Faster Bootstrapping of FHE over the integers with large prime message space

Zhizhu Lian[1], Yupu Hu[1], Hu Chen[1], and Baocang Wang[1]

ISN Laboratory, Xidian University, 710071 Xi'an, China
lzz600@126.com yphu@mail.xidian.edu.cn chenhuchh@163.com
bcwang79@aliyun.com

Abstract. Bootstrapping of FHE over the integer with large message is a open problem, which is to evaluate double modulo (c mod p) mod Q arithmetic homomorphically for large Q. In this paper, we express this double modulo reduction circuit as a arithmetic circuit of degree at most $\theta^2 \log^2 \theta/2$, with $O(\theta \log^2 \theta)$ multiplication gates, where $\theta = \frac{\lambda}{\log \lambda}$ and λ is the security parameter. The complexity of decryption circuit is independent of the message space size Q with a constraint $Q > \theta \log^2 \theta/2$.

Keywords: Fully homomorphic encryption, Bootstrapping, Restricted depth-3 circuit.

1 Introduction

Fully homomorphic encryption (FHE) enables computation of any circuits on the encryption data, which was first introduced by Rivest, Adleman and Dertouzos in 1978 [1]. Until 2009, Gentry presented a fully homomorphic encryption scheme based on ideal lattice [2]. Bootstrapping regard as Gentry's breakthrough is a homomorphic evaluation of decryption which transforms the homomorphic encryption with limited homomorphic capability into the FHE scheme that can evaluate any circuits homomorphically.

The first FHE over the integers was proposed by van Dijk et al. [3] at Eurocrypt 2010, called DGHV. The security of DGHV relies on the hardness of the Approximate Greatest Common Divisor problem (AGCD) and the Sparse Subset Sum problem (SSSP). Several works have dramatically improved the efficiency and the hardness assumption needed to implement it, including [4–9]. Some of schemes above are leveled FHE scheme, but there essentially follow Gentry's blueprint.

In DGHV scheme, the message space is binary space. It is easy to extend DGHV scheme with bigger message space such sa the message $m \in \mathbb{Z}_{\mathbb{Q}}$, namely the ciphertext $c = pq + Qe + m$ or $c = p^2 q + \lfloor \frac{p}{Q} \rceil (m + Qr)$ according to the schemes [3, 7, 9, 11], where integer p is the secret key, and q, r, e are uniform randomly chosen integers from some prescribed intervals. Here the bootstrapping procedure is homomorphically evaluate the decryption

$$m = c \mod p \mod Q = c - p\lfloor c/p \rceil \mod Q = (c - p(\lfloor \sum_{i=1}^{\Theta} s_i \mathbf{z}_i \rceil) \mod Q$$

where the complicated division by p is replace by using the hardness assumption of the SSSP: $1/p \approx \sum_{i=1}^{\Theta} s_i y_i \mod Q$ for secret key $\mathbf{s} = (s_1, \ldots, s_\Theta)_2 \in \{0,1\}^\Theta$ is Θ length vector with hamming weight $\theta = \lambda$, and $\mathbf{z}_i = (z_{i,0}.z_{i,-1}, \ldots, z_{i,-L})_Q < Q$ is a real number with $L = \lceil log_Q \lambda \rceil + 2$ bits of precision after the Q-ary point, satisfying $\sum_{i=1}^{\Theta} s_i \mathbf{z}_i \approx c/p$.

At Eurocrypt 2015, Nuida and Kurosawa [8] proposed a polynomial of multiplicative degree of Q yielding the carry a in the procedure $x + y = aQ + b$ for any $x, y \in \mathbb{Z}_Q$, called Q-ary half adder. Then they expressed the decryption as s mod-Q arithmetic circuit of multiplicative degree $Q^4\lambda$.

In 2017, Cheon et al. [12] presented a faster bootstrapping method by using a homomorphic encryption scheme with various message spaces and homomorphic digit extraction technique. The degree of the decryption is $O(\lambda^{1+\varepsilon})$, and the number of homomorphic multiplications is $O(\log^2 \lambda)$, where ε is some small constant (be affected by the modulus Q).

However, Both of the above bootstrapping procedure introduced in [8, 12] only support the message space Q size of a constant number. For $Q > 8\Theta^2$, Cheon and Kim [13] expressed the decryption circuit as an $\mathcal{L}$-restricted depth-3 $(\sum \prod \sum)$ circuit by the technique in [14]. The $\mathcal{L}$-degree is at most $8\Theta^2$ and the number of product gates is at most $8\Theta^2 + \Theta + 1$. As we know, Θ is $\omega(\lambda^6 \log \lambda)$ in [3] and is reduced to $\tilde{O}(\lambda^3)$ in [4].

In 2018, Lian et al. [15] presented a bootstrapping algorithm of FHE over the integers with a large message space. In bootstrapping procedure, they using ciphertext of homomorphic encryption scheme with message space $\mathbb{Z}_{\mathbb{Q}}$ which is encryption for bit message. And they apply mod-Q arithmetic circuit to simulate the bit operations in its decryptionThey use Lagrange interpolating polynomial to reduce the number of simulating the bit XOR operation (it take mod-Q multiplier gates, which It will cause the multiplicative degree of decryption to grow rapidly). The decryption circuit is expressed as a polynomial of multiplicative degree $108\theta \log^3 \theta$. It is clearly that the complexity of the decryption is independent of the message space size Q if $Q > \theta$.

One problem in [15] is that the constant factor of the multiplicative degree of decryption 108 is bigger than the parameter θ. the parameter θ is the hamming weight of the secret key $\boldsymbol{sk}$. It requires that $\binom{\Theta}{\theta/2} \geq 2^\lambda$ to avoid an attack on the SSSP [16]. In [3, 15], θ is set to be the security parameter λ, namely $\theta = \lambda$. While in [4], θ can be chose smaller as long as the hardness assumption of SSSP hold, namely $\theta = \frac{\lambda}{\log \lambda}$. If we set the security parameter $\lambda = 72$, then $\theta = \frac{\lambda}{\log \lambda} = 15$. It is easy to see that the constant factor in the multiplicative degree of decryption circuit is much bigger than θ. So, it requires to seek a method to reduce the measure of he constant factor.

In this paper, we modify the bootstrapping procedure presented in [15]. After using use Lagrange interpolating polynomial to obtain the hamming weight of some vectors in bits, we convert the result into a depth-3 circuit, and then, we apply the technique of [14] replace the 3-for-2 trick over $\mathbb{Z}_{\mathbb{Q}}$ in [15]. Finally, we express the decryption as a polynomial of degree $\theta^2 \log^2 \theta/2$. Compared to

Lian et al.'s bootstrapping algorithm, we get a quadratic function in security parameter, at first sight, it is worse complexity of decryption. In fact that, we improved it about more than 60 factor. For instance, if $\lambda = 72, \theta = 15$, our degree of decryption circuit is 1350, while [15] the one is 103680.

2 Preliminaries

2.1 Notations

For a real number z, we denote by $\lceil z \rceil$, $\lfloor z \rfloor$, $\lfloor z \rceil$ the rounding of a up, down, or to the nearest integer. For integers m, n, we denote the integer sets $\{m, m+1, \ldots, n-1, n\}$ and $\{m, m+1, \ldots, n-1\}$ by $[m, n]$, and $[m, n)$, respectively. For a real number r, we use $r = (\ldots, r_1, r_0.r_{-1}, r_{-2}, \ldots, r_{-n})_Q$ to denote the Q-ary representation of r with n bits of precision after the Q-ary point. When $Q = 2$, it denotes the binary representation of r. Given $x, p \in \mathbb{R}$, we let $[x]_p$ denote the unique number in $(-p/2, p/2]$ that is congruent to x mod p. All logarithms in the text are base-2 unless stated otherwise.

2.2 Lagrange Interpolating Polynomial

The Lagrange interpolating polynomial is the polynomial $f(x)$ of degree $n-1$ that passes through the n points $\{(x_0, y_0 = f(x_0)), \ldots, (x_{n-1}, y_{n-1} = f(x_{n-1}))\}$, and given by $f(x) = \sum_{j=0}^{n-1} f_j(x)$, where

$$f_j(x) = y_j \prod_{0 \le k \le n-1, k \ne j} (\frac{x - x_k}{x_j - x_k}).$$

Our goal of introducing the Lagrange interpolation polynomial is to obtain the mod-Q arithmetic polynomial expression of computing any bit in the binary representation of the integer $x \in [0, \theta]$. For every integer $b \in [0, \theta]$, let $b = (b_{n-1}, \ldots, b_0)_2$, where $n = \lceil \log\theta \rceil$. For each index $t \in [0, n-1]$, we construct a set consisting of integer b and its t-th bit b_t, where $b = 0, 1, ..., \theta$, namely, denote the set as $\{(b, b_t)\}_{b \in [0,\theta]}$ for each t. So for each index $t \in [0, n-1]$, the $\theta + 1$ points set is

$$\{(x_b = b, y_b = b_t)\}_{b \in [0,\theta]} =$$
$$\{(x_0 = 0, y_0 = 0_t), (x_1 = 1, y_1 = 1_t), \ldots, (x_\theta = \theta, y_\theta = \theta_t)\}.$$

If the variable x equates to an integer $b \in [0, \theta]$, for the index $t \in [0, \ldots, n-1]$, the output of the Lagrange interpolating polynomial

$$F_t(x) = \sum_{j=0}^{\theta} y_j \prod_{0 \le k \le \theta, k \ne j} (\frac{x - x_k}{x_j - x_k}) \bmod Q$$

is y_b, which equates to the t-th bit in the binary representation of x. The multiplicative degree of the mod-Q arithmetic circuit is θ.

2.3 Restricted Depth-3 Arithmetic Circuits

In [14], Gentry and Halevi show how to express the decryption function of Gentry's FHE in [2] as a restricted depth-3 circuit over a large enough ring. Here we will show to express the decryption function of FHE over the integers with a large enough message space as a a restricted depth-3 circuit. As mention in [14], By "restricted" means that the bottom sums in depth-3 circuit must come form a fixed (polynomial-size) set $\mathcal{L}$ of polynomials, where $\mathcal{L}$ is independent of the ciphertext.

Definition 1. *(Restricted Depth-3 Circuit) Let $\mathcal{L} = \{L_j(x_1, \cdots, x_n)\}$ be a set of polynomial, all in the same n variables. An arithmetic circuit C is an $\mathcal{L}$-restricted depth-3 circuit over $(x_1, \cdots, x_n)$ if there exists multisets $S_1, \cdots, S_t \subseteq \mathcal{L}$ and constants $\lambda_0, \cdots, \lambda_t$ such that*

$$C(\overrightarrow{x}) = \lambda_0 + \sum_{i=1}^{t} \lambda_i \cdot \Pi_{L_j \in S_i} L_j(x_1, \cdots, x_n)$$

The degree of C with respect to $\mathcal{L}$ is $d = \max_i |S_i|$ (we also call it the $\mathcal{L}$-degree of C).

Remark 1. In all our instantiations of decryption circuits for FHE schemes over the integers, the L_j's happen to be linear.

The following lemma 1 states Ben-or's observation that multilinear symmetric polynomials can be computed by restricted depth-3 arithmetic circuits that perform interpolation. Here recall that a multilinear symmetric polynomials $M(\overrightarrow{x})$ is a symmetric polynomial where, for each i, every monomial is of degree at most 1 in x_i; there are no high powers of x_i.

Lemma 1. *[14]: Let $Q > t2^n$ be a prime, let a set $A \subset \mathbb{Z}_Q$ have cardinality $t2^n + 1$, let $\boldsymbol{x} = (x_1, \ldots, x_{t2^n+1})$ be variables, denote $\mathcal{L}_A = \{(a + x_i : a \in A, 1 \leq i \leq t2^n + 1)\}$. For every multilinear symmetric polynomial $M(\boldsymbol{x})$ over $\mathbb{Z}_Q$, there is a circuit $C(\boldsymbol{x})$ such that:*

- *C is a $\mathcal{L}_A$-restricted depth-3 circuit over $\mathbb{Z}_Q$ such that $C(\boldsymbol{x}) \equiv M(\boldsymbol{x})$ mod Q.*
- *C has $t2^n + 1$ product gates of $\mathcal{L}_A$-degree $t2^n$, one gate for each value $a_j \in A$, with the j-th gate compute the value $\lambda_j \prod_i (a_j + x_i)$ for some constant λ_i.*
- *A description of C can be compute efficiently given the values $M(\boldsymbol{x})$ at all $\boldsymbol{x} = 1^i 0^{t-i}$.*

The final bullet clarifies that Ben-or's oservation is constructive, we can compute the restricted depth-3 representation from any initial representation that lets us evaluate M.

3 Bootstrapping the Decryption

This section deals mainly with how to implement the decryption $m \leftarrow (c \bmod p) \bmod Q$ with a mod-Q arithmetic circuit of a low degree.

3.1 Squashing the Decryption with SSSP Assumption

The decryption circuit is

$$m \leftarrow (c \bmod p) \bmod Q = c - p\lfloor c/p \rceil \bmod Q.$$

Let $\mathbf{y}$ be a vector of Θ rational number in $[0, Q)$ with κ bits of precision after the binary point, and let $\mathbf{sk} = (s_1, s_2, \ldots, s_\Theta)_2$ be the secret key vector of Θ bits with hamming weight θ such that $1/p = \langle \mathbf{sk}, \mathbf{y} \rangle + \varepsilon \bmod Q$, where $|\varepsilon| < 2^{-\kappa}$. We firstly compute $z_i = [c \cdot y_i]$, keeping only $n = \lceil \log(\theta + 1) \rceil$ bits of precision after the binary point for $i = 1, 2, \ldots, \Theta$.

Therefore we have

$$m \leftarrow (c \bmod p) \bmod Q = c - \lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \bmod Q$$

For $i \in [1, \Theta]$, let $z_i = z_i' + z_i'' \cdot 2^{-n} \bmod Q$, where $z_i' \in [0, Q)$ is the integer part of z_i and $z_i'' < 2^n$ is the fractional part. Then we have

$$m \leftarrow (c \bmod p) \bmod Q = c - \sum_{i=1}^{\Theta} s_i (z_i' \bmod Q) - \lfloor 2^{-n} \sum_{i=1}^{\Theta} s_i z_i'' \rceil \bmod Q.$$

3.2 Bootstrapping

For the integer part, it is easy to compute $(\sum_{i=1}^{\Theta} s_i (z_i' \bmod Q) \bmod Q)$ by using some mod-Q addition gates and multiplication-by-constant gates.

For the factional part, in order to compute $\lfloor 2^{-n} \sum_{i=1}^{\Theta} s_i z_i'' \rceil$, here we firstly denote the binary represent of z_i'' as $(z_{i,n-1}'', \ldots, z_{i,1}'', z_{i,0}'')_2$.

(1) $2^{-n} \sum_{i=1}^{\Theta} s_i z_i'' = 2^{-n} \sum_{i=1}^{\Theta} s_i \sum_{j=0}^{n-1} z_{i,j}'' 2^j = \sum_{j=0}^{n-1} 2^{j-n} \sum_{i=1}^{\Theta} s_i z_{i,j}''$. Let n integer numbers $\{W_{-j}\}_{j \in [1,n]}$ such that $W_j = \sum_{i=1}^{\Theta} s_i z_{i,j}''$, namely W_j is the hamming weight of the vector $(s_i z_{1,-j}'', \ldots, s_i z_{\Theta,-j}'')_2$. We can using mod-$Q$ addition gates to sum-up directly the hamming weight.

(2) Since the hamming weight of the secret key vector $\mathbf{sk}$ is θ, then W_j is not bigger than θ, i.e. $W_j \leq \theta$. Let $W_j = (w_{j,n}, \ldots, w_{j,1})_2$. By the Lagrange interpolating polynomial, for $1 \leq t, j \leq n$, the bit values $w_{j,t} = F_t(W_j)$, where the multiplicative degree of Lagrange interpolating polynomial F_t is θ.

(3) Now

$$\sum_{j=0}^{n-1} 2^{j-n} W_j = \sum_{j=0}^{n-1} \sum_{t=0}^{n-1} (2^{j+t-n} \cdot w_{j,t}) = \sum_{j+t \geq n} 2^{j+t-n} w_{j,t} + 2^{-n} \sum_{j+t<n} 2^{j+t} w_{j,t}, \tag{1}$$

thus we have

$$\lfloor 2^{-n} \sum_{i=1}^{\Theta} s_i z_i'' \rceil \bmod Q = \sum_{j+t \geq n} 2^{j+t-n} w_{j,t} + \lfloor 2^{-n} \sum_{j+t<n} 2^{j+t} w_{j,t} \rceil \quad \bmod Q \tag{2}$$

It is easy to compute $\sum_{j+t\geq n} 2^{j+t-n} w_{j,t}$ by using using some mod-Q addition gates and multiplication-by-constant gates. We now show how can compute $\lfloor 2^{-n} \sum_{j+t<n} 2^{j+t} w_{j,t} \rceil \mod Q$ in equation(2) using a $\mathcal{L}_A$-restricted circuit.

Lemma 2. *[14] Let Q be a prime with $Q > t2^n$, there is a univariate polynomial $f(x)$ of degree $\leq t2^n$ such that $f(\sum_{i=0}^{t-1} w_i' u_i) = \lfloor 2^{-n} \sum_{i=0}^{t-1} w_i' u_i \rceil \bmod Q$, where all $|u_i| < 2^n$.*

Lemma 3. *Let t be positive integer and $f(x)$ a univariate polynomial over $\mathbb{Z}_Q$ (for Q prime, $Q > t2^n$). Then there is a multilinear symmetric polynomial M_f on $t2^n$ variables such that*

$$f(\sum_{i=0}^{t-1} w_i' u_i) = M_f(\underbrace{w_0', \ldots, w_0'}_{u_1}, \underbrace{0, \ldots, 0}_{2^n-u_1}, \ldots, \underbrace{w_{t-1}', \ldots, w_{t-1}'}_{u_{t-1}}, \underbrace{0, \ldots, 0}_{2^n-u_{t-1}})$$

for all $w_i' \in \{0,1\}$, and $u_i \in [0, 2^n)$.

By Lemma 2,3, if $Q > \frac{n(n+1)}{2} \cdot 2^n$, there is a multilinear symmetric polynomial $M_f(\cdot)$ to compute the function $\lfloor 2^{-n} \sum_{j+t<n} 2^{j+t} w_{j,t} \rceil \bmod Q$. Then by lemma 1, this multilinear symmetric polynomial $M_f(\cdot)$ can be expressed as $\mathcal{L}_A$-restricted depth-3 circuit C over $\mathbb{Z}_Q$ of degree at most $\frac{n(n+1)}{2} \cdot 2^n$, having at most $\frac{n(n+1)}{2} \cdot 2^n + 1$ product gates.

Thus we obtain the following results: If Q be primes such that $Q > \frac{n(n+1)}{2} \cdot \theta$, the degree of the polynomial in the first step is 1, the degree of the polynomial in the second step is at most θ, the degree of the polynomial in the third step is $\frac{n(n+1)}{2} \cdot \theta$.

Therefore the total degree of the decryption circuit over $\mathbb{Z}_Q$ is bounded by $\theta^2 \frac{n(n+1)}{2} \approx \theta^2 \log^2 \theta/2$. The number of product gates is $\theta \cdot n + (\frac{n(n+1)}{2} \cdot \theta + 1)$. We notes that the complexity of decryption circuit is independent of Q except a constraint $Q > \frac{n(n+1)}{2} \cdot \theta \approx \theta \log^2 \theta/2$

4 Conclusion

We combine the techniques in [15] and [14], such as Lagrange interpolate and the trick to convert a rounding function into restricted depth-3 circuit, reduce the complexity of decryption circuit. The multiplicative degree of the decryption is $\theta^2 \log^2 \theta/2$, and the multiplication gates is $O(\theta \log^2 \theta)$. Then we get a faster bootstrapping procedure than the one proposed in [15]. Table 1 shows that

Asymptotically, the parameter θ, the hamming weight of the secret key vector, can be made as small as $\Theta(\lambda/\log\lambda)$(see e.g. [**?**]). so we can set it to be $\lambda/\log\lambda$, rounded up to the next power of two minus one. For $\lambda = 72$, we have $\lambda/\log\lambda \approx 11.7$, so we set $\theta = 15$. Then the degree of the decryption circuit is $\theta^2 \frac{n(n+1)}{2} = 2250$, while the one in our last work is $108 \cdot \theta \log^3 \theta = 103680$.

The number of product gates is $\theta \cdot n + (\frac{n(n+1)}{2} \cdot \theta + 1) = 211$, while the one in last work is $\theta n' + 8n'^2 + 4n' + 9 = 534$ $(n' = log\theta + 3)$.

Table 1. Complexity of Decryption

	Q	multiplicative degree
[15]	$Q > \theta$	$108\theta\log^3\theta$
$\lambda = 72, \theta = 15$	$Q > 15$	103680
this paper	$Q > \theta\log^2\theta/2$	$\theta^2\log^2\theta/2$
$\lambda = 72, \theta = 15$	$Q > 90$	1350

References

1. Rivest R. L., Adleman L., Dertouzos M. L.: On Data Banks and Privacy Homomorphism. Foundations of Secure Computation, PP: 452–473 (1978)
2. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher,M. (ed.) STOC, pp. 169C178. ACM (2009)
3. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24C43. Springer, Heidelberg (2010)
4. Coron, J.-S., Mandal, A., Naccache, D., Tibouchi, M.: Fully Homomorphic Encryption over the Integers with Shorter Public-Keys. In: Rogaway, P. (ed.) CRYPTO2011. LNCS, vol. 6841, pp. 487C504. Springer, Heidelberg (2011)
5. Coron, J.-S., Naccache, D., Tibouchi, M.: Public key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 446C464. Springer, Heidelberg (2012).
6. Coron, J.-S., Lepoint, T., Tibouchi, M., et al.: Batch fully homomorphic encryption over the integers. In: Johansson, T., Nguyen, P. Q. (eds.) EUROCRYPT 2013. LNCS, vol.7881, pp. 315-335. Springer, Heidelberg (2013).
7. Coron, J.-S., Lepoint, T., Tibouchi, M. Scale-Invariant Fully Homomorphic Encryption over integers. In Krawczyk, H.(ed.) PKC 2014. LNCS, vol. 8383, pp. 311-328, Springer, Heidelberg (2014)
8. Nuida K., Kurosawa K.: (Batch) Fully homomorphic Encryption over integers for Non-Binary Message Spaces. at EUROCRYPT 2015.(1)2015:537–555
9. Cheon J. H., Stehl D.: Fully Homomophic Encryption over the Integers Revisited. at EUROCRYPT2015 (1)2015:513–536
10. Daniel Benarroch, Zvika Brakerski, Tancrde Lepoint: FHE over the Integers: Decomposed and Batched in the Post-Quantum Regime. PKC (2) 2017: 271-301
11. Jinsu Kim, Sungwook Kim, Jae Hong Seo: A new scale-invariant homomorphic encryption scheme. Inf. Sci. 422: 177-187 (2018)
12. Cheon J H, Han K, Kim D. Faster Bootstrapping of FHE over the Integers[J]. IACR Cryptology ePrint Archive, 2017, 2017: 79.
13. Cheon J H, Kim J. A hybrid scheme of public-key encryption and somewhat homomorphic encryption[J]. IEEE transactions on information forensics and security, 2015, 10(5): 1052-1063.
14. Gentry, C., Halevi, S.: Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In FOCS, pp. 107–109(2011)
15. Zhizhu Lian, Yupu Hu, Hu Chen, Baocang Wang. Bootstrapping of FHE over the integers with large message space. Security and Communication Networks 2018.

16. Bhattacharyya A, Indyk P, Woodruff D P, et al. The Complexity of Linear Dependence Problems in Vector Spaces[C]//ICS. 2011: 496-508.