Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5787930, 7 pages
https://doi.org/10.1155/2018/5787930

Research Article

Privacy Protection of IoT Based on Fully Homomorphic Encryption

Wei-Tao Song, Bin Hu, and Xiu-Feng Zhao

Information Science and Technology Institute, Zhengzhou 450001, China

Correspondence should be addressed to Wei-Tao Song; weitaosong@163.com

Received 4 March 2018; Revised 27 April 2018; Accepted 15 May 2018; Published 20 June 2018

Academic Editor: Ximeng Liu

With the rapid development of Internet of Things (IoT), grave questions of privacy protection are raised. This greatly impacts the large-scale applications of IoT. Fully homomorphic encryption (FHE) can provide privacy protection for IoT. But, its efficiency needs to be greatly improved. Nowadays, Gentry's bootstrapping technique is still the only known method of obtaining a "pure" FHE scheme. And it is also the key for the low efficiency of FHE scheme due to the complexity homomorphic decryption. In this paper, the bootstrapping technique of Halevi and Shoup (EUROCRYPT 15) is improved. Firstly, by introducing a definition of "load capacity", we optimize the parameter range for which their bootstrapping technique works. Next we generalize their ciphertext modulus from closing to a power of two to more general situations. This enables the method to be applied in a larger number of situations. Moreover, this paper also shows how to introduce SIMD homomorphic computation techniques into the new method, to improve the efficiency of recryption.

1. Introduction

Nowadays, the IoT is becoming an attractive system paradigm to drive a substantive leap on goods and services through physical, cyber, and social spaces. It covers from traditional equipment to general household equipment, which bring more efficiency and convenience to the users and change current ways of life greatly [1]. See Figure 1.

However, the application of IoT involves mass private information about users, such as healthcare, location, etc. For the users, they want service providers to process the data accurately and efficiently and extract the contained valuable information with keeping user data unknown by others (including themselves). All these problems are difficult to achieve by traditionally encryption schemes. Homomorphic encryption technology is a good choice to solve all these problems [2, 3].

FHE permits a worker to perform arbitrarily complex programs on encrypted data without knowing the secret key [4]. And FHE has been the focus of extensive study [5–13], since the first candidate scheme was introduced by Gentry [14]. But its efficiency needs to be greatly improved.

Since bootstrapping technology is the essential technology to obtain a "pure" FHE at present. Meanwhile, it is also the main bottleneck in any practical implementation due to the complexity homomorphic decryption. It is very meaningful to improve the efficiency of bootstrapping, which mainly refers to fast low-circuit implementation of decryption function. Without loss of generality, the decryption function for LWE- (Learning with Errors-) based FHE can be computed by evaluating some linear operation between ciphertext and secret key, then reducing the result modulo a big odd modulus q and then reducing the result modulo a small modulus p, to get the plaintext m, namely, $m = [[L_c(s)]_q]_p$. For the decryption function, the modular-reduction operation of $[[z]_q]_p$ ($z \in \mathbb{Z}$) affects the depth of decryption circuit most.

The past few years have seen an intensive study of bootstrapping technique. In the original bootstrapping technique of Gentry [14], he put forward an idea of "squash the decryption circuit" to transform modular-reduction operation into summing operation. This got a moderate polynomial $O(\lambda^4)$ runtime. By proposed an amortized bootstrapping method, Brakerski, Gentry, and Vaikuntanathan (BGV) reduced the

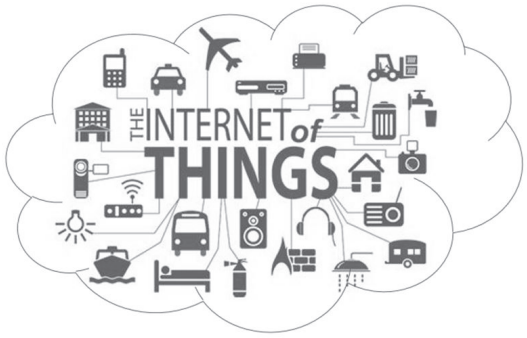

Figure 1: IoT architecture.

runtime to $O(\lambda^2)$ [10]. However, these results applied only to "non-packed" ciphertexts (i.e., ones that encrypt just one bit each).

Gentry, Halevi, and Smart (GHS, PKC2012) reached a major milestone of a bootstrapping algorithm concentrating on the BGV ring-LWE-based scheme (ideal lattice-based FHE) [15]. They proposed a simpler decryption formula. This is done by choosing a prime plaintext modulus p and a ciphertext modulus q close to a power of p. Besides, they utilized packed ciphertexts and Fourier Transform to aid efficiency. To improve the Fourier Transform step of [15], [16] presented a ring/field switching technique. This obtained an asymptotically efficient bootstrapping method for BGV style SHE scheme. Orsini, Pol, and Smart (PKC15) proposed a bootstrapping BGV ciphertexts with a wider choice of p and q, but their decryption formula was not simple as GHS's work. Halevi and Shoup (HS, EUROCRYPT 15) generalized the plaintext modulus p in [17] to more general situations and got a better efficiency by improving the bit-extraction way. This is asymptotically optimal space and time so far.

In another line of work, [18–22] present a bootstrapping technique for the GSW-FHE [13] scheme. They get significant progress in improving the bootstrapping technique on standard lattice-based FHE. And their progress mainly relies on the characteristic that noise in ciphertexts of GSW-FHE grows asymmetrically. Since compared with standard LWE-based FHE schemes, ring-LWE-based FHE schemes always have more efficient homomorphic operations. And among all the ring-LWE-based FHE schemes, BGV ring-LWE-based FHE scheme is optimal (note that GSW-FHE scheme is better than BGV-FHE scheme only in standard LWE-based FHE schemes). Note that, in this paper, the aim is to provide practical FHE scheme for the privacy protection of IoT. Thus, it concentrates on improving the bootstrapping technique of BGV ring-LWE-based FHE schemes in this paper.

The starting point of this paper is the HS's work [17], where decryption procedure consists of a linear algebra step $< c, s >$ and a modular reduction step $[[z]_q]_p$. When $|z| \leq q^2/4 - q$ and $|[z]_q| \leq q/4$, modular reduction step $[[z]_q]_p$ can be converted to simple bit operations. This greatly reduces the circuit depth of modular reduction. When homomorphically performed above simple decryption formula, the deepest part is homomorphic bit-extraction procedure, and its complexity (both time and depth) increases with the most-significant extracted bit. In [17], by adding to ciphertext multiples of q and also multiples of p, they proposed a lower-degree homomorphic bit-extraction procedure. And the bigger the parameter range of z for the simple formula of modular reduction, the better the performance for the improved homomorphic bit-extraction procedure. See [17] for further details.

Contributions. In this paper, we optimize the parameters of bootstrapping algorithm proposed in EUROCRYPT 2015 by Halevi and Shoup. Firstly, by introducing a definition of load capacity, we optimize the parameters range for which their bootstrapping technique works for the first time. Next we generalize their ciphertext modulus q to more general situations. This makes our method applicable to more cases. Moreover, we also show how to introduce SIMD technique into our new method, to improve the efficiency of bootstrapping technique.

Organization. Section 2 presents the notations and some background on the BGV cryptosystem. Section 3 optimizes the parameter range for which bootstrapping technique of Halevi and Shoup works. Next, the ciphertext modulus is generalized from closing to a power of two to more general situations in Section 4. Moreover, it also shows how to introduce SIMD homomorphic computation techniques into the new method to get an efficient bootstrapping method. And in Section 5, an implementation is made of BGV ring-LWE-based scheme based on our efficient bootstrapping method. Finally, Section 6 concludes.

2. Preliminaries

Basic Notations. Set $\mathbb{Z}_q \in (-q/2, q/2] \bigcap \mathbb{Z}$, and the notation $[a]_q$ is referred to as a mod q, with coefficients being reduced into the range $(-q/2, q/2]$. For an integer z (positive or negative), we consider the base-p representation of z and denote its digits by $z\langle 0\rangle_p, z\langle 1\rangle_p, \cdots$.

2.1. Homomorphic Encryption Schemes. Let $\mathbb{M}$ be the message space and $\mathbb{C}$ be the ciphertext space. A homomorphic encryption scheme $HE = \{KeyGen, Enc, Dec, Eval\}$ is as follows:

(i) $KeyGen(1^\lambda)$: output public key pk, secret key sk, and evaluation key evk.

(ii) $Enc_{pk}(\mu)$: output ciphertext $c \in \mathbb{C}$ encrypted by plaintext $\mu \in \mathbb{M}$ with public key pk.

(iii) $Dec_{sk}(c)$: recover the message encrypted in the ciphertext c by secret key sk.

(iv) $Eval_{evk}(f, c_1, \ldots, c_l)$: output ciphertext $c_f \in \mathbb{C}$ which is obtained by applying evaluation key evk and the function $f : \mathbb{M}^l \to \mathbb{M}$ to $c_1, \ldots, c_l$.

Suppose that (sk_1, pk_1) and (sk_2, pk_2) are two key-pairs of scheme HE. Let c be a ciphertext of plaintext μ under pk. Let $\overline{sk_{1i}}$ be a ciphertext of the i-th bit of the first secret key sk_1

under the second public key pk_2. D is a decryption circuit. See Algorithm 1 for the "Bootstrapping" algorithm.

It can be found that $HE.Dec(sk_2, c') = HE.Dec(sk_1, c) = \mu$ only when scheme HE can compactly evaluate its decryption circuit. However, most of the existing schemes do not satisfy this condition naturally. It needs some extra operations, such as "squashing the decryption circuit", which cause the low efficiency of FHE. Thus, it is very meaningful for lower-depth circuit implementation of decryption function.

3. Analysis of HS Recryption Procedure

We start by introducing the HS recryption procedure [17] on that how to homomorphically compute the modular-reduction operation in a lower-depth circuit. The specifics are in Lemma 1.

Lemma 1 (see [17]). *Let $p > 1$, $r \geq 1$, $e \geq r+2$ and $q = p^e + 1$ be integers, and also let z be an integer such that $|z| \leq q^2/4 - q$ and $|[z]_q| \leq q/4$.*

(i) *If p is odd then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\bmod p^r)$.*

(ii) *If $p = 2$ then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\bmod 2^r)$.*

Lemma 1 transforms complex modular operations into simple bit operation, to get a lower-depth circuit of decryption function. But it is still not easy to execute a homomorphic bit-extraction operation. Next, [17] proposed a fast bit-extraction procedure. As stated in the former introduction, the performance of fast bit-extraction procedure is dependent on the parameter range of z in Lemma 1. That is, the bigger parameter range of z, the better performance of fast bit-extraction procedure. Thus, next we analyse whether the parameter range of z in Lemma 1 is optimal. In order to do so, we introduce a new concept called "load capacity".

Definition 2 (load capacity). Let $q \in \mathbb{Z}^+$, $z \in \mathbb{Z}$. Suppose the formula of modular reduction converted to simple bit operations works when $-q/2 < a \leq [z]_q \leq b \leq q/2$, and $c \leq z < d$. Then the load capacity is defined by the product of two span lengths of z and $[z]_q$, namely, $(b-a) \times (d-c)$.

Next Theorem 3 presents the general relationship between the value z and $[z]_q$ for the formula of modular reduction converted to simple bit operations.

Theorem 3. *Let $p > 1$, $r \geq 1$, $e \geq r+2$ and $q = p^e + 1$ be integers, and also let z be an integer such that $[z]_q \in [a, b]$, and let $a \cdot (1-q) \leq z < (q-1) \times (q-b)$. Then*

(i) *if p is odd then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\bmod p^r)$;*

(ii) *if $p = 2$ then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\bmod 2^r)$.*

Input: $pk_2, D, \langle \overline{sk_{1i}} \rangle, c$
Output: c'
Step 1. $\overline{c_i} \xleftarrow{R} HE.Enc(pk_2, c_i)$ where c_i is the i-bit of c
Step 2. $c' \leftarrow HE.Eval(pk_2, D, \langle \langle \overline{sk_{1i}} \rangle, \overline{c_i} \rangle)$

Algorithm 1: "Bootstrapping" algorithm.

Proof. It starts with the odd-p case. Let $z_0 = [z]_q \in [a, b]$ and $z = z_0 + k \cdot q$ with $k \in \mathbb{Z}$. Then

$$z = z_0 + k \cdot (p^e + 1) = (z_0 + k) + k \cdot p^e. \tag{1}$$

Since $e \geq r+2$, we can get that

$$z = z_0 + k \,(\bmod p^r). \tag{2}$$

Besides, since

$$\begin{gathered} a \cdot (1-q) \leq z < (q-1) \times (q-b), \\ z_0 + k = z_0 + \frac{z - z_0}{q} = \frac{z + z_0 (q-1)}{q}, \end{gathered} \tag{3}$$

then

$$\begin{gathered} z_0 + k \in \left[\frac{a \cdot (1-q) + z_0 (q-1)}{q}, \right. \\ \left. \frac{(q-1) \times (q-b) + z_0 (q-1)}{q} \right). \end{gathered} \tag{4}$$

And since $-q/2 \leq a \leq z_0 \leq b \leq q/2$, then

$$\begin{gathered} z_0 + k \in \left[\frac{a \cdot (1-q) + a (q-1)}{q}, \right. \\ \left. \frac{(q-1) \times (q-b) + b (q-1)}{q} \right) = [0, p^e). \end{gathered} \tag{5}$$

Thus, combined with formula (2), we can get that

$$k \langle r-1, \cdots, 0 \rangle = z \langle e+r-1, \cdots, e \rangle, \tag{6}$$

where $k < r-1, \cdots, 0 >$ and $z < e+r-1, \cdots, e >$ are mod-p representation. Then it follows that

$$\begin{aligned} z_0 \langle r-1, \cdots, 0 \rangle &= z \langle r-1, \cdots, 0 \rangle - k \langle r-1, \cdots, 0 \rangle \\ &= z \langle r-1, \cdots, 0 \rangle \\ &\quad - z \langle e+r-1, \cdots, e \rangle \,(\bmod p^r). \end{aligned} \tag{7}$$

The proof for the $p = 2$ case is similar. The details can be referred to in the proof of [17]. It is omitted here. □

Next we discuss how to choose the value of a, b in order to obtain the maximum "load capacity". Load capacity is denoted by t, then

$$\begin{aligned} t &= ((q-1) \cdot (q-b) - a (1-q)) \cdot (b-a) \\ &= (1-q) \cdot (b-a)^2 + q (q-1) \cdot (b-a). \end{aligned} \tag{8}$$

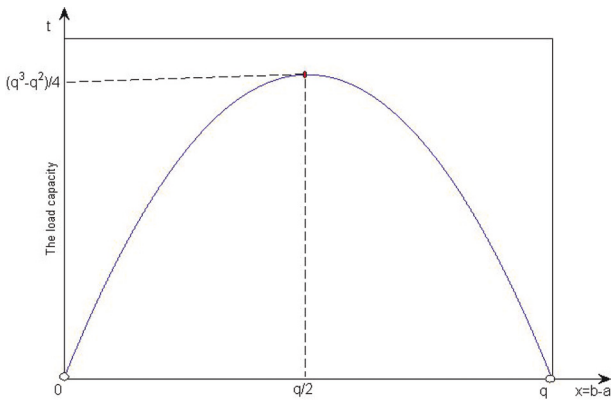

Figure 2: The load capacity on the span length of a and b.

Let $x = b - a$, then $t = (1-q)\cdot x^2 + q(q-1)\cdot x$. The concrete relations are as shown in Figure 2

It can be easily seen from Figure 2 that the load capacity takes the maximum value when $b - a = q/2$. That is, the load capacity for HS work is only related to the span length, not to the value of a and b. Then Corollary 4 presents the optimal choice of $[z]_q$ and z for the formula of modular reduction converted to simple bit operations.

Corollary 4. *Let $p > 1$, $r \geq 1$, $e \geq r+2$, $q = p^e + 1$ and $a \in (-q/2, 0]$ be integers, and also let z be an integer such that $z \in \mathbb{Z}$, $[z]_q \in [a, a+q/2]$, and $a\cdot(1-q) \leq z < (q-1)\times(q/2-a)$.*

(i) *If p is odd then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > (\bmod p^r)$.*

(ii) *If $p = 2$ then $[z]_q = z < r-1, \cdots, 0 > -z < e+r-1, \cdots, e > -z < e-1 > (\bmod 2^r)$.*

The conclusion is obvious; the proof is omitted here.

Note that, when $a = -q/4$, namely, $|[z]_q| \leq q/4$, it is the same as HS's work. But, the load capacity of this paper is bigger than that of HS's work, since z of ours has a bigger span length, namely, $|z| \leq q^2/4 - q/4$. The details are present in Table 1.

As seen from Table 1, compared to HS' work, it can be seen that our scheme has a better load capacity. Note that, while on the surface, it appears to obtain a tiny improvement in a nondominant term, i.e., where the load capacity of the choice in HS is $q^3/4 - q^2$, this is improved to $q^3/4 - q^2/4$, it is actually a meaningful job when you carefully analyse the principle of the trick of the fast bit-extraction procedure in [17]. That is, add to the coefficients of ct multiples of q and p^r, making them divisible by $p^{e'}$ for some $r \leq e' < e$ without increasing them too much and also without increasing the noise too much. This means that bit-extraction can be implemented using only polynomials of degree at most $e - e'$, smaller than e. Since the load capacity of this paper is $3q^2\backslash 4$ bigger than that of HS's work, it means our work allows adding more multiples of q and p^r to the coefficients of ct. That is, bit-extraction can be implemented using polynomials of lower degree to get a faster implementation. Besides, our variant of HS is more flexible and general on parameters.

4. Generalize Modulus to More General Situations

In this section, it extends HS recryption procedure to have a wider choice of ciphertext modulus. The specifics are in Theorem 5.

Theorem 5. *Let $p > 1$, $r \geq 1$, $e \geq r+2$, and $q = u\cdot p^e + v$ with $u, v \in \mathbb{Z}$ and $u \in [1, p^r - 1]$, $v \in [1, p^e - 1]$, $p \nmid u, v$, also let z be an integer such that $[z]_q \in [a, b]$, and*

$$\frac{a\cdot(v-q)}{v} \leq z < \frac{b\cdot(v-q) + p^e q}{v}. \tag{9}$$

Then,

(i) *if p is odd then $[z]_q = z < r-1, \cdots, 0 > -(((v < r-1, \cdots, 0 >)\times(z < e+r-1, \cdots, e >))/u < r-1, \cdots, 0 >)(\bmod p^r)$,*

(ii) *if $p = 2$ then $[z]_q = z < r-1, \cdots, 0 > -((v < r-1, \cdots, 0 >)\times(z < e+r-1, \cdots, e >))/u < r-1, \cdots, 0 > -z < e-1 > (\bmod p^r)$,*

where "$\times$" refers to scalar multiplication.

Proof. We begin with the odd-p case. Let $z_0 = [z]_q \in [a, b]$ and $z = z_0 + k\cdot q$ with $k \in \mathbb{Z}$. Then

$$z = z_0 + k\cdot(u\cdot p^e + v) = (z_0 + kv) + ku\cdot p^e. \tag{10}$$

Since $e \geq r+2$, we can get that

$$z = z_0 + kv\,(\bmod p^r). \tag{11}$$

Besides, since

$$\begin{gathered}\frac{a\cdot(v-q)}{v} \leq z < \frac{b\cdot(v-q) + p^e q}{v},\\ z_0 + kv = z_0 + \frac{z - z_0}{q}v = \frac{zv + z_0(q-v)}{q},\end{gathered} \tag{12}$$

then

$$\begin{gathered}z_0 + kv \in \left[\frac{((a\cdot(v-q))/v)\,v + z_0(q-v)}{q},\right.\\ \left.\frac{((b\cdot(v-q) + p^e q)/v)\,v + z_0(q-v)}{q}\right).\end{gathered} \tag{13}$$

And since $-q/2 \leq a \leq z_0 \leq b \leq q/2$, then

$$\begin{gathered}z_0 + kv \in \left[\frac{a\cdot(v-q) + a(q-v)}{q},\right.\\ \left.\frac{b\cdot(v-q) + p^e q + b(q-v)}{q}\right) = [0, p^e).\end{gathered} \tag{14}$$

Table 1: The parameter of low-circuit implementation of modular reduction on HS and our work.

Scheme	z	$[z]_q$	load capacity
HS' work [17]	$[q-q^2\backslash 4, q^2\backslash 4-q]$	$[-q\backslash 4, q\backslash 4]$	$q^3/4-q^2$
Our work ($a \in (-q/2, 0]$)	$[a\cdot(1-q), (q-1)\times(q/2-a))$	$[a, a+q/2]$	$q^3/4-q^2/4$

Input: Modulus $q = u\cdot p^e + v$, and a ciphertext c encrypting a constant $b \in (\mathbb{Z}/p^{e+1}\mathbb{Z})$ relative to secret key s and modulus p^{e+1}
Output: A ciphertext c' encrypting the bit operation relative to secret key s and modulus p^e
1. Set $c_0 \leftarrow c$ //c encrypt z w.r.t. s
2. For $i = 1$ to r
3. Set $acc \leftarrow c$ // acc is an accumulator
4. For $j = 0$ to $i = 1$ //Compute $z - \sum_j p^j w_j^{i-1}$
5. Set $tmp \leftarrow HomExp(c_j, p^{i-j})$ //Homomorphic exponentiation to the power p^{i-j}
6. Set $acc \leftarrow acc - p^j \cdot tmp \bmod p^{r+1}$
7. Set $c_i \leftarrow acc\cdot((q_0+1)/p)^i \bmod p^{r+1}$ // encrypts $z < i >_p$
8. Output $c_0 - (v\langle 0\rangle_p / u\langle 0\rangle_p)c_r \bmod p^{r+1}$

Algorithm 2: Bit-extraction (c, r, e, p).

Thus, combined with formula (11), we can get that

$$\begin{aligned} z\langle r-1,\cdots,0\rangle &= z_0\langle r-1,\cdots,0\rangle \\ &\quad + (k\langle r-1,\cdots,0\rangle) \\ &\quad \times (v\langle r-1,\cdots,0\rangle)\,(\bmod p^r) \\ z\langle e+r-1,\cdots,e\rangle &= (k\langle r-1,\cdots,0\rangle) \\ &\quad \times (u\langle r-1,\cdots,0\rangle)\,(\bmod p^r) \\ p &\nmid u, v \end{aligned} \tag{15}$$

Thus,

$$\begin{aligned} [z]_q &= z\langle r-1,\cdots,0\rangle \\ &\quad - \frac{(v\langle r-1,\cdots,0\rangle)\times(z\langle e+r-1,\cdots,e\rangle)}{u\langle r-1,\cdots,0\rangle}\,(\bmod p^r). \end{aligned} \tag{16}$$

The proof for the $p = 2$ case is similar. We omit it here. □

Next we discuss how to choose the value of a, b in order to obtain the maximum "load capacity". Load capacity is denoted by t, then

$$\begin{aligned} t &= \left(\frac{b\cdot(v-q)+p^e q}{v} - \frac{a\cdot(v-q)}{v}\right)\cdot(b-a) \\ &= \frac{(v-q)}{v}\cdot(b-a)^2 + \frac{(q-v)\,q}{uv}\cdot(b-a). \end{aligned} \tag{17}$$

Let $x = b - a$, then

$$t = \frac{(v-q)}{v}\cdot x^2 + \frac{(q-v)\,q}{uv}\cdot x. \tag{18}$$

It is easy to get that t takes the maximum value when

$$x = \frac{(q-v)\,q/uv}{-2\,((v-q)/v)} = \frac{q}{2u}. \tag{19}$$

That is, the load capacity is also only related to the span length, not to the value of a and b.

Then Corollary 6 presents the optimal choice of $[z]_q$ and z for the formula of modular reduction converted to simple bit operations.

Corollary 6. *Let $p > 1$, $r \geq 1$, $e \geq r+2$ and $q = u\cdot p^e + v$ with $u, v \in \mathbb{Z}$ and $u \in [1, p^r - 1]$, $v \in [1, p^e - 1]$, $p \nmid u, v$, also let z be an integer such that $[z]_q \in [a, a+q/2u]$, and*

$$\frac{a\cdot(v-q)}{v} \leq z < \frac{q^2-(v+2au)\,q+2auv}{2uv}. \tag{20}$$

Then

(i) *if p is odd then $[z]_q = z < r-1,\cdots,0 > -(((v < r-1,\cdots,0 >)\times(z < e+r-1,\cdots,e >))/u < r-1,\cdots,0 >)(\bmod p^r)$;*

(ii) *if $p = 2$ then $[z]_q = z < r-1,\cdots,0 > -((v < r-1,\cdots,0 >)\times(z < e+r-1,\cdots,e >))/u < r-1,\cdots,0 > -z < e-1 > (\bmod p^r)$.*

The conclusion is obvious; the proof is omitted here.

To get a homomorphic implementation of the simple decryption formula from above, firstly a homomorphic bit-extraction procedure (Algorithm 2) is presented, which is slightly varied from the bit-extraction procedure of [17].

$HomExp(c, n)$ uses native homomorphic multiplication to multiply *operatornamec* by itself n times. To aid exposition, this code assumes that the modulus and secret key remain fixed; otherwise modulus-switching and key-switching should be added (and the level should be increased correspondingly to some $i > 0$).

Then Algorithm 3 shows how to combine our optimized "bootstrapping" techniques with the SIMD homomorphic computation techniques of Smart-Vercauteren [23], to get a bootstrapping method that works in time quasilinear in the security parameter.

Step 1. The user first post-processes the q_L-secret-key by encrypting s as a q_0-ciphertext $c' = (c_0', c_1')$ with respect to the q_0-secret-key $s' = (1, s)$, namely the user has $[< c', s' > \bmod \Phi_m]_{q_0} = [c_0', c_1' \cdot s \bmod \Phi_m]_{q_0} = p^{r+1} \cdot k + s$ where $k \in \mathbb{Z}[X]/\Phi_m(X)$ with small coefficients.

Step 2. The server computes z homomorphically. Specifically, the server compute the mod-p^{r+1} inner product homomorphically by setting

$$\tilde{z} = \left([c_0 + c_1 \cdot c_0' \bmod \Phi_m]_{q_0}, [c_1 \cdot c_1' \bmod \Phi_m]_{q_0}\right).$$

Step 3. Apply a homomorphic inverse-DFT transformation to convert to CRT-based "packed" ciphertexts that hold the coefficients of z in their plaintext slots.

Step 4. Apply the bit extraction procedure to all these slots in parallel. The result is encryption of polynomials that have the coefficients of z in their plaintext slots.

Step 5. Apply a homomorphic DFT transformation to get back a ciphertext that encrypts the polynomial z itself.

Algorithm 3: Batched bootstrapping implementation of our scheme.

Table 2: Experimental results for our batched bootstrapping and HS.

cyclotomic ring m	plaintext space	number of slots	security level	total recrypt (sec)		space usage (GB)	
				Our work	HS' work	Our work	HS' work
$21845 = 257 \cdot 5 \cdot 17$	2^{16}	1024	76	97	172	2.3	3.0
$18631 = 601 \cdot 31$	2^{25}	720	110	168	235	2.6	3.2
$45551 = 41 \cdot 11 \cdot 101$	17^{40}	1000	106	1475	2037	11.2	13.8
$51319 = 19 \cdot 73 \cdot 37$	127^{36}	1296	161	984	1461	31.7	36.4

5. Implementation and Performance

In this section, an implementation of BGV ring-LWE-based scheme is made, since it offers nearly the most efficient homomorphic operations. This scheme is defined over a ring $R \stackrel{def}{=} \mathbb{Z}[X]/(\Phi_m(X))$, where $\Phi_m(X)$ is the m th cyclotomic polynomial. Let p be a prime or a prime power, and $\mathbb{A}_p := \mathbb{Z}_p[X]/\Phi_m(X)$. Specifically, assume $\Phi_m(X) \equiv F_1(X) \cdots F_\ell(X) (\bmod p)$, where each F_i has the same degree d, which is equal to the order of p modulo m. Then, by the Chinese Remainder Theorem, it has the isomorphism $R_p \cong \bigoplus_{i=1}^{k} (\mathbb{Z}[X]/(p, F_i(X))$. Besides, suppose $\mathrm{sk} = (1, s)$ is the q_L-secret-key, where $s \in \mathbb{Z}[X]/\Phi_m(X)$ is an integer polynomial with small coefficients. $\mathrm{sk}' = (1, s')$ is the q_0-secret-key. $c = (c_0, c_1)$ is the q_L-ciphertext.

First, several groups (m, p, r) are chosen which satisfy $\Phi_m(X) \equiv F_1(X) \cdots F_\ell(X) (\bmod p)$. For each triple (m, p, q), a test is run separately based on our work and HS' work. These tests were run on a four-year-old IBM System x3850 server, with two 64-bit 4-core Intel Xeon E5450 processors, and 35MB L2 cache and 32GB of RAM at 3.0 GHz. And the implementation was mainly based on Shoup's NTL library [23] version 9.10.0 and GNU's GMP library [24]. The former is used for high-level numeric algorithms, and the latter is used for the underlying integer arithmetic operations. Besides, the code was compiled using the gcc compiler (version 4.9.1). Table 2 summarizes the results from our experiments based on our work and HS'.

The first column gives cyclotomic ring m and its factorization into prime powers. The second column gives the plaintext space, i.e., the field/ring that is embedded in each slot. The third column gives the number of slots packed into a single ciphertext. The fourth column gives the effective security level, computed using the formula that is used in HElib taken from [15, Eqn. (8)]. The total recrypt gives the total time for a single recryption, while the previous two rows give a breakdown of that time (note that the time for the linear transforms includes some trivial preprocessing time, as well as the less trivial unpacking/repacking time). The last two rows give the memory used (in gigabytes).

As seen from Table 2, compared to HS' work, it can be easy seen that the variant of HS has advantages both in efficiency and in storage space. Besides the variant of HS is more flexible and general on parameters. This enables our method to be applied in a larger number of situations.

6. Conclusions

Up to now, Gentry's bootstrapping technique is still the only known method of obtaining a "pure" FHE scheme. Meanwhile it is also the key for the low efficiency of FHE scheme. It is very meaningful to improve the efficiency of bootstrapping, which mainly refers to lower-depth circuit implementation of decryption function. In this paper, it improves the "load capacity" of HS's work with a better efficiency for bootstrapping and to generalize q to more general situations in a similar simple way. This enables our method to be applied in a larger number of situations, such as privacy protection of IoT.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was sponsored in part by the National Natural Science Foundation of China (Grants nos. 61272041, 61202491, 61272488, and 61601515) and was also supported by the Foundation of Science and Technology on Information Assurance Laboratory (no. KJ-15-006).

References

[1] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[2] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[3] W.-T. Song, B. Hu, and X.-F. Zhao, "Optimizing LWE-based FHE for better security and privacy protection of smart city," *Journal of Information Science and Engineering*, vol. 33, no. 4, pp. 939–952, 2017.

[4] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and pri-vacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[5] M. van Dijk, C. Gentry, S. Halevi et al., "Fully homomorphic encryption over the integers," in *Proceedings of the 29th International Conference on The-ory and Application of Cryptographic Techniques*, pp. 24–43, Springer, Berlin, Germany, 2010.

[6] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proceedings of the 13th International Con-ference on Pronctice and Theory in Public Key Cryptography*, pp. 420–443, Springer, Berlin: Germany, 2010.

[7] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Scale-invariant fully homomorphic encryption over the integers," in *Proceedings of the 17th Interna-tional Conference on Practice and Theory in Public Key Cryptography*, vol. 8383 of *Lecture Notes in Computer Science*, pp. 311–328, Springer, Berlin, Germany, 2014.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic en-cryption from (standard) LWE," in *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pp. 97–106, IEEE Computer Society, Washington, Wash, USA, 2011.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proceedings of the 31st Annual Conference on Advances in Cryptology*, pp. 505–524, Springer, Berlin, Germany, 2011.

[10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully ho-momorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, Optimizing GSW-FHE and Private Information Retrieval 21, pp. 309–325, ACM Press, New York, NY, USA.

[11] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Proceedings of the 32nd Cryptology Conference*, pp. 868–886, Springer, Berlin, Germany, 2012.

[12] C. Gentry, S. Halevi, C. Peikert et al., "Ring switching in BGV-style homomorphic encryption," in *Proceedings of the 8th International Security and Cryptography for Networks*, pp. 19–37, Springer, Berlin, Germany, 2012.

[13] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attrib-ute-based," in *Proceedings of the 33rd Annual Cryptology Conference*, pp. 75–92, Springer, Berlin, Germany, 2013.

[14] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 169–178, ACM Press, New York, NY, USA, 2009.

[15] C. Gentry, S. Halevi, and N. P. Smart, "Better bootstrapping in fully ho-momorphic encryption," in *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography*, pp. 1–16, Springer, Berlin, Germany, 2012.

[16] J. Alperin-Sheriff and C. Peikert, "Practical bootstrapping in quasilinear time," in *Proceedings of the 33rd Annual Cryptology Conference*, pp. 1–20, 2013.

[17] S. Halevi and V. Shoup, "Bootstrapping for helib," in *Eurocrypt*, pp. 641–670, 2015.

[18] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *ITCS*, pp. 1–8, 2014.

[19] J. Alperin-Sheriff and C. Peikert, *Faster Bootstrapping with Polyno-mial Error*, vol. 8616 of *Lecture Notes in Computer Science*, 2014.

[20] R. Hiromasa, M. Abe, and T. Okamoto, "Packing Messages and Optimiz-ing Boot-strapping in GSW-FHE," in *Public-Key Cryptography-PKC*, pp. 699–715, Springer, Heidelberg, Berlin, Germany, 2015.

[21] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *Eurocrypt*, pp. 617–640, 2015.

[22] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 seconds.

[23] V. Shoup, *NTL: A Library for doing Number Theory*, 2016, http://shoup.net/ntl/.

[24] "The GNU Multiple Precision Arithmetic Library," 2016, http://gmplib.org/.