(Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces

Koji Nuida[1,2] and Kaoru Kurosawa[3]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan, k.nuida@aist.go.jp
[2] Japan Science and Technology Agency (JST) PRESTO Researcher
[3] Ibaraki University, Hitachi, Ibaraki 316-8511, Japan, kurosawa@mx.ibaraki.ac.jp

Abstract. In this paper, we construct a fully homomorphic encryption (FHE) scheme over integers with the message space $\mathbb{Z}_Q$ for any prime Q. Even for the binary case $Q = 2$, our decryption circuit has a smaller degree than that of the previous scheme; the multiplicative degree is reduced from $O(\lambda(\log \lambda)^2)$ to $O(\lambda)$, where λ is the security parameter. We also extend our FHE scheme to a batch FHE scheme.

Keywords: Fully homomorphic encryption, non-binary message

1 Introduction

Fully homomorphic encryption (*FHE*) enables computation of any function on the encrypted data. Many FHE schemes appeared recently after the first construction of Gentry [9]. In [9], the following general framework for constructing FHE schemes was also presented. (1) Construct a somewhat homomorphic encryption (SHE) scheme which can evaluate a limited class of functions homomorphically. (2) Transform (or *squash*) the SHE scheme into a bootstrappable scheme whose decryption circuit has a low enough multiplicative degree. (3) Apply Gentry's transformation to get an FHE scheme from the bootstrappable scheme.

At Eurocrypt 2010, van Dijk et al. [8] constructed an "FHE scheme over the integers". At Eurocrypt 2013, Cheon et al. [3] extended it to a batch FHE scheme, where the message space is extended from $\mathbb{Z}_2$ to $(\mathbb{Z}_2)^k$. In [3], they also presented a batch *SHE* scheme for the message space $\mathbb{Z}_{Q_1} \times \cdots \times \mathbb{Z}_{Q_k}$. However, FHE has not been achieved for the case of primes $Q_i > 2$, even for the non-batch case $k = 1$.

1.1 What Is the Problem?

Let λ be the security parameter, and let $\mathcal{M}$ denote the message space. In the scheme of van Dijk et al. [8], $\mathcal{M} = \mathbb{Z}_2$ and the ciphertext c of a plaintext $m \in \mathcal{M}$ is $c = pq + 2r + m$, where p is a secret prime and r is a small noise. In their SHE scheme, the decryption is given by $m = (c \bmod p) \bmod 2 = c - p \cdot$

$\lfloor c/p \rceil \bmod 2 = c - \lfloor c/p \rceil \bmod 2$. In the bootstrappable scheme, the (squashed) decryption algorithm works as

$$m \leftarrow (c \bmod 2) \oplus \left(\left\lfloor \sum_{i=1}^{\Theta} s_i z_i \right\rceil \bmod 2 \right) \ . \tag{1}$$

Here $(s_1, \ldots, s_\Theta) \in \{0,1\}^\Theta$ is the secret key with Hamming weight λ and each $z_i = (z_{i,0}.z_{i,1} \ldots z_{i,L})_2$ is a real number with $L = \lceil \log_2 \lambda \rceil + 3$ bits of precision after the binary point, satisfying $\sum_{i=1}^{\Theta} s_i z_i \approx c/p$.[4] They constructed a low multiplicative degree circuit computing $(\lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \bmod 2)$ in (1) by two steps [8]:

1. The first circuit computes $W_j = \sum_{i=1}^{\Theta} s_i z_{i,j}$ for $j = 0, 1, \ldots, L$. Hence
$$\sum_{i=1}^{\Theta} s_i z_i = W_0 + 2^{-1} W_1 + \ldots + 2^{-L} W_L \ .$$
2. By applying the three-for-two trick repeatedly, the second circuit computes a and b satisfying
$$W_0 + 2^{-1} W_1 + \ldots + 2^{-L} W_L = a + b \bmod 2 \ .$$

The multiplicative degree of the first circuit is λ since $W_j \leq \lambda$, and it is $O((\log \lambda)^2)$ for the second circuit. Hence, the total degree of the decryption circuit is $O(\lambda (\log \lambda)^2)$; see also the second last paragraph of the proof of Theorem 6.2 in [8] for the evaluation of the total degree.

Now to compute W_j, we have to homomorphically compute (at least) a half adder; it needs a pair of polynomials, one for the sum and the other for the carry. However, such a polynomial computing the carry is not known for non-binary cases.[5] This is the main reason why it is hard to extend the circuit above to non-binary message spaces.

1.2 Our Contributions

In this paper, we solve the problem above; for $\mathcal{M} = \mathbb{Z}_Q$ where Q is *any* (constant) prime, we construct an FHE scheme over integers based on a new design principle. We also extend it to a batch FHE scheme with $\mathcal{M} = \mathbb{Z}_{Q_1} \times \cdots \times \mathbb{Z}_{Q_k}$, where $Q_1, \ldots, Q_k$ may be different. Our main advantages are as follows:

1. The FHE scheme for $Q > 2$ was not achieved in [3, 8].
2. Our decryption circuit has multiplicative degree $O(\lambda)$ for any Q; even for $Q = 2$, it is significantly improved from $O(\lambda (\log \lambda)^2)$ of [3, 8].

[4] See [8] for how to compute z_i from c and the public key.

[5] In fact, they exploited the fact that the binary expression of W_j is given by elementary symmetric polynomials in $(s_1 z_{1,j}), \ldots, (s_\Theta z_{\Theta,j})$ [1]. However, such an expression is unknown for non-binary cases.

For $\mathcal{M} = \mathbb{Z}_Q$, the encryption is given by $c = pq + Qr + m$. The decryption of the SHE scheme is given by $m = (c \bmod p) \bmod Q = c - p \cdot \lfloor c/p \rceil \bmod Q$. Then our squashed decryption algorithm works as

$$m \leftarrow c - p \cdot \left\lfloor \sum_{i=1}^{\Theta} s_i z_i \right\rceil \bmod Q \ .^{6}$$

Here, $z_i = (z_{i,0}.z_{i,1} \dots z_{i,L})_Q$ is a real number with $L = \lceil \log_Q \lambda \rceil + 2$ digits of precision after the Q-ary point satisfying $\sum_{i=1}^{\Theta} s_i z_i \approx c/p$.

Now we first determine a polynomial $f(x, y)$ computing the carry of a half adder for any prime Q. Namely, when $x, y \in \mathbb{Z}_Q$ and $x + y = \beta \cdot Q + \alpha \in \mathbb{Z}$, our polynomial f computes $\beta = f(x, y) \bmod Q$.[7] It has degree Q which is proven to be the lowest. See Sec. 3. Then we compute $\sum_{i=1}^{\Theta} s_i z_i = (w_0.w_1 \dots w_L)_Q \bmod Q$ as follows.[8]

- First, we compute the sum of the last digits as shown in Fig. 1 (where each box is a half adder) so that we obtain w_L and the $\Theta - 1$ carries $\beta_1, \dots, \beta_{\Theta-1}$ with
$$s_1 z_{1,L} + \dots + s_\Theta z_{\Theta,L} = w_L + Q \cdot (\beta_1 + \dots + \beta_{\Theta-1}) \ .$$
- Secondly, we compute $(s_1 z_{1,L-1} + \dots + s_\Theta z_{\Theta,L-1}) + (\beta_1 + \dots + \beta_{\Theta-1})$ similarly so that we obtain w_{L-1} and the $2(\Theta - 1)$ carries.
- Iterating this process, we obtain $(w_0.w_1 \dots w_L)_Q$.[9]

The circuit computing each step has multiplicative degree Q ($= \deg f$). Hence, the multiplicative degree D of our decryption circuit is $Q^{L+1} = O(\lambda)$, which is significantly lower than $O(\lambda(\log \lambda)^2)$ of [3, 8].

Finally, in the same way as [3, 8], we make our scheme bootstrappable by letting the bit length of p be $\rho \cdot \Theta(D)$, where ρ is the size of noise r in a fresh ciphertext c. Since the degree D of the decryption circuit has been decreased in comparison to [3, 8], the size of p is also reduced, therefore the size of our ciphertexts is much smaller than that of [3, 8] even for the previously known case $Q = 2$. See Table 1.

Moreover, we emphasize that we also give a *concrete*, not just asymptotic, condition for the parameters of our scheme to make the scheme bootstrappable; see (13).

---

[6] In Sec. 6.1, the information of p is involved in an element X and is reflected by public key components u_ℓ, therefore the decryption does not need p itself.

[7] One may think that this polynomial would be immediately derived from the Witt polynomials [19], which determine the carry functions in the addition of Q-adic integers. However, Q-adic integers are different from Q-ary expression of integers.

[8] For the sake of our analysis in Sec. 7, our algorithm in Sec. 4 is described in a different, but essentially equivalent manner.

[9] We choose the parameters to guarantee that $w_1 \in \{0, Q - 1\}$; consequently $\lfloor \sum_{i=1}^{\Theta} s_i z_i \rceil \equiv w_0 - w_1 \pmod{Q}$.

Table 1. Bootstrappable Bit Lengths of Secret Prime p

	binary message	non-binary message
DGHV'10 [8], CCK+'13 [3]	$\rho \cdot \Theta(\lambda(\log \lambda)^2)$	—
Our result	$\rho \cdot \Theta(\lambda)$	$\rho \cdot \Theta(\lambda)$

1.3 Organization of the Paper

In Sec. 2, we summarize some definitions and notations used in this paper. In Sec. 3, we study the polynomial expression of the carry function in the addition of two Q-ary digits for any prime Q. Based on the result, in Sec. 4, we construct an algorithm for addition of Q-ary integers which is composed of polynomial evaluations modulo Q. Then, in Sec. 5, we recall the previous SHE; and in Sec. 6, we describe our proposed bootstrapping algorithm based on the result in Sec. 4. Finally, in Sec. 7, we analyze our proposed method to verify that the bootstrapping is indeed achieved.

2 Preliminaries

In this paper, we naturally identify the integer residue ring $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$ modulo an integer $n > 0$ with the set $\{0, 1, \ldots, n-1\}$. For real numbers x, y, we write $x \equiv y \pmod{n}$ if $(x-y)/n \in \mathbb{Z}$. On the other hand, we consider the following two kinds of remainder operations; we define $x \bmod n$ to be the unique $y \in \mathbb{Z}_n$ with $y \equiv x \pmod{n}$, and $x \operatorname{Mod} n$ to be the unique integer y in $(-n/2, n/2]$ with $y \equiv x \pmod{n}$.

For a Q-ary representation $A = (a_0.a_1, a_2, \ldots)_Q$ (with $a_j \in \mathbb{Z}_Q$) of a real number A and an integer $L \geq 0$, we define

$$(A)_L := (a_0.a_1, a_2, \ldots, a_L)_Q \ .$$

For a prime Q, an integer a and an integer $b \in \mathbb{Z}_Q$, we define

$$\binom{a}{b}_Q := a(a-1)\cdots(a-b+1) \cdot \mathsf{Inv}_Q(b!) \tag{2}$$

which is a polynomial in a of degree $b \leq Q-1$, where $\mathsf{Inv}_Q(x)$ (for $x \in \mathbb{Z}$ coprime to Q) denotes the unique integer $y \in \mathbb{Z}_Q$ with $xy \equiv 1 \pmod{Q}$. Then we have

$$\binom{a}{b}_Q \equiv \binom{a}{b} \pmod{Q} \tag{3}$$

(the right-hand side is the usual binomial coefficient).

3 Q-ary Half Adder

Let Q be a prime. For $x, y \in \mathbb{Z}_Q$, let

$$x + y = (c, s)_Q = c \cdot Q + s \ .$$

It is clear that $s = x + y \bmod Q$. In this section, we construct the lowest degree polynomial $f_{\mathtt{carry},Q}(x, y)$ yielding the carry c, as follows:

Theorem 1. *We define a polynomial $f_{\mathtt{carry},Q}(x, y)$ over the field $\mathbb{Z}_Q$ by*

$$f_{\mathtt{carry},Q}(x, y) := \sum_{i=1}^{Q-1} \binom{x}{i}_Q \binom{y}{Q-i}_Q \ ,$$

having total degree $\deg f_{\mathtt{carry},Q} = Q$ (see (2) for the notations). Then for any $x, y \in \mathbb{Z}_Q$, we have

$$c = f_{\mathtt{carry},Q}(x, y) \bmod Q \ .$$

Theorem 2. *The total degree of $f_{\mathtt{carry},Q}$ is lowest among all polynomials $g(x, y)$ over $\mathbb{Z}_Q$ satisfying that $c = g(x, y) \bmod Q$ for any $x, y \in \mathbb{Z}_Q$.*

From now, we prove these two theorems. We note that the first part of the proof of Theorem 1 can be also derived by Lucas' Theorem [14]; here we give a direct proof for the sake of completeness.

Proof (Theorem 1). We first show that, for any $x, y \in \mathbb{Z}_Q$,

$$c = \binom{x+y}{Q} \bmod Q \ . \tag{4}$$

If $0 \le x + y < Q$, then we have $c = 0$, while we have $\binom{x+y}{Q} = 0$ by the definition of the binomial coefficient. For the other case $Q \le x + y < 2Q$, we have $c = 1$, while we have

$$\begin{aligned}
\binom{x+y}{Q} &= \binom{x+y}{x+y-Q} \\
&\equiv (x+y)(x+y-1)\cdots(Q+1)\cdot \mathsf{Inv}_Q((x+y-Q)\cdots 1) \\
&\equiv (x+y-Q)(x+y-Q-1)\cdots 1\cdot \mathsf{Inv}_Q((x+y-Q)\cdots 1) \\
&\equiv 1 \pmod{Q}
\end{aligned}$$

(see Sec. 2 for the definition of Inv_Q). Therefore (4) holds.

Next, from the meaning of $\binom{x+y}{Q}$, it is easy to see that

$$\binom{x+y}{Q} = \binom{x}{0}\binom{y}{Q} + \binom{x}{1}\binom{y}{Q-1} + \cdots + \binom{x}{Q}\binom{y}{0} \ .$$

Now we have $0 \le x < Q$ and $0 \le y < Q$ since $x \in \mathbb{Z}_Q$ and $y \in \mathbb{Z}_Q$, therefore $\binom{x}{Q} = \binom{y}{Q} = 0$. Hence, by (3), Theorem 1 holds.

We use the following property in the proof of Theorem 2; we note that the proof of this lemma is similar to the proof of Schwartz–Zippel Theorem [15]:

Lemma 1. *Let $f'(x, y)$ be a polynomial over $\mathbb{Z}_Q$ of degree at most $Q - 1$ with respect to each of x and y. If $f'(x, y) = f_{\mathtt{carry},Q}(x, y)$ for every $x, y \in \mathbb{Z}_Q$, then f' coincides with $f_{\mathtt{carry},Q}$ as polynomials.*

Proof. Assume that $f' \neq f_{\mathtt{carry},Q}$ as polynomials. Set $g := f_{\mathtt{carry},Q} - f'$, which is now a non-zero polynomial. Write $g(x,y) = \sum_{i=0}^{Q-1} g_i(y)x^i$, where each $g_i(y)$ is a polynomial of degree at most $Q-1$. Then g_i is a non-zero polynomial for at least one index i. By the polynomial remainder theorem, we have $g_i(b) = 0$ for at most $Q-1$ elements $b \in \mathbb{Z}_Q$; therefore $g_i(b) \neq 0$ for some $b \in \mathbb{Z}_Q$. Now $g(x,b)$ is a non-zero polynomial in x of degree at most $Q-1$, therefore $g(a,b) \neq 0$ for some $a \in \mathbb{Z}_Q$ by the same reason. On the other hand, we must have $g(x,y) = f_{\mathtt{carry},Q}(x,y) - f'(x,y) = 0$ for any $x, y \in \mathbb{Z}_Q$; this is a contradiction. Hence Lemma 1 holds.

Proof (Theorem 2). If such a polynomial $g(x,y)$ has degree at least Q with respect to x (respectively, y), then $\deg g(x,y)$ can be decreased without changing the values $g(x,y) \bmod Q$ by using the relation $x^Q \equiv x \pmod{Q}$ (respectively, $y^Q \equiv y \pmod{Q}$) derived from Fermat's Little Theorem. Iterating the process, we obtain a polynomial $g_*(x,y)$ of degree at most $Q-1$ with respect to each of x and y, satisfying that $\deg g_* \leq \deg g$ and $g_*(x,y) \equiv g(x,y) \equiv f_{\mathtt{carry},Q}(x,y) \pmod{Q}$ for any $x, y \in \mathbb{Z}_Q$. Then Lemma 1 implies that $g_* = f_{\mathtt{carry},Q}$ as polynomials. Hence we have $\deg f_{\mathtt{carry},Q} = \deg g^* \leq \deg g$. Therefore Theorem 2 holds.

4 Low-Degree Circuit for Sum of Integers

For $i = 1, \ldots, m$, let $\mathsf{a}_i = (a_{i,1}, \ldots, a_{i,n})_Q$. In this section, we give a circuit of low (multiplicative) degree which computes

$$\mathsf{a}_1 + \ldots + \mathsf{a}_m \bmod Q^n. \tag{5}$$

We call the $m \times n$ matrix $A = (a_{i,j})_{i,j}$ the *matrix representation* of $(\mathsf{a}_1, \ldots, \mathsf{a}_m)$.

First we define an algorithm $\mathsf{StreamAdd}_Q(x_1, \ldots, x_m)$ for $x_1, \ldots, x_m \in \mathbb{Z}_Q$ (see also Fig. 1).

$\mathsf{StreamAdd}_Q(x_1, \ldots, x_m)$

$s_2 \leftarrow x_1 + x_2 \mod Q$
$c_2 \leftarrow f_{\mathtt{carry},Q}(x_1, x_2) \bmod Q$ $\quad \%(c_2, s_2)_Q = x_1 + x_2$
For $i = 3, \ldots, m$,
$s_i \leftarrow s_{i-1} + x_i \mod Q$
$c_i \leftarrow f_{\mathtt{carry},Q}(s_{i-1}, x_i) \bmod Q$ $\quad \%(c_i, s_i)_Q = s_{i-1} + x_i$
Return $(s_m, (c_2, \ldots, c_m))$

For $(s_m, (c_2, \ldots, c_m)) \leftarrow \mathsf{StreamAdd}(x_1, \ldots, x_m)$, it is easy to see that

$$x_1 + \cdots + x_m = s_m + Q \times (c_2 + \ldots + c_m) \ . \tag{6}$$

We next define an algorithm $\mathsf{MatrixAdd}_Q(A)$, where $A = (a_{i,j})_{i,j}$ is an $m \times n$ matrix as above.

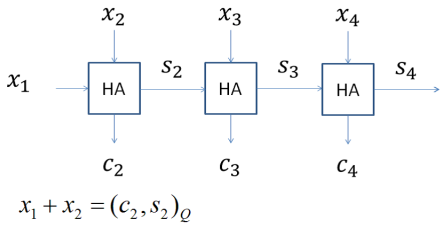

Fig. 1. $\mathsf{StreamAdd}_Q(x_1, \ldots, x_4)$

$\mathsf{MatrixAdd}_Q(A)$

For $j = 1, \ldots, n$,
  $(\alpha_j, (\beta_{2,j}, \ldots, \beta_{m,j})) \leftarrow \mathsf{StreamAdd}_Q(a_{1,j}, \ldots, a_{m,j})$
    % Apply $\mathsf{StreamAdd}_Q$ to the jth column of A.

For $j = 1, \ldots, n-1$,
  $(b_{1,j}, \ldots, b_{m,j}) \leftarrow (\alpha_j, \beta_{2,j+1}, \ldots, \beta_{m,j+1})$
    % Shift $(\beta_{2,j+1}, \ldots, \beta_{m,j+1})^T$ to the left.

Return $B = (b_{i,j})$ and α_n, where B is an $m \times (n-1)$ matrix

Visually, it can be expressed as

$$B = \begin{pmatrix} \ldots, \boxed{\alpha_{n-1}} \\ \ldots, \boxed{\beta_{2,n}} \\ \vdots \quad \vdots \\ \ldots, \boxed{\beta_{m,n}} \end{pmatrix}, \boxed{\alpha_n} \leftarrow (\mathsf{StreamAdd}_Q) \leftarrow \begin{pmatrix} \ldots, \boxed{a_{1,n}} \\ \vdots \quad \vdots \\ \ldots, \boxed{a_{m,n}} \end{pmatrix} = A$$

Given $(B = (b_{i,j}), \alpha_n) \leftarrow \mathsf{MatrixAdd}_Q(A)$, let $\mathsf{b}_i = (b_{i,1}, \ldots, b_{i,n-1}, 0)_Q$ for $i = 1, \ldots, m$. Then from (6), we can see that

$$\mathsf{a}_1 + \ldots + \mathsf{a}_m \equiv (\mathsf{b}_1 + \ldots + \mathsf{b}_m) + \alpha_n \pmod{Q^n} \ .$$

We finally define an algorithm $\mathsf{FinalAdd}_Q(A)$, where $A = (a_{i,j})$ is an $m \times n$ matrix.

$\mathsf{FinalAdd}_Q(A)$

Let $A^{(0)} \leftarrow A$
For $j = 1, \ldots, n$,
  $(A^{(j)}, d_{n-j+1}) \leftarrow \mathsf{MatrixAdd}_Q(A^{(j-1)})$,
    where $A^{(j)}$ is an $m \times (n-j)$ matrix
Return $(d_1, \ldots, d_n)$

Suppose that A is the matrix representation of $(\mathsf{a}_1, \ldots, \mathsf{a}_m)$. Let

$$(d_1, \ldots, d_n) \leftarrow \mathsf{MatrixAdd}_Q(A) \ .$$

Then it is easy to see that the followings hold (since $\deg f_{\mathtt{carry},Q} = Q$):

Theorem 3. *We have* $(d_1, \ldots, d_n)_Q = \mathsf{a}_1 + \ldots + \mathsf{a}_m \bmod Q^n$.

Theorem 4. *For* $i = 1, \ldots, n$, *there is a polynomial* $f_{Q,i}(x_{1,1}, \ldots, x_{m,n})$ *over* $\mathbb{Z}_Q$ *satisfying* $\deg f_{Q,i} = Q^{n-i}$ *and* $d_i = f_{Q,i}(a_{1,1}, \ldots, a_{m,n}) \bmod Q$.

5 Batch SHE Scheme over Integers

In this section, we describe an SHE scheme over integers with message space $\mathcal{M} = (\mathbb{Z}_{Q_1})^{h_1} \times \cdots \times (\mathbb{Z}_{Q_k})^{h_k}$, where $k \geq 1$, $h_j \geq 1$ and $Q_1, \ldots, Q_k$ are distinct primes. This scheme is essentially the one proposed in [3] which is semantically secure under the (ρ, η, γ)-decisional approximate GCD assumption (see [3] for details), with slight notational modifications.[10] [11] To simplify the notations, set

$$\mathcal{I} := \{(i,j) \mid i, j \in \mathbb{Z}, 1 \leq i \leq k, 1 \leq j \leq h_i\} \ .$$

The choices of other parameters ρ, γ, η, τ are discussed later.

– Key generation $\mathsf{KeyGen}(1^\lambda)$: Choose η-bit primes $p_{i,j}$ for $(i,j) \in \mathcal{I}$ uniformly at random in a way that all $p_{i,j}$ and $Q_{i'}$ are different. Choose

$$q_0 \xleftarrow{\$} \Big[1, 2^\gamma / \prod_{(i,j)\in\mathcal{I}} p_{i,j}\Big) \cap \mathsf{ROUGH}(2^{\lambda^2})$$

in a way that q_0 is coprime to all $p_{i,j}$ and all $Q_{i'}$, where $\mathsf{ROUGH}(2^{\lambda^2})$ denotes the set of integers having no prime factors less than 2^{λ^2}. Set

$$N := q_0 \prod_{(i,j)\in\mathcal{I}} p_{i,j} \ .$$

Choose $e_{\xi;0}$ and $e_{\xi;i,j}$ for $\xi \in \{1, \ldots, \tau\}$ and $(i,j) \in \mathcal{I}$ by

$$e_{\xi;0} \xleftarrow{\$} [0, q_0) \cap \mathbb{Z} \, , \ e_{\xi;i,j} \xleftarrow{\$} (-2^\rho, 2^\rho) \cap \mathbb{Z} \ .$$

Then let x_ξ be the unique integer in $(-N/2, N/2]$ satisfying

$$x_\xi \equiv e_{\xi;0} \pmod{q_0} \, , \ x_\xi \equiv e_{\xi;i,j} Q_i \pmod{p_{i,j}} \text{ for } (i,j) \in \mathcal{I} \ .$$

[10] In fact, our proposed bootstrapping method is directly extendable to the variant of their scheme in [3] based on the error-free approximate GCD assumption.

[11] We note that the components of the message space is changed from $(-Q/2, Q/2] \cap \mathbb{Z}$ as in [3] to $\{0, 1, \ldots, Q-1\}$, but it does not affect the security of the scheme. Indeed, by the map $c \mapsto c + \sum_{(i,j)\in\mathcal{I}} \lfloor (Q_i - 1)/2 \rfloor x'_{i,j}$ we can convert any ciphertext with the former message space to that with the latter message space, and vice versa.

Similarly, for $(i,j),(i',j') \in \mathcal{I}$, choose $e'_{i,j;0}$ and $e_{i,j;i',j'}$ by

$$e'_{i,j;0} \xleftarrow{\$} [0, q_0) \cap \mathbb{Z} \,,\; e_{i,j;i',j'} \xleftarrow{\$} (-2^\rho, 2^\rho) \cap \mathbb{Z} \;\;,$$

and let $x'_{i,j}$ be the unique integer in $(-N/2, N/2]$ satisfying

$$\begin{aligned} x'_{i,j} &\equiv e'_{i,j;0} \pmod{q_0} \;\;, \\ x'_{i,j} &\equiv e'_{i,j;i',j'} Q_{i'} + \delta_{(i,j),(i',j')} \pmod{p_{i',j'}} \text{ for } (i',j') \in \mathcal{I} \;\;, \end{aligned}$$

where $\delta_{*,*}$ are Kronecker delta. Then output a public key pk consisting of all N, x_ξ and $x'_{i,j}$, and a secret key sk consisting of all $p_{i,j}$.

- Encryption $\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m})$: Given a plaintext $\boldsymbol{m} = (m_{i,j})_{(i,j)\in\mathcal{I}} \in \mathcal{M}$, output a ciphertext c defined by

$$c := \sum_{(i,j)\in\mathcal{I}} m_{i,j} x'_{i,j} + \sum_{\xi \in T} x_\xi \operatorname{Mod} N \in (-N/2, N/2] \cap \mathbb{Z} \;\;,$$

  where T is a uniformly random subset of $\{1, 2, \ldots, \tau\}$.

- Decryption $\mathsf{Dec}(\mathsf{sk}, c)$: Given a ciphertext c, output $\boldsymbol{m} \in \mathcal{M}$ given by

$$\boldsymbol{m} := ((c \operatorname{Mod} p_{i,j}) \bmod Q_i)_{(i,j)\in\mathcal{I}} \;\;.$$

- Evaluation $\mathsf{Eval}(\mathsf{pk}, f, c_1, \ldots, c_n)$: Given a polynomial f with integer coefficients and ciphertexts $c_1, \ldots, c_n$, output c^* given by

$$c^* := f(c_1, \ldots, c_n) \operatorname{Mod} N \;\;.$$

Following the arguments in [3], we let the parameters ρ, γ, η and τ satisfy the following conditions (see Sec. 6.3 for further details):

- $\rho = \omega(\lambda)$, to resist the attack by Chen and Nguyen [4] for the approximate GCD assumption.
- $\gamma > \eta^2/\rho$, to resist Howgrave-Graham's attack [11] for the approximate GCD assumption.
- $\eta = \Omega(\lambda^2)$ and $\gamma = (\sum_{i=1}^{k} h_i) \cdot \eta + \Omega(\lambda^2)$, to resist Lenstra's elliptic curve method [13] for factoring the integer N (the latter is to make the (approximate) bit length $\gamma - (\sum_{i=1}^{k} h_i) \cdot \eta$ of q_0 sufficiently large).
- $\gamma = \eta^2 \omega(\log \lambda)$, to resist the attack by Cohn and Heninger [5] and the attack using Lagarias algorithm [12] on the approximate GCD assumption. (This implies the condition $\gamma = \Omega(\lambda^3)$ arisen from the general number field sieve [2] for factoring N.)
- $\tau = \gamma + \omega(\log \lambda)$, in order to use the Leftover Hash Lemma in the security proof (see [3] for the details).

6 Our FHE Scheme: Bootstrapping for Large Plaintexts

We now describe our bootstraping algorithm for the SHE scheme in Sec. 5 with non-binary plaintexts, based on our results in Sec. 4.

6.1 Squashed Scheme

In this subsection, we squash the decryption algorithm of the SHE scheme in Sec. 5, i.e., we modify the scheme in such a way that the multiplicative degree of the resulting decryption circuit is low enough to make the bootstrapping possible. It is a natural generalization of the squashing method in [8] to the large message space. The choices of additional parameters κ_i, θ_i, Θ_i and L_i for $i \in \{1, \ldots, k\}$ will be discussed in Sec. 6.3. Set

$$\Theta_{\max} := \max\{\Theta_1, \ldots, \Theta_k\} \ .$$

From now, we describe the squashed scheme.

- Key Generation $\mathsf{KeyGen}^*(1^\lambda)$: First, generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ as in Sec. 5. Then choose a subset Π of the product of symmetric groups $\mathcal{S}_{h_1} \times \cdots \times \mathcal{S}_{h_k}$ satisfying that Π contains the identity permutation id and Π generates the group $\mathcal{S}_{h_1} \times \cdots \times \mathcal{S}_{h_k}$. Secondly, for each $(i, j) \in \mathcal{I}$, choose uniformly at random a Θ_i-bit vector

  $$(s_{i,j;1}, \ldots, s_{i,j;\Theta_i}) \in \{0,1\}^{\Theta_i}$$

  with Hamming weight θ_i, and set

  $$X_{i,j} := \lfloor {Q_i}^{\kappa_i} \cdot (p_{i,j} \bmod Q_i)/p_{i,j} \rceil \ .$$

  For $1 \le i \le k$ and $1 \le \ell \le \Theta_i$, choose

  $$u_{i,\ell} \xleftarrow{\$} [0, {Q_i}^{\kappa_i+1}) \cap \mathbb{Z}$$

  in such a way that

  $$\sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} u_{i,\ell} \equiv X_{i,j} \pmod{{Q_i}^{\kappa_i+1}} \text{ for } 1 \le j \le h_i \ .$$

  Moreover, for each $\sigma = (\sigma_1, \ldots, \sigma_k) \in \Pi$, generate

  $$v_\ell^\sigma \leftarrow \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_\ell^\sigma) \text{ for } 1 \le \ell \le \Theta_{\max} \ ,$$

  where $\boldsymbol{m}_\ell^\sigma = (m_{\ell;i,j}^\sigma)_{(i,j)\in\mathcal{I}} \in \mathcal{M}$ is defined by

  $$m_{\ell;i,j}^\sigma = s_{i,\sigma_i(j);\ell} \text{ if } \ell \le \Theta_i \ , \ m_{\ell;i,j}^\sigma = 0 \text{ otherwise.}$$

  Then output a public key pk^* consisting of all pk, Π, $u_{i,\ell}$ and v_ℓ^σ, and secret key sk^* consisting of all $s_{i,j;\ell}$.
- Encryption $\mathsf{Enc}^*(\mathsf{pk}^*, \boldsymbol{m})$ and evaluation $\mathsf{Eval}^*(\mathsf{pk}^*, f, c_1, \ldots, c_n)$: These are the same as the scheme in Sec. 5 (with public key pk).

– Decryption $\mathsf{Dec}^*(\mathsf{sk}^*, c)$: Given a ciphertext c, for $1 \le i \le k$ and $1 \le \ell \le \Theta_i$, compute

$$z_{i,\ell} := (c \cdot u_{i,\ell}/{Q_i}^{\kappa_i} \bmod Q_i)_{L_i} \ .$$

Then output $\boldsymbol{m} = (m_{i,j})_{(i,j)\in\mathcal{I}}$ defined by

$$m_{i,j} := c - \left\lfloor \sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} z_{i,\ell} \right\rceil \bmod Q_i \text{ for each } (i,j) \in \mathcal{I} \ .$$

We note that, the possible difference of the security of this scheme from the one in Sec. 5 comes from the components $u_{i,\ell}$ involved in the new public key, which are dependent on the secret values $s_{i,j;\ell}$. The situation is the same as the previous FHE schemes [3, 8]. In [8], it was observed that revealing the secret by using the "hints" $u_{i,\ell}$ is related to the sparse subset sum problem and the low-weight knapsack problem. They proposed the following choices of parameters θ_i and Θ_i to avoid the known attacks:

– Θ_i is $\omega(\log \lambda)$ times the bit lengths $(\kappa_i + 1) \log_2 Q_i$ of $u_{i,\ell}$.
– θ_i is large enough to resist brute-force attacks; e.g., $\theta_i := \lambda$ as in [8].

6.2 Our Bootstrapping Procedure

In this subsection, we describe our proposed bootstrapping algorithm based on the results in Sec. 4. More precisely, in the same manner as the previous FHE scheme with modulo-two plaintexts [3], we construct "permuted bootstrapping" algorithm $\mathsf{Bootstrap}(\mathsf{pk}^*, c, \sigma)$ for a ciphertext c for plaintext $(m_{i,j})_{(i,j)\in\mathcal{I}}$ and a permutation $\sigma \in \Pi$, which generates a ciphertext for permuted plaintext $(m_{i,\sigma_i(j)})_{(i,j)\in\mathcal{I}}$ with reduced noise (the case $\sigma = \mathsf{id}$ yields the usual bootstrapping).

Let $\mathsf{StreamAdd}'_Q$ be a variant of the algorithm $\mathsf{StreamAdd}_Q$ defined in Sec. 4, obtained in such a way that the inputs are ciphertexts rather than elements of $\mathbb{Z}_Q$, and the additions and evaluations of the polynomial $f_{\mathtt{carry},Q}$ in $\mathsf{StreamAdd}_Q$ modulo Q are replaced with the corresponding homomorphic evaluations for ciphertexts, i.e., additions and evaluations of $f_{\mathtt{carry},Q}$ "Modulo N". Let $\mathsf{MatrixAdd}'_Q$ and $\mathsf{FinalAdd}'_Q$ be the corresponding variants of $\mathsf{MatrixAdd}_Q$ and $\mathsf{FinalAdd}_Q$, respectively. We construct the algorithm $\mathsf{Bootstrap}(\mathsf{pk}^*, c, \sigma)$ by using $\mathsf{FinalAdd}'_Q$, as follows:

– Permuted bootstrapping $\mathsf{Bootstrap}(\mathsf{pk}^*, c, \sigma)$: First, compute $z_{i,\ell}$ for $1 \le i \le k$ and $1 \le \ell \le \Theta_i$ as in Dec^*, and write

$$z_{i,\ell} = (z_{i,\ell;0}.z_{i,\ell;1}, \ldots, z_{i,\ell;L_i})_{Q_i}$$

where $z_{i,\ell;\xi} \in \mathbb{Z}_{Q_i}$ for each $0 \le \xi \le L_i$. For $0 \le \xi \le L_i$, set

$$v_{i,\ell;\xi} := z_{i,\ell;\xi} \cdot v_\ell^\sigma \bmod N \ . \tag{7}$$

Then compute

$$(w_{i;0}, w_{i;1}, \ldots, w_{i;L_i}) \leftarrow \mathsf{FinalAdd}'_{Q_i}(V_i) \ ,$$

where $V_i = (v_{i,\ell;\xi})_{1 \le \ell \le \Theta_i, 0 \le \xi \le L_i}$ is a $\Theta_i \times (L_i + 1)$ matrix consisting of ciphertexts. Moreover, for $1 \le i \le k$, compute

$$c^{\langle i \rangle} \leftarrow \mathsf{crt}_i \cdot (w_{i;0} - w_{i;1}) \bmod N \ ,$$

where crt_i denotes the unique integer in $(-(\prod_{i'=1}^{k} Q_{i'})/2, (\prod_{i'=1}^{k} Q_{i'})/2]$ with $\mathsf{crt}_i \equiv \delta_{i,i'} \pmod{Q_{i'}}$ for any $1 \le i' \le k$. Finally, output

$$c^* \leftarrow \left(c \bmod \prod_{i=1}^{k} Q_i \right) - c^{\langle 1 \rangle} - \cdots - c^{\langle k \rangle} \bmod N \ .$$

From Theorem 4, the multiplicative degree of the $\mathsf{FinalAdd}'$ circuit is

$$Q_i{}^{L_i+1} = Q_i{}^{\lceil \log_{Q_i} \lambda \rceil + 3} \le Q_i{}^{\log_{Q_i} \lambda + 4} = Q_i{}^4 \cdot \lambda$$

(see Sec. 6.3 below for the choice of L_i) which is $O(\lambda)$ for a constant Q_i.

6.3 Choice of Parameters

We give an instance of the choice of parameters for our scheme, where we regard all of k, Q_i and h_i as constants. First, we set

$$\rho = \Theta(\lambda \log\log\log \lambda), \eta = \Theta(\lambda^2 \log\log \lambda), \gamma = \Theta(\lambda^4 (\log \lambda)^2), \tau = \gamma + \lambda \ .$$

Then all the conditions mentioned in Sec. 5 are indeed satisfied. Secondly, for the additional parameters in the squashed scheme, we set

$$\begin{aligned} &L_i = \lceil \log_{Q_i} \theta_i \rceil + 2 \, , \, \kappa_i = \lceil (\gamma - \log_2(4Q_i - 5)) / \log_2 Q_i \rceil + 2 \ , \\ &\Theta_i = \Theta((\lambda \log \lambda)^4) \, , \, \theta_i = \lambda \text{ for each } 1 \le i \le k \ . \end{aligned} \tag{8}$$

Then the conditions mentioned in Sec. 6.1 are also satisfied. Moreover, the analysis given in Sec. 7 below shows that our scheme is indeed bootstrappable by using these parameters. We emphasize that the order of the bit length η of $p_{i,j}$ is only slightly higher than $\rho \cdot \lambda$, which is significantly lower than $\rho \cdot \lambda (\log \lambda)^2$ required in the previous FHE schemes [3, 8] (see also (13) in Sec. 7 for a concrete lower bound for η). This reduces the key sizes for the scheme, even in the previously achieved cases $Q_i = 2$.

7 Analysis of Our Proposed Scheme

In this section, we analyze our proposed scheme, especially our bootstrapping algorithm, to see the correctness of the scheme and estimate appropriate parameters. For the purpose, we introduce the following definition, which intuitively means the amount of noise in a ciphertext:

Definition 1. *Let c be a ciphertext for plaintext $\boldsymbol{m} = (m_{i,j})_{(i,j)\in\mathcal{I}}$. We define the* weight $\mathsf{wt}_{i,j}(c)$ *of c at position $(i,j) \in \mathcal{I}$ to be the minimum integer satisfying the following for some $\alpha_{i,j}(c), \beta_{i,j}(c) \in \mathbb{Z}$:*

$$c = \alpha_{i,j}(c) \cdot p_{i,j} + \beta_{i,j}(c) \cdot Q_i + m_{i,j} \text{ and } |\beta_{i,j}(c) \cdot Q_i + m_{i,j}| \leq \mathsf{wt}_{i,j}(c) \ .$$

We evaluate the weights of fresh ciphertexts:

Proposition 1. *For any $c \leftarrow \mathsf{Enc}^*(\mathsf{pk}^*, \boldsymbol{m})$ with $\boldsymbol{m} \in \mathcal{M}$, we have $\mathsf{wt}_{i,j}(c) \leq Q_i \Gamma$ for any $(i,j) \in \mathcal{I}$, where we define*

$$\Gamma := \big(h_1(Q_1 - 1) + \cdots + h_k(Q_k - 1) + \tau\big) \cdot 2^\rho \ .$$

Proof. For $(i,j) \in \mathcal{I}$, since N is a multiple of $p_{i,j}$, we have

$$\begin{aligned} c &\equiv \sum_{(i',j')\in\mathcal{I}} m_{i',j'}(e'_{i',j';i,j}Q_i + \delta_{(i',j'),(i,j)}) + \sum_{\xi\in T} e_{\xi;i,j}Q_i \\ &\equiv \left(\sum_{(i',j')\in\mathcal{I}} m_{i',j'}e'_{i',j';i,j} + \sum_{\xi\in T} e_{\xi;i,j}\right) Q_i + m_{i,j} \pmod{p_{i,j}} \ . \end{aligned}$$

Since $m_{i',j'} \in [0, Q_{i'} - 1]$ and $e_{\xi;i,j}, e'_{i',j';i,j} \in (-2^\rho, 2^\rho)$, the absolute value of the right-hand side is bounded by

$$\left(\sum_{(i',j')\in\mathcal{I}} (Q_{i'} - 1) + \tau\right)(2^\rho - 1)Q_i + Q_i - 1 \leq \left(\sum_{i'=1}^{k} h_{i'}(Q_{i'} - 1) + \tau\right) Q_i 2^\rho \ .$$

Hence we have $\mathsf{wt}_{i,j}(c) \leq Q_i\Gamma$, therefore Proposition 1 holds.

The next property is implied directly by the definition of $\mathsf{wt}_{i,j}(c)$:

Proposition 2. *Let c_ℓ be a ciphertext for plaintext $(m_{\ell;i,j})_{(i,j)\in\mathcal{I}}$, $1 \leq \ell \leq n$. Let f be a polynomial, and let f_{abs} denote the polynomial obtained by replacing the coefficients in f with their absolute values. Then the output $c \leftarrow \mathsf{Eval}^*(\mathsf{pk}^*, f, c_1, \ldots, c_\ell)$ of the evaluation algorithm is a ciphertext for plaintext $(f(m_{1;i,j}, \ldots, m_{\ell;i,j}) \mod Q_i)_{i,j} \in \mathcal{M}$, and we have*

$$\mathsf{wt}_{i,j}(c) \leq f_{\mathrm{abs}}(\mathsf{wt}_{i,j}(c_1), \ldots, \mathsf{wt}_{i,j}(c_\ell)) \text{ for any } (i,j) \in \mathcal{I} \ .$$

From now, we show the correctness of the squashed scheme:

Lemma 2. *Let c be a ciphertext for plaintext $\boldsymbol{m} = (m_{i,j})_{(i,j)\in\mathcal{I}}$ with weight $\mathsf{wt}_{i,j}(c)$. Then for any $(i,j) \in \mathcal{I}$, we have*

$$\sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} z_{i,\ell} = p_{i,j} \cdot \alpha_{i,j}(c) + \widetilde{\beta}_{i,j} \cdot Q_i + \varepsilon_{i,j} \tag{9}$$

for the integer $\alpha_{i,j}(c)$ in Definition 1, some integer $\widetilde{\beta}_{i,j}$ and some value $\varepsilon_{i,j}$ with $|\varepsilon_{i,j}| \leq \widetilde{\varepsilon}_{i,j}(c)$, where

$$\widetilde{\varepsilon}_{i,j}(c) := Q_i \mathsf{wt}_{i,j}(c)/(2p_{i,j}) + \theta_i \cdot {Q_i}^{-L_i} + N{Q_i}^{-\kappa_i}/4 \ .$$

Proof. First, by the definition of $z_{i,\ell}$, we have $z_{i,\ell} = c \cdot u_{i,\ell}/{Q_i}^{\kappa_i} + A_{i,\ell}Q_i + \Delta_{i,\ell}$ for some $\Delta_{i,\ell}$ with $|\Delta_{i,\ell}| < {Q_i}^{-L_i}$ and an integer $A_{i,\ell}$. Then we have

$$\sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} z_{i,\ell} = c \cdot \sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} u_{i,\ell}/{Q_i}^{\kappa_i} + A'_{i,j}Q_i + \Delta'_{i,j} \tag{10}$$

for some $A'_{i,j} \in \mathbb{Z}$ and $\Delta'_{i,j} := \sum_{\ell=1}^{\Theta_i} s_{i,j;\ell}\Delta_{i,\ell}$ with $|\Delta'_{i,j}| < \theta_i \cdot {Q_i}^{-L_i}$ (recall that $(s_{i,j;1}, \ldots, s_{i,j;\Theta_i})$ has Hamming weight θ_i). Now, by the definitions of $X_{i,j}$ and $u_{i,\ell}$, there are an integer $B_{i,j}$ and a value $\Delta''_{i,j} \in [-1/2, 1/2]$ satisfying that the right-hand side of (10) is equal to

$$\begin{aligned} &c \cdot ({Q_i}^{\kappa_i} \cdot (p_{i,j} \bmod Q_i)/p_{i,j} + \Delta''_{i,j} + B_{i,j}{Q_i}^{\kappa_i+1})/{Q_i}^{\kappa_i} + A'_{i,j}Q_i + \Delta'_{i,j} \\ &= (p_{i,j} \bmod Q_i) \cdot c/p_{i,j} + (cB_{i,j} + A'_{i,j})Q_i + \Delta'_{i,j} + \Delta''_{i,j}c/{Q_i}^{\kappa_i} \ . \end{aligned}$$

Moreover, by the expression of c as in Definition 1, the right-hand side above is equal to the right-hand side of (9), where

$$\begin{aligned} \widetilde{\beta}_{i,j} &:= cB_{i,j} + A'_{i,j} - \frac{p_{i,j} - (p_{i,j} \bmod Q_i)}{Q_i} \cdot \alpha_{i,j}(c) \in \mathbb{Z} \ , \\ \varepsilon_{i,j} &:= (p_{i,j} \bmod Q_i)(\beta_{i,j}(c) \cdot Q_i + m_{i,j})/p_{i,j} + \Delta'_{i,j} + \Delta''_{i,j}c/{Q_i}^{\kappa_i} \ . \end{aligned}$$

Now, by the definition of $\mathsf{wt}_{i,j}$, we have $|\beta_{i,j}(c) \cdot Q_i + m_{i,j}| \leq \mathsf{wt}_{i,j}(c)$ and

$$|\varepsilon_{i,j}| \leq (Q_i/2) \cdot \mathsf{wt}_{i,j}(c)/p_{i,j} + \theta_i \cdot {Q_i}^{-L_i} + (1/2) \cdot (N/2)/{Q_i}^{\kappa_i} = \widetilde{\varepsilon}_{i,j}(c) \ .$$

Hence, Lemma 2 holds.

Proposition 3. *Let c be as in Lemma 2. If $\widetilde{\varepsilon}_{i,j}(c) < 1/2$ for any $(i,j) \in \mathcal{I}$ (see Lemma 2 for the definition), then $\mathsf{Dec}^*(\mathsf{sk}^*, c)$ outputs $\boldsymbol{m}$ correctly.*

Proof. Recall the result of Lemma 2. Then for $(i,j) \in \mathcal{I}$, we have $|\varepsilon_{i,j}| < 1/2$. Therefore, by Definition 1, we have

$$c - \left\lfloor \sum_{\ell=1}^{\Theta_i} s_{i,j;\ell} z_{i,\ell} \right\rceil \equiv c - p_{i,j} \cdot \alpha_{i,j}(c) \equiv m_{i,j} \pmod{Q_i} \ .$$

Hence, Proposition 3 holds.

From now, in order to analyze our bootstrapping algorithm, we consider the following condition for ciphertexts which is in general stronger than the condition mentioned in Proposition 3 for correct decryption:

Definition 2. *We say that a ciphertext c is* bootstrappable, *if $\widetilde{\varepsilon}_{i,j}(c) < 1/Q_i$ for any $(i,j) \in \mathcal{I}$ (see Lemma 2 for the definition of $\widetilde{\varepsilon}_{i,j}(c)$).*

We analyze the algorithm $\mathsf{Bootstrap}(\mathsf{pk}^*, c, \sigma)$. We assume that c is bootstrappable. For any ciphertext c', let $m(c') = (m(c')_{i,j})_{(i,j)\in\mathcal{I}}$ denote the plaintext for c'. Set

$$w'_i := w_{i;0} - w_{i;1} \bmod N \text{ for any } 1 \leq i \leq k \ .$$

We analyze $\mathsf{FinalAdd}'_{Q_i}(V_i)$ for $1 \le i \le k$. First, for $1 \le j \le h_i$, we have $m(v_{i,\ell;\xi})_{i,j} = s_{i,\sigma_i(j);\ell} z_{i,\ell;\xi}$ for each ℓ, ξ. Therefore, by Theorem 3 (applied to the L_i-digit shift of the sum of $s_{i,\sigma_i(j);\ell} z_{i,\ell}$ to the left), we have

$$\begin{aligned}&(m(w_{i;0})_{i,j}.m(w_{i;1})_{i,j}, \dots, m(w_{i;L_i})_{i,j})_{Q_i} \\ &\equiv \sum_{\ell=1}^{\Theta_i} s_{i,\sigma_i(j);\ell} z_{i,\ell} \equiv p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) + \varepsilon_{i,\sigma_i(j)} \pmod{Q_i}\end{aligned}$$

(see Lemma 2 for the last relation). By Lemma 2, we have $|\varepsilon_{i,\sigma_i(j)}| < 1/Q_i$ since c is bootstrappable. Therefore, one of the followings holds:

$$\begin{cases} m(w_{i;1})_{i,j} = 0 \text{ and } p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) \equiv m(w_{i;0})_{i,j} \pmod{Q_i} \ , \\ m(w_{i;1})_{i,j} = Q_i - 1 \text{ and } p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) \equiv m(w_{i;0})_{i,j} + 1 \pmod{Q_i} \ . \end{cases}$$

In any case, we have

$$m(w'_i)_{i,j} \equiv m(w_{i;0})_{i,j} - m(w_{i;1})_{i,j} \equiv p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) \pmod{Q_i} \ .$$

On the other hand, Definition 1 applied to ciphertexts $w'_{i'}$ implies that

$$c^* = p_{i,j}\alpha_{i,j}(c^*) + (c \operatorname{Mod} Q_1 \cdots Q_k) - \sum_{i'=1}^{k} \mathsf{crt}_{i'} \left(\beta_{i,j}(w'_{i'}) \cdot Q_i + m(w'_{i'})_{i,j} \right) \ ,$$

where $\alpha_{i,j}(c^*) = -\sum_{i'=1}^{k} \mathsf{crt}_{i'} \cdot \alpha_{i,j}(w'_{i'})$. Now, by the definition of $\mathsf{crt}_{i'}$,

$$\begin{aligned}&(c \operatorname{Mod} Q_1 \cdots Q_k) - \sum_{i'=1}^{k} \mathsf{crt}_{i'} \left(\beta_{i,j}(w'_{i'}) \cdot Q_i + m(w'_{i'})_{i,j} \right) \\ &\equiv c - m(w'_i)_{i,j} \equiv c - p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) \equiv m(c)_{i,\sigma_i(j)} \pmod{Q_i}\end{aligned}$$

(note that $c = p_{i,\sigma_i(j)} \cdot \alpha_{i,\sigma_i(j)}(c) + \beta_{i,\sigma_i(j)} \cdot Q_i + m(c)_{i,\sigma_i(j)}$ by Definition 1). Therefore, c^* is a ciphertext for plaintext $(m_{i,\sigma_i(j)}(c))_{(i,j)\in\mathcal{I}}$, with weights satisfying the following (since $|\mathsf{crt}_{i'}| \le Q_1 \cdots Q_k/2$):

$$\begin{aligned}\mathsf{wt}_{i,j}(c^*) &\le \left| (c \operatorname{Mod} Q_1 \cdots Q_k) - \sum_{i'=1}^{k} \mathsf{crt}_{i'} \left(\beta_{i,j}(w'_{i'}) \cdot Q_i + m(w'_{i'})_{i,j} \right) \right| \\ &\le \frac{Q_1 \cdots Q_k}{2} \left(1 + \sum_{i'=1}^{k} \mathsf{wt}_{i,j}(w'_{i'}) \right) \\ &\le \frac{Q_1 \cdots Q_k}{2} \left(1 + \sum_{i'=1}^{k} \left(\mathsf{wt}_{i,j}(w_{i';0}) + \mathsf{wt}_{i,j}(w_{i';1}) \right) \right) \ .\end{aligned}$$

From now, we evaluate the weights $\mathsf{wt}_{i,j}(w_{i';0})$ and $\mathsf{wt}_{i,j}(w_{i';1})$:

Lemma 3. *Let $(i,j) \in \mathcal{I}$. For two ciphertexts c_1, c_2, suppose that $\mathsf{wt}_{i,j}(c_1) \le \alpha$ and $\mathsf{wt}_{i,j}(c_2) \le \beta$, where $1 < \beta < \alpha$. Then we have*

$$\mathsf{wt}_{i,j}(f_{\mathtt{carry},Q}(c_1,c_2)) \le \lfloor Q/2 \rfloor \cdot \frac{\beta}{\beta-1} \cdot \frac{(\alpha/\beta)}{(\alpha/\beta)-1} \cdot \alpha^{Q-1}\beta \ .$$

Proof. First, each monomial in $f_{\mathtt{carry},Q}(t_1,t_2)$ is of the form $a{t_1}^{d_1}{t_2}^{d_2}$ with $|a| \le Q/2$, $d_1, d_2 \in \{1,2,\ldots,Q-1\}$ and $d_1+d_2 \le Q$. Therefore, by Proposition 2, we have

$$\mathsf{wt}_{i,j}(f_{\mathtt{carry},Q}(c_1,c_2)) \le \lfloor Q/2 \rfloor \sum_{\substack{d_1,d_2 \in \{1,2,\ldots,Q-1\} \\ d_1+d_2 \le Q}} \alpha^{d_1}\beta^{d_2} \ .$$

Now the sum in the right-hand side is

$$\sum_{d_1=1}^{Q-1} \alpha^{d_1} \sum_{d_2=1}^{Q-d_1} \beta^{d_2} = \sum_{d_1=1}^{Q-1} \alpha^{d_1} \frac{\beta}{\beta-1}(\beta^{Q-d_1}-1) \le \frac{\beta}{\beta-1} \sum_{d_1=1}^{Q-1} \alpha^{d_1}\beta^{Q-d_1}$$

(where we used the relation $\beta > 1$), and similarly, the sum in the right-hand side above is

$$\alpha\beta^{Q-1} \sum_{d_1=0}^{Q-2} (\alpha/\beta)^{d_1} \le \alpha\beta^{Q-1} \cdot \frac{(\alpha/\beta)^{Q-1}}{(\alpha/\beta)-1} = \alpha^{Q-1}\beta \cdot \frac{(\alpha/\beta)}{(\alpha/\beta)-1}$$

(where we used the relation $\alpha/\beta > 1$). Hence, Lemma 3 holds.

Lemma 4. *Let $A = (a_{\ell,\xi})_{\ell,\xi}$ be a matrix of ciphertexts $a_{\ell,\xi}$ with Θ rows. Let $\mu > 1$. For each $(i,j) \in \mathcal{I}$, if $\mathsf{wt}_{i,j}(a_{\ell,\xi}) \le \mu$ for any ℓ, ξ, then the output $((b_{\ell,\xi})_{\ell,\xi}, d)$ of $\mathsf{MatrixAdd}'_Q(A)$ satisfies that $\mathsf{wt}_{i,j}(d) \le \Theta \cdot \mu$ and*

$$\mathsf{wt}_{i,j}(b_{\ell,\xi}) \le \lfloor Q/2 \rfloor \cdot \frac{\mu}{\mu-1} \cdot \frac{\Theta}{\Theta-1} \cdot \Theta^{Q-1}\mu^Q \text{ for any } \ell, \xi \ . \tag{11}$$

Proof. For the intermediate objects s_ξ and c_ξ in the subroutine $\mathsf{StreamAdd}'_Q$ whose inputs are Θ components of A, we have $\mathsf{wt}_{i,j}(s_\xi) \le \Theta\mu$ for any ξ by the choice of μ, while Lemma 3 with $\alpha := \Theta\mu$ and $\beta := \mu$ implies that $\mathsf{wt}_{i,j}(c_\xi)$ is bounded by the right-hand side of (11). Hence, Lemma 4 holds (note that the right-hand side of (11) is larger than $\Theta\mu$).

Lemma 5. *Let $(i,j) \in \mathcal{I}$ and $1 \le i' \le k$. For $\xi = 1, \ldots, L_{i'}+1$, let $(V^{(\xi)}, w_{i';L_{i'}+1-\xi})$ denote the output of the subroutine $\mathsf{MatrixAdd}'_{Q_{i'}}(V^{(\xi-1)})$ in $\mathsf{FinalAdd}'_{Q_{i'}}(V_{i'})$, where $V^{(0)} := V_{i'}$. Then we have $\mathsf{wt}_{i,j}(w_{i';L_{i'}+1-\xi}) \le (\Xi_{i,i'})^{{Q_{i'}}^{\xi-1}}$, where*

$$\Xi_{i,i'} := \left(\left\lfloor \frac{Q_{i'}}{2} \right\rfloor \frac{Q_{i'}Q_i\Gamma}{Q_{i'}Q_i\Gamma - 1} \cdot \frac{\Theta_{i'}}{\Theta_{i'}-1} \right)^{(Q_{i'}-1)^{-1}} \cdot \Theta_{i'}Q_{i'}Q_i\Gamma$$

(see Proposition 1 for the definition of Γ).

Proof. We show that $\mathsf{wt}_{i,j}(v^{(\xi)}) \leq \mu_\xi/\Theta_{i'}$ for any $0 \leq \xi \leq L_{i'}+1$ and any component $v^{(\xi)}$ of $V^{(\xi)}$, where

$$\mu_\xi := \left(\left\lfloor \frac{Q_{i'}}{2} \right\rfloor \frac{Q_{i'}Q_i\Gamma}{Q_{i'}Q_i\Gamma - 1} \cdot \frac{\Theta_{i'}}{\Theta_{i'}-1}\right)^{1+Q_{i'}+\cdots+{Q_{i'}}^{\xi-1}} \cdot (\Theta_{i'}Q_{i'}Q_i\Gamma)^{{Q_{i'}}^{\xi}} \ .$$

Once this is shown, we have $\mathsf{wt}_{i,j}(w_{i';L_{i'}+1-\xi}) \leq \mu_{\xi-1}$ by Lemma 4, while we have $\mu_\xi \leq (\Xi_{i,i'})^{{Q_{i'}}^{\xi}}$ since $1+Q_{i'}+\cdots+{Q_{i'}}^{\xi-1} \leq {Q_{i'}}^{\xi}/(Q_{i'}-1)$, therefore the claim will follow.

First, for $\xi = 0$, we have $\mu_0/\Theta_{i'} = Q_{i'}Q_i\Gamma$, while we have $\mathsf{wt}_{i,j}(v^{(0)}) \leq Q_{i'}Q_i\Gamma$ by (7) and Proposition 1. Hence, the claim holds for the case.

For $\xi > 0$, we use the induction on ξ. First, we have

$$\frac{\mu_{\xi-1}/\Theta_{i'}}{\mu_{\xi-1}/\Theta_{i'}-1} \leq \frac{Q_{i'}Q_i\Gamma}{Q_{i'}Q_i\Gamma-1}$$

since $\mu_{\xi-1}/\Theta_{i'} \geq \mu_0/\Theta_{i'} = Q_{i'}Q_i\Gamma$. Then by Lemma 4 and the induction hypothesis, we have

$$\mathsf{wt}_{i,j}(v^{(\xi)}) \leq \left\lfloor \frac{Q_{i'}}{2} \right\rfloor \frac{Q_{i'}Q_i\Gamma}{Q_{i'}Q_i\Gamma-1} \cdot \frac{\Theta_{i'}}{\Theta_{i'}-1} \cdot {\Theta_{i'}}^{Q_{i'}-1} \left(\frac{\mu_{\xi-1}}{\Theta_{i'}}\right)^{Q_{i'}} = \frac{\mu_\xi}{\Theta_{i'}} \ .$$

Hence the claim holds for the case, therefore Lemma 5 holds.

By Lemma 5, we have

$$\begin{aligned}\mathsf{wt}_{i,j}(c^*) &\leq \frac{Q_1\cdots Q_k}{2}\left(1+\sum_{i'=1}^{k}\left((\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}}}+(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}}\right)\right)\\ &\leq \frac{Q_1\cdots Q_k}{2}\sum_{i'=1}^{k}\left(1+(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}}}+(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}}\right) \leq Q_1\cdots Q_k\sum_{i'=1}^{k}(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}}}\end{aligned}$$

where we used the relation

$$1+(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}} \leq \left((\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}}\right)^2 \leq \left((\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}}\right)^{Q_{i'}}$$

(note that $(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}-1}} \geq 2$). Summarizing, we have the following result:

Theorem 5. *Suppose that the parameters satisfy*

$$Q_i \cdot Q_1\cdots Q_k\sum_{i'=1}^{k}(\Xi_{i,i'})^{{Q_{i'}}^{L_{i'}}}/(2p_{i,j}) + \theta_i\cdot {Q_i}^{-L_i} + NQ_i{}^{-\kappa_i}/4 < 1/Q_i \quad (12)$$

for any $(i,j) \in \mathcal{I}$ (see Lemma 5 for the definition of $\Xi_{i,i'}$). Then, for any ciphertext c for plaintext $(m_{i,j})_{(i,j)\in\mathcal{I}}$ which is bootstrappable in the sense of Definition 2 and any $\sigma \in \Pi$, the output $c^* \leftarrow \mathsf{Bootstrap}(\mathsf{pk}^*, c, \sigma)$ is a ciphertext for plaintext $(m_{i,\sigma_i(j)})_{(i,j)\in\mathcal{I}}$ which is bootstrappable.

Finally, we investigate the choice of parameters to satisfy the condition (12). First, for the parameters L_i and κ_i in (8), we have

$$\theta_i \cdot {Q_i}^{-L_i} + N{Q_i}^{-\kappa_i}/4 \leq 1/{Q_i}^2 + (4Q_i - 5)/(4{Q_i}^2) = 1/Q_i - 1/(4{Q_i}^2)$$

since $N \leq 2^\gamma$, while we have ${Q_{i'}}^{L_{i'}} \leq \theta_{i'} {Q_{i'}}^3$. On the other hand, we have $p_{i,j} \geq 2^{\eta-1}$ since $p_{i,j}$ is an η-bit prime. Therefore, to satisfy (12), it suffices to satisfy the following (where we used $\theta_{i'} = \lambda$ as in Sec. 6.3):

$$Q_i \cdot Q_1 \cdots Q_k \sum_{i'=1}^{k} (\Xi_{i,i'})^{\lambda {Q_{i'}}^3}/2^\eta \leq 1/(4{Q_i}^2) \ ,$$

or, more strongly,

$$\eta \geq 2 + \log_2({Q_i}^3 \cdot Q_1 \cdots Q_k \cdot k) + \lambda \max_{1 \leq i' \leq k} {Q_{i'}}^3 \log_2 \Xi_{i,i'} \ . \tag{13}$$

From now, we study the asymptotic behavior of the parameters. By using the relation $t/(t-1) \leq e^{(t-1)^{-1}}$ for $t > 1$, we have

$$\Xi_{i,i'} \leq \left(\frac{Q_{i'}}{2} \cdot e^{(Q_{i'} Q_i \Gamma - 1)^{-1} + (\Theta_{i'} - 1)^{-1}} \right)^{(Q_{i'}-1)^{-1}} \Theta_{i'} Q_{i'} Q_i \Gamma \ ,$$

therefore

$$\begin{aligned} \log_2 \Xi_{i,i'} \leq \frac{1}{Q_{i'} - 1} \left(\log_2 Q_{i'} - 1 + \left(\frac{1}{Q_{i'} Q_i \Gamma - 1} + \frac{1}{\Theta_{i'} - 1} \right) \log_2 e \right) \\ + \log_2 \Theta_{i'} + \log_2 Q_{i'} + \log_2 Q_i + \log_2 \Gamma \ . \end{aligned}$$

Moreover, we have

$$\log_2 \Gamma = \rho + \log_2(h_1(Q_1 - 1) + \cdots + h_k(Q_k - 1) + \tau) \ .$$

Now, for the choice of parameters in Sec. 6.3, the term ρ in $\log_2 \Gamma$ is dominant among the terms in the upper bound for $\log_2 \Xi_{i,i'}$ above, therefore it suffices to set $\eta = \omega(\rho \cdot \lambda)$ to satisfy (13) asymptotically. Hence, the choice of parameters in Sec. 6.3 is suitable to enable the bootstrapping.

Acknowledgments. The authors thank Shizuo Kaji, Toshiaki Maeno and Yasuhide Numata for their valuable comments on this work, and thank the members of Shin-Akarui-Angou-Benkyo-Kai for discussions on this work. The authors also thank the anonymous reviewers for their precious comments.

References

1. J. Boyar, R. Peralta, D. Pochuev: On the Multiplicative Complexity of Boolean Functions over the Basis $(\wedge, \oplus, 1)$. Theor. Comput. Sci., 235(1), pp.43–57, 2000.

2. J. Buhler, H. W. Lenstra Jr., C. Pomerance: Factoring Integers with the Number Field Sieve. In: A. Lenstra, H. W. Lenstra Jr. (eds.), The Development of the Number Field Sieve, Lecture Notes in Mathematics vol.1554, pp.50–94, Springer, 1993.
3. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, A. Yun: Batch Fully Homomorphic Encryption over the Integers. In: EUROCRYPT 2013, LNCS 7881, pp.315–335, 2013.
4. Y. Chen, P. Nguyen: Faster Algorithms for Approximate Common Divisors: Breaking Fully-Homomorphic-Encryption Challenges over the Integers. In: EUROCRYPT 2012, LNCS 7237, pp.502–519, 2012.
5. H. Cohn, N. Heninger: Approximate Common Divisors via Lattices. IACR Cryptology ePrint Archive 2011/437, 2011.
6. J.-S. Coron, T. Lepoint, M. Tibouchi: Scale-Invariant Fully Homomorphic Encryption over the Integers. In: PKC 2014, LNCS 8383, pp.311–328, 2014.
7. J.-S. Coron, A. Mandal, D. Naccache, M. Tibouchi: Fully Homomorphic Encryption over the Integers with Shorter Public Keys. In: CRYPTO 2011, LNCS 6841, pp.483–500, 2011.
8. M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan: Fully Homomorphic Encryption over the Integers. In: EUROCRYPT 2010, LNCS 6110, pp.24–43, 2010.
9. C. Gentry: Fully Homomorphic Encryption Using Ideal Lattices. In: STOC 2009, pp.169–178, 2009.
10. C. Gentry, S. Halevi, N. P. Smart: Better Bootstrapping in Fully Homomorphic Encryption. In: PKC 2012, LNCS 7293, pp.1–16, 2012.
11. N. Howgrave-Graham: Approximate Integer Common Divisors. In: CaLC, pp.51–66, 2001.
12. J. C. Lagarias: The Computational Complexity of Simultaneous Diophantine Approximation Problems. SIAM J. Comput., 14(1), pp.196–209, 1985.
13. H. W. Lenstra, Jr.: Factoring Integers with Elliptic Curves. Annals of Math., Second Series, 126(3), pp.649–673, 1987.
14. E. Lucas: Théorie des fonctions numériques simplement périodiques. Amer. J. Math., 1(3), pp.197–240, 1878.
15. J. Schwartz: Fast probabilistic algorithms for verification of polynomial identities. J. ACM 27, pp.701–717, 1980.
16. N. P. Smart, F. Vercauteren: Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In: PKC 2010, LNCS 6056, pp.420–443, 2010.
17. N. P. Smart, F. Vercauteren: Fully Homomorphic SIMD Operations. Des. Codes Cryptography, 71(1), pp.57–81, 2014.
18. R. P. Stanley, Enumerative Combinatorics, Volume I (first edition). Cambridge University Press, 1997.
19. Witt vector. http://en.wikipedia.org/wiki/Witt vector