Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 50-59
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An Improved Multiple to One Fully Homomorphic Encryption on the Integers

Chaoju Hu, Jianwei Zhao

School of Control & Computer Engineering, North China Electric Power University, Baoding, China
Email: zhaojianwei14@sina.com

How to cite this paper: Hu, C.J. and Zhao, J.W. (2018) An Improved Multiple to One Fully Homomorphic Encryption on the Integers. *Journal of Computer and Communications*, **6**, 50-59.
https://doi.org/10.4236/jcc.2018.69005

Received: August 20, 2018
Accepted: September 9, 2018
Published: September 12, 2018

Abstract

The public key of the integer homomorphic encryption scheme which was proposed by Van Dijk *et al.* is long, so the scheme is almost impossible to use in practice. By studying the scheme and Coron's public key compression technique, a scheme which is able to encrypt n bits plaintext once was obtained. The scheme improved the efficiency of the decrypting party and increased the number of encrypting parties, so it meets the needs of cloud computing better. The security of the scheme is based on the approximate GCD problem and the sparse-subset sum problem.

Keywords

Fully Homomorphic Encryption, Multipart to One Fully Homomorphism Encryption, Approximate GCD Problem, Sparse-Subset Sum Problem

1. Introduction

Full homomorphic encryption (FHE) was proposed by Rivest, Adleman, and Dertouzos in 1978 [1]. This encryption method can perform operations on ciphertext. After decryption, the same operation is performed on the corresponding plaintext. The results are consistent. With such characteristics, the data can be encrypted and handed over to the cloud for processing, which not only utilizes the computing power of the cloud, but also reduces the amount of local computing and ensures the security of the data [2] [3].

To this end, many scholars have studied how to construct a homomorphic encryption scheme, and proposed a variety of encryption schemes that satisfy partial homomorphism [4]-[9]. In 2009, Gentry *et al.* [10] proposed the first true homomorphic encryption scheme based on the ideal lattice on a polynomial ring, but because it is too complicated and inefficient, it has larger difficulty in prac-

tical applications. In 2010, Dijk *et al.* [11] improved the above ideal lattice scheme and proposed an integer homomorphic encryption scheme, namely the DGHV scheme. The public key size of this scheme is $\tilde{O}\left(\lambda^{10}\right)$. In 2011, Coron *et al.* [12] optimized the DGHV scheme. For the problem of the large number of public keys in the DGHV scheme, a scheme for generating public key integers in quadratic form was proposed, which shortened the public key length to $\tilde{O}\left(\lambda^{7}\right)$. The following year, Coron *et al.* [13] proposed a "public key compression technique" for the problem of excessive public key elements in the DGHV scheme, shortening the public key length to $\tilde{O}\left(\lambda^{5}\right)$.

Through comparison and research, it is found that the above schemes are composed of one encryption party and also a single decryption party, which is difficult to meet the problem of multi-party interaction in the cloud computing environment. In view of the above problems, this paper studies Coron's public key compression technology, shortens the size of the public key, expands the plaintext space in the scheme to n bits, and expands the number of encryption parties to achieve multiple encryption methods. A solution composed of a decryption party is more in line with the application needs of actual scenarios such as cloud computing. The public key size of this scheme is $\tilde{O}\left(\lambda^{5}\right)$.

2. Basic Symbols and Concepts

2.1. Fully Homomorphic Encryption

A compact encryption scheme E encrypts the plaintext according to the encryption method in scheme E. After the obtained ciphertext is arbitrarily operated, the result is decrypted and the result is the same as the plaintext, and the scheme E is fully homomorphic Encryption scheme. Expressed as a mathematical formula as:

$$Dec\left[f\left(Enc\left(m_1\right),Enc\left(m_2\right),\cdots,Enc\left(m_n\right)\right)\right]=f\left(m_1,m_2,\cdots,m_n\right)$$

Enc is an encryption algorithm, *Dec* is a decryption algorithm, *f* is an arbitrary function, c_n is ciphertext, and m_n is plaintext.

In general, a fully homomorphic encryption algorithm consists of four parts:

Key generation algorithm KeyGen (λ): Generate public key *pk*, private key *sk*.

Encryption algorithm Encrypt (*pk*, *m*): encrypts the plaintext m with the public key *pk* to obtain the ciphertext *c*.

Decryption algorithm Decrypt (*sk*, *c*): Decrypt the ciphertext *c* with the private key *sk* to obtain the plaintext *m*.

The ciphertext calculation algorithm Evaluates $\left(pk,f,c_1,c_2,\cdots,c_n\right)$: the operation of the ciphertext should satisfy:

$$Dec\left[f\left(Enc\left(m_1\right),Enc\left(m_2\right),\cdots,Enc\left(m_n\right)\right)\right]=f\left(m_1,m_2,\cdots,m_n\right)$$

2.2. DGHV Program

In 2010, Dijk, Gentry *et al.* proposed an integer homomorphic encryption

scheme, namely the DGHV scheme, which is no longer based on ideal lattices but on modular operations on integers.

The encryption algorithm of the DGHV scheme is

$$c \leftarrow m + 2r + pq$$

where c is ciphertext, m is plaintext, r is random noise interference, p is a private key, and q is a large positive integer generated during the key generation phase.

The decryption algorithm is

$$(c \bmod p) \bmod 2 = \left(c - p * \frac{c}{p}\right) \bmod 2 = Lsb(c)\, xor Lsb\left(\frac{c}{p}\right)$$

The public key size of this scheme is $\tilde{O}\left(\lambda^{10}\right)$.

In order to ensure security, the approximate maximum common divisor problem is introduced. In the encryption process, some ciphertext x_i encrypted by 0, $\{x_i : x_i = r_i + pq\}$ is added, then reorder x_i so that x_0 is the largest and x_0 is odd. $(x_0 \bmod p)$ is an even number, then the public key $pk = (x_0, x_1, \cdots, x_\tau)$. When encrypting, a subset of the set is randomly added to the ciphertext, and the encryption algorithm is

$$c \leftarrow \left(m + 2r + \sum_{1 \le i \le \tau} x_i\right) \bmod x_0 .$$

The addition of the approximate greatest common divisor problem means that the attack on the program is due to an attack on the approximate greatest common divisor problem, so the scheme is safe [14].

2.3. Many-to-One Homomorphic Encryption

The many-to-one fully homomorphic encryption scheme [15] contains a plurality of encryption parties $P_i (i = 1, 2, \cdots, n)$ and a decryption party P. The plaintext space is M, the public-private key pair (pk_i, sk_i) of the encrypting party, and the public-private key pair (pk, sk) of the decrypting party. And the encryption algorithm $E(\cdot)$ and the corresponding decryption algorithm $D(\cdot)$. In this model, the plaintext $m_i \in M$ of the encryption side, the generated ciphertext c_i is m_i encrypted by pk_i, and needs to satisfy the following properties, where $\oplus$ denotes the operator, $i \neq j$:

1) Both the encrypting party and the decrypting party can use their own private key to decrypt the message encrypted by their own public key, *i.e.*

$$D_{sk_i}\left(E_{pk_i}(m_i)\right) = m_i, \quad D_{sk}\left(E_{pk}(m)\right) = m .$$

2) The encrypting party i cannot use its own private key sk_i to decrypt the message encrypted by the encrypting party j with its own public key pk_j, that is,

$$D_{sk_i}\left(E_{pk_j}(m_j)\right) \neq m_j .$$

3) The message encrypted by the encrypting party i with its own public key pk_i can be decrypted by the decrypting party P with its own private key sk, that is,

$$D_{sk}\left(E_{pk_i}\left(m_i\right)\right)=m_i.$$

4) Different messages encrypted by the encrypting party i with its own public key pk_i have homomorphism under the operation of the decrypting party P, that is,

$$D_{sk}\left(E_{pk_i}\left(m_1\oplus m_2\right)\right)=D_{sk}\left(E_{pk_i}\left(m_1\right)\oplus E_{pk_i}\left(m_2\right)\right).$$

5) Different messages encrypted by different encrypting party i and encrypting party j have homomorphism under the operation of decrypting party P, that is,

$$D_{sk}\left(E_{pk_i}\left(m_1\right)\oplus E_{pk_j}\left(m_2\right)\right)=D_{sk}\left(E_{pk}\left(m_1\oplus m_2\right)\right).$$

3. Improved N-Bit "One-to-One" Homomorphic Encryption Scheme

The DGHV scheme can only encrypt 1 bit of plaintext at a time, and the size of the public key element is too large. This section extends the plaintext space to n bits, and uses the public key compression technique to improve the key generation algorithm, using pseudo-random number generation. The f and the seed se generate a set of integers χ_i having the same number of bits as x_i, so that it is not necessary to store the large integer x_i, and it is only necessary to store the difference between χ_i and x_i as a public key element.

3.1. Program Establishment

KeyGen: randomly generate a large prime number $p\in\left[2^{\eta-1},2^{\eta}\right)$ of length η bits, and calculate $x_0=q_0\cdot p$, where $q_0\in\left[0,2^{\gamma}/p\right)$ is a random odd number. Initialize the pseudo-random number generator f and the seed se, and generate τ integers by using $f\left(se\right)$, that is, $\chi_1,\chi_2,\cdots,\chi_{\tau}$, and calculate

$$\delta_i=\left(\chi_i \bmod p\right)+\xi_i\cdot p-r_i,\ 1\le i\le\tau,$$

where $r_i\in Z\bigcap\left(-2^{\rho},2^{\rho}\right)$, $\xi_i\in Z\bigcap\left[0,2^{\lambda+\eta}/p\right)$. Then $x_i=\chi_i-\delta_i$. The public key $pk=\left(se,x_0,\delta_1,\delta_2,\cdots,\delta_{\tau}\right)$, private key $sk=p$.

Encrypt: Randomly select the integer vector $b=\left(b_i\right)_{1\le i\le\tau}\in\left[0,2^{\alpha}\right)^{\tau}$, randomly select the integer $r\in Z\bigcap\left(-2^{\rho'},2^{\rho'}\right)$, ciphertext

$$c\leftarrow\left(m+2^n\cdot r+2^n\cdot\sum_{1\le i\le\tau}b_i\cdot x_i\right)\bmod x_0.$$

Decrypt: $m\leftarrow\left(c \bmod p\right)\bmod 2^n$.

In order to ensure the security of the scheme, the parameters in the above method need to meet the following restrictions: to resist violent attacks, select $\rho=\omega\left(\log\lambda\right)$; in order to make the compressed decryption circuit belong to the permissible circuit, select $\eta\ge\rho\cdot\Theta\left(\lambda\log^2\lambda\right)$; to resist the lattice-based attack, choose $\gamma=\omega\left(\eta^2\cdot\log\lambda\right)$; apply the residual hash theorem to the approximation of the approximate GCD problem, choose $\alpha\cdot\tau\ge\gamma+\omega\left(\log\lambda\right)$; To ensure correct decryption of ciphertext, select $\eta\ge\rho+\alpha+2+\log_2\tau$; second noise parameter $\rho'=\rho+\alpha+\omega\left(\log\lambda\right)$. In this scheme, the parameters take $\rho=\lambda$,

$$\eta = \tilde{O}\left(\lambda^2\right),\ \gamma = \tilde{O}\left(\lambda^5\right),\ \tau = \tilde{O}\left(\lambda^3\right),\ \alpha = \tilde{O}\left(\lambda^2\right),\ \rho' = \tilde{O}\left(\lambda^2\right).$$

3.2. Proof of Correctness

$$\begin{aligned}
&\left(c \bmod p\right) \bmod 2^n \\
&= \left[\left(m + 2^n \cdot r + 2^n \cdot \sum_{1\le i\le \tau} b_i \cdot x_i\right) \bmod p\right] \bmod 2^n \\
&= \left[\left(m + 2^n \cdot r + 2^n \cdot \sum_{1\le i\le \tau} b_i \cdot \left(\chi_i - \delta_i\right)\right) \bmod p\right] \bmod 2^n
\end{aligned}$$

In which

$$\begin{aligned}
&\left(\chi_i - \delta_i\right) \bmod p = \left(\chi_i - \left(\chi_i \bmod p\right) - \xi_i \cdot p + r_i\right) \bmod p \\
&= \left\{\left[\chi_i - \left(\chi_i \bmod p\right)\right] \bmod p - \left[\xi_i \cdot p\right] \bmod p + r_i\right\} \bmod p \\
&= \left(0 - 0 + r_i\right) \bmod p \\
&= r_i
\end{aligned}$$

So the original

$$\left(c \bmod p\right) \bmod 2^n = \left(m + 2^n \cdot r + 2^n \cdot \sum_{1\le i\le \tau} b_i \cdot r_i\right) \bmod 2^n = m$$

3.3. Test of Homomorphism

There are ciphertext $c_1 = m_1 + 2^n \cdot r_1 + 2^n \cdot \sum_{1\le i\le \tau} b_i \cdot x_i$ and ciphertext $c_2 = m_2 + 2^n \cdot r_2 + 2^n \cdot \sum_{1\le i\le \tau} b_j \cdot x_j$, then

$$\begin{aligned}
&\left[\left(c_1 + c_2\right) \bmod p\right] \bmod 2^n \\
&= \left(\left(\left(m_1 + m_2\right) + 2^n \cdot \left(r_1 + r_2\right) + 2^n \cdot \left(\sum_{1\le i\le \tau} b_i \cdot x_i + \sum_{1\le j\le \tau} b_j \cdot x_j\right)\right) \bmod p\right) \bmod 2^n \\
&= \left(\left(m_1 + m_2\right) + 2^n \cdot \left(r_1 + r_2\right) + 2^n \cdot \left(\sum_{1\le i\le \tau} b_i \cdot r_i + \sum_{1\le j\le \tau} b_j \cdot r_j\right)\right) \bmod 2^n \\
&= m_1 + m_2
\end{aligned}$$

$$\begin{aligned}
&\left[\left(c_1 c_2\right) \bmod p\right] \bmod 2^n \\
&= \left(\left(\left(m_1 + 2^n \cdot r_1\right)\left(m_2 + 2^n \cdot r_2\right) + \left(m_1 + 2^n \cdot r_1\right)\left(2^n \cdot \sum_{1\le j\le \tau} b_j \cdot x_j\right)\right.\right. \\
&\quad \left.\left. + \left(2^n \cdot \sum_{1\le i\le \tau} b_i \cdot x_i\right)\left(m_2 + 2^n \cdot r_2\right) + \left(2^n \cdot \sum_{1\le i\le \tau} b_i \cdot x_i\right)\left(2^n \cdot \sum_{1\le j\le \tau} b_j \cdot x_j\right)\right) \bmod p\right) \bmod 2^n \\
&= \left(\left(m_1 + 2^n \cdot r_1\right)\left(m_2 + 2^n \cdot r_2\right) + \left(m_1 + 2^n \cdot r_1\right)\left(2^n \cdot \sum_{1\le j\le \tau} b_j \cdot r_j\right)\right. \\
&\quad \left. + \left(2^n \cdot \sum_{1\le i\le \tau} b_i \cdot r_i\right)\left(m_2 + 2^n \cdot r_2\right) + \left(2^n \cdot \sum_{1\le i\le \tau} b_i \cdot r_i\right)\left(2^n \cdot \sum_{1\le j\le \tau} b_j \cdot r_j\right)\right) \bmod 2^n \\
&= \left(\left(m_1 + 2^n \cdot r_1\right)\left(m_2 + 2^n \cdot r_2\right)\right) \bmod 2^n \\
&= m_1 m_2
\end{aligned}$$

4. Improved N-Bit "Many-to-One" Fully Homomorphic Encryption Scheme

Based on the scheme given in Section 3.1, this section changes the key generation algorithm, expands the number of encryption parties, and gives a "many-to-one" fully homomorphic encryption scheme for processing n-bit plaintext, and corrects it. Sex and homomorphism have been proved.

4.1. Program Establishment

KeyGen: There are multiple encryption parties $P_i\left(i=1,2,\cdots,n\right)$ and one decryption party P in this scheme. The decryption party P generates the public key $pk=\left(se,x_0,\delta_1,\delta_2,\cdots,\delta_\tau\right)$ from the 3.1 scheme, the encryption side $P_i\left(i=1,2,\cdots,n\right)$ selects the integer $p_i\in\left[2^{\eta_i-1},2^{\eta_i}\right)\cap\left(2Z+1\right)$ as its own key sk_i, then change the order of $\left(\delta_1,\delta_2,\cdots,\delta_\tau\right)$ in the public key randomly to obtain $\overline{pk}=\left(se,x_0,\overline{\delta_1},\overline{\delta_2},\cdots,\overline{\delta_\tau}\right)$, and then randomly selected integer

$$q_{i,0},\cdots,q_{i,\tau}\in Z\cap\left[0,\frac{2^{\gamma_i}}{p_i}\right),$$

randomly select integer $r_{i,0},\cdots,r_{i,\tau}\in Z\cap\left[-2^{\rho_i},2^{\rho_i}\right]$, such that

$$x_{i,j}=\chi_{i,j}-\delta_{i,j}=p_iq_{i,j}\left(\chi_j-\overline{\delta_j}\right)+2^nr_{i,j},1\le j\le\tau_i,x_{i,0}=p_iq_{i,0}x_0+2^nr_{i,0},$$

and $x_{i,0}$ is the largest. Then the public key of P_i is

$$pk_i=\left(se,x_0,x_{i,0},\delta_{i,1},\delta_{i,2},\cdots,\delta_{i,\tau_i}\right).$$

Encrypt: Encryption party P_i randomly selects the integer vector

$$b_i=\left(b_{i,j}\right)_{1\le j\le\tau_i}\in\left[0,2^\alpha\right)^{\tau_i},$$

randomly selects integer $s_i\in Z\cap\left(-2^{\rho_i},2^{\rho_i}\right)$, ciphertext

$$c_i\leftarrow\left(m+2^n\cdot s_i+2^n\cdot\sum\nolimits_{1\le j\le\tau_i}b_{i,j}\cdot x_{i,j}\right)\bmod x_{i,0}.$$

Decrypt: The encryption party P_i can decrypt $sk_i=p_i$ according to $sk_i=p_i$; the decryption party P can decrypt $m_i\leftarrow\left(c_i\bmod p\right)\bmod 2^n$ according to $sk=p$.

4.2. Proof of Correctness

1) $\left(pk,sk\right)$ is the public-private key pair generated by P. It can be seen from 3.2 that P can perform correct encryption and decryption.

2) P_i can correctly decrypt c_i using the key $sk_i=p_i$ prove:

$$c_i=\left(m_i+2^n\cdot s_i+2^n\cdot\sum_{1\le j\le\tau_i}b_{i,j}\cdot x_{i,j}\right)\bmod x_{i,0}$$

Since $x_{i,0}$ is the largest, it can be written as

$$\begin{aligned}c_i&=m_i+2^n\cdot s_i+2^n\cdot\sum_{1\le j\le\tau_i}b_{i,j}\cdot x_{i,j}+k_i\cdot x_{i,0}\\&=m_i+2^n\cdot s_i+2^n\cdot\sum_{1\le j\le\tau_i}b_{i,j}\cdot\left(p_iq_{i,j}\left(\chi_j-\overline{\delta_j}\right)+2^nr_{i,j}\right)+k_i\cdot\left(p_iq_{i,0}x_0+2^nr_{i,0}\right)\end{aligned}$$

Finishing can get $c_i = m_i + 2^n A + p_i B$, where

$A = s_i + \sum_{1 \le j \le \tau_i} b_{i,j} \cdot 2^n r_{i,j} + k_i r_{i,0}$, $B = 2^n \cdot \sum_{1 \le j \le \tau_i} b_{i,j} \cdot q_{i,j} \left(\chi_j - \overline{\delta_j} \right) + k_i q_{i,0} x_0$.

According to the defined parameters, $m_i + 2^n A < p_i$, then

$$\left(c_i \bmod p_i \right) \bmod 2^n = m_i .$$

So P_i can correctly decrypt c_i using the key $sk_i = p_i$.

3) P can correctly decrypt c_i using the key $sk = p$ prove:

As can be seen from 2),

$$c_i = m_i + 2^n \left(s_i + \sum_{1 \le j \le \tau_i} b_{i,j} \cdot 2^n r_{i,j} + k_i r_{i,0} \right) + p_i \left(2^n \cdot \sum_{1 \le j \le \tau_i} b_{i,j} \cdot q_{i,j} \left(\chi_j - \overline{\delta_j} \right) + k_i q_{i,0} x_0 \right)$$

According to 3.2, $\left(\chi_j - \overline{\delta_j} \right) \bmod p = r_j$, $\chi_j - \overline{\delta_j}$ can be written as $r_j + l_j \cdot p$, $l_j \in Z$, and $x_0 = q_0 \cdot p$, so it is sorted into: $c_i = m_i + 2^n A + pB$, where

$$A = s_i + \sum_{1 \le j \le \tau_i} b_{i,j} \cdot \left(2^n r_{i,j} + p_i \cdot q_{i,j} \cdot r_j \right) + k_i r_{i,0},$$

$$B = p_i \cdot 2^n \sum_{1 \le j \le \tau_i} b_{i,j} \cdot q_{i,j} \cdot l_j + k_i q_{i,0} q_0 .$$

According to the defined parameters, $m_i + 2^n A < p$, then $\left(c_i \bmod p \right) \bmod 2^n = m_i$. So P can correctly decrypt c_i using the key $sk = p$.

4.3. Test of Homomorphism

1) P has the homomorphism of the decrypted ciphertext. Proof from 3.3 is known.

2) The encrypting party P_i has homomorphism to the encrypted ciphertext. Proof: From 4.2 (2), $c_i = m_i + 2^n A + p_i B$, with ciphertext $c_{i,1} = m_{i,1} + 2^n A_1 + p_i B_1$ and ciphertext $c_{i,2} = m_{i,2} + 2^n A_2 + p_i B_2$, then

$$\begin{aligned}
&\left(\left(c_{i,1} + c_{i,2} \right) \bmod p_i \right) \bmod 2^n \\
&= \left(\left(m_{i,1} + 2^n A_1 + p_i B_1 + m_{i,2} + 2^n A_2 + p_i B_2 \right) \bmod p_i \right) \bmod 2^n \\
&= \left(\left(m_{i,1} + m_{i,2} + 2^n \left(A_1 + A_2 \right) + p_i \left(B_1 + B_2 \right) \right) \bmod p_i \right) \bmod 2^n \\
&= m_{i,1} + m_{i,2}
\end{aligned}$$

$$\begin{aligned}
&\left(\left(c_{i,1} c_{i,2} \right) \bmod p_i \right) \bmod 2^n \\
&= \left(\left(m_{i,1} + 2^n A_1 + p_i B_1 \right) \left(m_{i,2} + 2^n A_2 + p_i B_2 \right) \bmod p_i \right) \bmod 2^n \\
&= \left(\left(m_{i,1} m_{i,2} + 2^n \left(m_{i,1} A_2 + A_1 m_{i,2} + A_1 2^n A_2 \right) \right. \right. \\
&\quad \left. \left. + p_i \left(m_{i,1} B_2 + 2^n A_1 B_2 + B_1 \left(m_{i,2} + 2^n A_2 + p_i B_2 \right) \right) \right) \bmod p_i \right) \bmod 2^n \\
&= m_{i,1} m_{i,2}
\end{aligned}$$

Therefore, P_i has homomorphism to the encrypted ciphertext.

3) The decryption party P has homomorphism to the encrypted ciphertext. Proof: From 4.2 (3), $c_i = m_i + 2^n A + pB$, with ciphertext $c_{i,1} = m_{i,1} + 2^n A_1 + pB_1$ and ciphertext $c_{i,2} = m_{i,2} + 2^n A_2 + pB_2$, the

$$
\begin{aligned}
&\left(\left(c_{i,1}+c_{i,2}\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_{i,1}+2^n A_1+pB_1+m_{i,2}+2^n A_2+pB_2\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_{i,1}+m_{i,2}+2^n\left(A_1+A_2\right)+p\left(B_1+B_2\right)\right) \bmod p\right) \bmod 2^n \\
&=m_{i,1}+m_{i,2}
\end{aligned}
$$

$$
\begin{aligned}
&\left(\left(c_{i,1} c_{i,2}\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_{i,1}+2^n A_1+pB_1\right)\left(m_{i,2}+2^n A_2+pB_2\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_{i,1} m_{i,2}+2^n\left(m_{i,1} A_2+A_1 m_{i,2}+A_1 2^n A_2\right)\right.\right. \\
&\quad \left.\left.+p\left(m_{i,1} B_2+2^n A_1 B_2+B_1\left(m_{i,2}+2^n A_2+pB_2\right)\right)\right) \bmod p\right) \bmod 2^n \\
&=m_{i,1} m_{i,2}
\end{aligned}
$$

Therefore, the decryption party P has homomorphism to the encrypted ciphertext.

4) The decryption party P has homomorphism for different encryption parties P_i and the ciphertext of the encryption party P_j. Proof: From 4.2 (3), $c_i = m_i + 2^n A + pB$, with ciphertext $c_i = m_i + 2^n A_i + pB_j$ and ciphertext $c_j = m_j + 2^n A_j + pB_j$, then

$$
\begin{aligned}
&\left(\left(c_i+c_j\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_i+2^n A_i+pB_i+m_j+2^n A_j+pB_j\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_i+m_j+2^n\left(A_i+A_j\right)+p\left(B_i+B_j\right)\right) \bmod p\right) \bmod 2^n \\
&=m_i+m_j
\end{aligned}
$$

$$
\begin{aligned}
&\left(\left(c_i c_j\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_i+2^n A_i+pB_j\right)\left(m_j+2^n A_j+pB_j\right) \bmod p\right) \bmod 2^n \\
&=\left(\left(m_i m_j+2^n\left(m_i A_j+A_i m_j+A_i 2^n A_j\right)\right.\right. \\
&\quad \left.\left.+p\left(m_i B_j+2^n A_i B_j+B_i\left(m_j+2^n A_j+pB_j\right)\right)\right) \bmod p\right) \bmod 2^n \\
&=m_i m_j
\end{aligned}
$$

Therefore, the decryption party P has homomorphism to different encryption parties P_i and the ciphertext of the encryption party P_j.

4.4. Compression and Decryption Circuit

In order to avoid excessive noise generated during the encryption process and affect the correctness of the homomorphic operation, the ciphertext needs to be re-encrypted, and the condition of re-encryption is that the decryption circuit can be operated in the Evaluate algorithm. This requires that the depth of the decryption circuit is less than the maximum depth allowed by the Evaluate algorithm. Therefore, the decryption circuit needs to be compressed to preprocess some of the calculations in the decryption circuit.

KeyGen: With the KeyGen algorithm in Section 4.1, generate

$$pk_i^* = \left(se, x_0, x_{i,0}, \delta_{i,1}, \delta_{i,2}, \cdots, \delta_{i,\tau_i}\right).$$

On the basis of this, add three parameters κ, θ, Θ, and randomly generate a bit vector $s = \left(s_{i,1}, s_{i,2}, \cdots, s_{i,\Theta}\right)$ with length Θ. Its Hamming weight is θ.

The pseudo random number generator f_2 and the seed se_2 are initialized, and an integer $u_{i,j} \in \left[0, 2^{\kappa+1}\right)$, $2 \le j \le \Theta$ is generated by using $f_2\left(se_2\right)$.

$$\sum_{1 \le j \le \Theta} s_{i,j} \cdot u_{i,j} = x_{p_i} \bmod 2^{\kappa+1},$$

where $x_{p_i} = \lfloor 2^{\kappa}/p_i \rfloor$. Let $y_{i,j} = u_{i,j} \big/ 2^{\kappa}$.

Initialize pseudo-random number generator f_3 and seed se_3, use $f_3\left(se_3\right)$ to generate integer $\chi'_{i,j} \in \left[0, 2^{\gamma_i}\right)$, and randomly generate integer $r'_{i,j} \in \left(-2^{\rho_i}, 2^{\rho_i}\right)$, $\xi'_{i,j} \in \left[0, 2^{\gamma_i+\eta_i} \big/ p_i\right)$, calculate

$$\delta'_{i,j} = \left(\chi'_{i,j} \bmod p_i\right) + \xi'_{i,j} \cdot p_i - 2r'_{i,j} - s_{i,j},$$

then the result of the vector s is encrypted $\sigma_{i,j} = \chi'_{i,j} - \delta'_{i,j}$. The public key $pk = \left(pk_i^*, se_2, u_{i,1}, se_3, \delta'_{i,j}\right)$, and the private key $sk = s$.

Encrypt: Encrypt the plaintext m_i^* with an encryption algorithm to obtain the ciphertext c_i^*, and find $z_{i,j} = \left(c_i^* \cdot y_{i,j}\right) \bmod 2^n$. The ciphertext c_i^* and the extended ciphertext $z = \left(z_{i,1}, z_{i,2}, \cdots, z_{i,\Theta}\right)$ are the output.

Decrypt: Decrypt, output secret civilization

$$m_i = \left(c_i^* - \left(\sum_{1 \le j \le \Theta} s_{i,j} \cdot z_{i,j}\right)\right) \bmod 2^n.$$

This results in a fully homomorphic encryption scheme.

5. Conclusion

Based on the DGHV scheme and the public key compression technology of Coron *et al.*, this paper improves the encryption process, expands the number of encryption parties, and builds a multi-party encryption with a smaller public key size. The integer-homomorphic encryption scheme can encrypt the n-bit plaintext at a time, which is more in line with the needs of practical applications such as cloud computing. Whether it can further reduce the amount of calculation, whether it can achieve "multi-party encryption, multi-party decryption" will be the direction that will be improved in the future.

Conflicts of Interest

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

References

[1] Rivest, R.L., Adleman, L. and Dertouzos, M.L. (1978) On Data Banks and Privacy Homomorphisms. In: *Foundations of Secure Computation*, Academia Press, Ghent, 169-179.

[2] Chen, Z.-G., Wang, J. and Song, X.-X. (2014) Survey on Fully Homomorphic En-

cryption. *Application Research of Computers*, **31**, 1624-1631.

[3] Lin, C., Su, W.-B., Meng, K., Liu, Q. and Liu, W.-D. (2013) Cloud Computing Security: Architecture, Mechanism and Modeling. *Chinese Journal of Computers*, **36**, 1765-1784.

[4] Elgamal, T. (1984) A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, **31**, 469-472.

[5] Fellows, M.R. and Koblitz, N. (1994) Combinatorial Cryptosystems Galore! *Contemporary Mathematics*, **168**, 51-62. https://doi.org/10.1090/conm/168/01688

[6] Benaloh, J. (1994) Dense Probabilistic Encryption. *Proceedings of the Workshop on Selected Areas of Cryptography*, Kingston, 1994, 120-128.

[7] Okamoto, T. and Uchiyama, S. (1998) A New Public-Key Cryptosystem as Secure as Factoring. In: Nyberg, K., Ed., *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 308-318.

[8] Naccache, D. and Stern, J. (1998) A New Public Key Cryptosystem Based on Higher Residues. *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, CA, USA, 2-5 November 1998, 59-66. https://doi.org/10.1145/288090.288106

[9] Damgård, I. and Jurik, M. (2001) A Generalisation, a Simplication and Some Applications of Paillier's Probabilistic Public-Key System. In: *International Workshop on Practice and Theory in Public Key Cryptography*: *Public Key Cryptography*, Springer-Verlag, Berlin, 119-136.

[10] Gentry, C. (2009) Fully Homomorphic Encryption Using Ideal Lattices. *ACM Symposium on Theory of Computing*, *STOC* 2009, Bethesda, MD, USA, 31 May-2 June 2009, 169-178.

[11] Dijk, M.V., Gentry, C., Halevi, S., *et al.* (2010) Fully Homomorphic Encryption over the Integers. *Lecture Notes in Computer Science*, **6110**, 24-43. https://doi.org/10.1007/978-3-642-13190-5_2

[12] Coron, J.S., Mandal, A., Naccache, D., *et al.* (2011) Fully Homomorphic Encryption over the Integers with Shorter Public Keys. In: Rogaway, P., Ed., *CRYPTO* 2011. *LNCS*, Vol. 6841, 487-504.

[13] Coron, J., Naccache, D. and Tibouchi, M. (2012) Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers. In: David, P. and Thomas, J., Eds., *Advances in Cryptology-EUROCRYPT* 2012, Springer, Berlin, Heidelberg, 446-464.

[14] Tang, Q.-Y. and Ma, C.-G. (2014) Feedback Attack against Fully Homomorphic Encryption System. *Computer Engineering*, **40**, 79-84.

[15] Xia, C. (2013) Research of Homomorphic Encryption Technology and Application. Anhui University, Hefei.