully homomorphic encryption scheme. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

15 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, STOC, pages 84-93. ACM, 2005.

16 Regev O. The learning with errors problem[J]. Invited survey in CCC, 2010.

Table 1 Efficiency Comparison of BV11b [9], BGV [10] and Our Scheme

Schemes	Public Key Size	Secret Key Size	Circular Security	CPA Security
BV11b [9]	$[m\times(n+1)]\,\mathrm{lb}\,q$	$L+1$	No	Yes
BGV [10]	$[N(n+1)]\cdot(L+1)\cdot\mathrm{lb}\,q$	L	No	Yes
Our Scheme	$[N(n+1)]\cdot(L+1)\cdot\mathrm{lb}\,q$	1	Yes	Yes