International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.365-374
http://dx.doi.org/10.14257/ijsia.2014.8.6.31

Improved Fully Homomorphic Encryption over the Integers with Shorter Public Keys

Can Xiang[1,2] and Chun-ming Tang[1,2,3,*]

[1]*School of Mathematics and Information Science, Guangzhou University, guangzhou 510006, P.R.China*
[2]*Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong Higher Education Institutes, Guangzhou University, Guangzhou, P.R. China*
[3]*State Key Laboratory of Information Security, Beijing 100093, P.R.China*
**Corresponding E-mail: ctang@gzhu.edu.cn*

Abstract

Fully homomorphic encryption (FHE) is a "holy grail" of cryptography. However, it is not yet adopted in practice because no known scheme is efficient. In this paper, we mainly focus on how to reduce the public key sizes in FHE. Based on Dijk et al.'s FHE scheme (*DGHV*) and *Gentry's fully homomorhpic technology, we propose two schemes with shorter public keys by encrypting with a combination of the public key elements to replace a linear form in DGHV scheme and a quadratic form in Coron et al.'s scheme. Compared with DGHV scheme and Coron et al.'s scheme, our schemes can greatly reduce the public key sizes, which make our schemes more efficient. Furthermore, we prove the security of our schemes based on difficulties of the approximate greatest common divisor (AGCD) problem and the sparse subset sum problem.*

Keywords: *Fully Homomorphic Encryption, Shorter Public Key Size, AGCD problem*

1. Introduction

Fully Homomorphic Encryption (FHE) has been referred to as a "holy grail" of cryptography. In a nutshell, a FHE scheme is an encryption scheme that allows evaluation of arbitrarily complex programs on encrypted data. The problem was suggested by Rivest, *et al.*, [1] back in 1978, yet the first plausible candidate came thirty years later with Gentry's breakthrough work in 2009 [2]. There has been partial progress in the meanwhile [3-13].

In encryption schemes, Bob encrypts a plaintext message to obtain a ciphertext. Alice decrypts the ciphertext to recover the plaintext. In FHE, parties that do not know the plaintext data can perform computations on it by performing computations on the corresponding ciphertexts.

A major application of FHE is to cloud computing. Alice can store her data in the cloud, for example, on remote servers that she accesses via the Internet. The cloud has more storage capabilities and computing power than does Alice, so when Alice needs computations to be done on her data, she would like those computations to be done by the cloud. However, Alice doesn't trust the cloud. Her data might be sensitive (for example, Alice might be a hospital and the data might be patients' medical records), and Alice would like the cloud to know as little as possible about her data, and about the results of the computations. So Alice sends encrypted data to the cloud, which can

ISSN: 1738-9976 IJSIA

perform arithmetic operations on it without learning anything about the original raw data, by performing operations on the encrypted data.

FHE can be used to query a search engine, without revealing what is being searched for (here, the search engine is doing the computations on encryptions of information that it doesn't know).

1.1. Related Work

In [2], Gentry utilized ideal lattice to construct a FHE scheme, the scheme mainly used addition and multiplication over the ideal lattices of a polynomial ring, each cycle needs to calculate each element respectively, and it needs to deal with itself after calculating the matrix. In addition, each time it can only encrypt a single bit, thus the algorithm complexity is high and the practicality is not strong. According to Gentry's ideas, Dijk, *et al.*, described a FHE scheme over the integers [3], denoted by DGHV scheme. Compared with Gentry's construction, this somewhat homomorphic scheme merely used addition and multiplication over the integers rather than working with ideal lattices over a polynomial ring [2]. The main advantage of the approach was the conceptual simplicity, namely all operations were done over the integers instead of ideal lattices. However, the public-key was in $O(\lambda^{10})$ which was too large for any practical system. Later, Coron, *et al.*, improved the method of generating public keys of the DGHV scheme, they reduced the public key size of the somewhat homomorphic scheme (SWHE) from $O(\lambda^{10})$ down to $O(\lambda^{7})$ in [4]. Namely they used a quadratic form in the public key elements, instead of a linear form in the DGHV scheme. Meanwhile, they require using an error-free x_0, where $x_0 = q_0 p$, otherwise the error would grow too large. However, the security of this scheme was based on the partially approximate common divisors (PACD) problem rather than the AGCD problem. At Eurocrypt 2012, Yuanmi Chen and Phong Q. Nguyen described a faster algorithm for PACD problem in [5], the safety will suffer the serious threat for the scheme over the integers in [4].

1.2. Our Contributions

In this paper, we put forward two improved fully homomorphic encryption schemes over the integers with shorter public keys, the security of our schemes are based on both the AGCD problem and the sparse subset sum problem (SSSP). Our schemes reach the security level of the in distinguish ability against chosen ciphertext attacks (IND-CPA).

Firstly, we reduce the public key sizes of the somewhat homomorphic scheme in the DGHV scheme from $O(\lambda^{10})$ to $O(\lambda^{5}\log\lambda)$ and $O(\lambda^{5})$ respectively. The main idea for the reduction of public key size is that storing only a smaller set and then generating the full public keys on the fly by adding all the elements of each combination in the set, that is, each public key is the sum of all elements of each subset of the small set. Moreover, we prove that our schemes are still semantically secure under the AGCD problem assumption. However, the SWHE schemes only supports a limited number of additions and multiplications on ciphertexts, since every ciphertext has noise and any homomorphic operation applied to ciphertexts increases the noise in the resulting ciphertext. Once this noise reaches a certain threshold, the resulting ciphertext can not be decrypted correctly; it will limit the degree of the polynomial that can be applied to ciphertexts. Secondly, we can squash the decryption circuit to get a bootstrappable scheme for our two SWHE schemes, and then invoke Gentry's bootstrapping theorem to obtain a fully homomorphic public-key encryption schemes.

1.3. Organization

The rest of the paper is organized as follows: In Section 2, we recall some notations and definitions. In Section 3, we review somewhat homomorphic schemes in the DGHV scheme and design two somewhat homomorphic schemes (SWHE) over the integers with shorter public keys, and we also prove that our schemes are semantically secure based on the AGCD problem. In section 4, we make our two SWHE schemes fully homomorphic schemes with Gentry's technique. In Section 5, concludes are given.

2. Preliminaries

For a real number z, we denote by $\lceil z \rceil$, $\lfloor z \rfloor$, $\lfloor z \rceil$ the rounding of an up, down, or to the nearest integer. Namely, they are the unique integers in the half open intervals $[z, z+1)$, $(z-1, z]$ and $(z-1/2, z+1/2]$ respectively. For a real number z and an integer p, we use $q_p(z)$ and $r_p(z)$ to denote the quotient and remainder of z with respect to p, namely $q_p(z) = \lfloor z/p \rceil$ and $r_p(z) = z - q_p(z) \cdot p$. We also denote the remainder by $[z]_p$ or (z mod p), we use these notations interchangeably throughout the paper.

All logarithms in the text are base-2 unless stated otherwise. A homomorphic public key encryption scheme ε has four algorithms: the usual KeyGen, Encrypt, Decrypt, and Evaluate algorithm. The algorithm Evaluate takes as input a public key pk, a circuit C and a tuple of ciphertexts $\vec{c} = (c_1, c_2, \cdots, c_t)$ (one for every input bit of C), outputs another ciphertext c^*.

Definition 2.1 Correct Homomorphic Encryption

The scheme $\varepsilon = (KeyGen, Encrypt, Decrypt, Evaluate)$ is correct for a given t-input circuit C if for any key-pair (sk, pk) output by KeyGen(λ), any t plaintext bits $m_1, m_2, \cdots, m_t$, and any ciphertexts $\vec{c} = (c_1, c_2, \cdots, c_t)$ with $c_i = Encrypt(\mathrm{pk}, \mathrm{m}_i)$, it is the case that: $\mathrm{Dncrypt}(sk, \mathrm{Evaluate}(pk, C, (c_1, c_2, \cdots, c_t))) = C(m_1, m_2, \cdots m_t)$ .The circuit C is called permitted circuits of the scheme ε, we denote the set of this permitted circuits C by $C_\varepsilon(\lambda)$.

Definition 2.2 Fully Homomorphic Encryption (FHE)

The scheme $\varepsilon = (KeyGen, Encrypt, Decrypt, Evaluate)$ is fully homomorphic if any element of the Permitted circuits set $D_\varepsilon(\lambda)$ can be any circuit. In other words, ε is fully homomorphic if it is correct for all boolean circuits, otherwise ε is a somewhat homomorphic encryption scheme (SWHE).

Definition 2.3 Augmented Decryption Circuits

Let ε be an encryption scheme, where decryption is implemented by a circuit that depends only on the security parameter. For a given value of the security parameter λ, the set of augmented decryption circuits consists of two circuits, both of which take as input a secret key and two ciphertexts: one circuit decrypts both ciphertexts and adds the resulting plaintext bits mod 2, the other decrypts both ciphertexts and multiplies the resulting plaintext bits mod 2. We denote this set by $D_\varepsilon(\lambda)$.

Definition 2.4 Bootstrappable Encryption Scheme

Let ε be a homomorphic encryption scheme, for each value of the security parameter λ, let $C_\varepsilon(\lambda)$ be a set of circuits with respect to which ε is correct. We say that the scheme ε is bootstrappable if $D_\varepsilon(\lambda) \subseteq C_\varepsilon(\lambda)$ holds for every λ.

Definition 2.5 AGCD problem over the integers
Given a list of approximate multiples of p : $\{x_i = q_i p + r_i : q_i \in z^+, r_i \in z\}_{i=0}^{\tau}$, find p.

3. Somewhat Homomorphic Encryption Scheme

In this section, we first recall the SWHE in the DGHV scheme over the integers [3], the scheme is based on the AGCD problem. Then, we give two improved SWHE schemes over the integers with shorter public keys. Finally, we prove that our schemes are semantically secure based on the AGCD problem.

The Scheme Parameters. Given the security parameter λ, we use the following parameters [3]:

- ρ is the bit-length of the noise,
- η is the bit-length of the secret key.
- γ is the bit-length of the integers in the public key.
- τ is the number of integers in the public key.
- ρ' is a secondary noise parameter used for encryption.

These parameters must satisfy the following constraints [3]**:**

- $\rho = \omega(\log \lambda)$, to protect against brute-force attacks on the noise,
- $\eta \geq \rho \cdot O(\lambda \log^2 \lambda)$, in order to support homomorphic operations for evaluating the squashed decryption circuit,
- $\gamma = \omega(\eta^2 \log \lambda)$, to thwart various lattice-based attacks on the underlying AGCD problem,
- $\tau \geq \gamma + \omega(\log \lambda)$, in order to use the leftover hash lemma in the reduction to AGCD problem,
- $\rho' = \rho + \omega(\log \lambda)$ for the secondary noise parameter.

For a specific (μ -bit) odd positive integer p, we use the following distribution over γ -bit integers:

$$D_{\gamma,\rho}(p) = \{choose\ \ q \leftarrow Z \cap [0, 2^{\gamma} / p), r \leftarrow Z \cap [-2^{\rho}, 2^{\rho}), Output\ \ x = pq + r\}$$

3.1. SWHE Scheme in the DGHV Scheme over the Integers

The SWHE scheme is composed of four algorithms in [3]: $KeyGen(\lambda)$, $Encrypt(pk, m)$, $Decrypt(sk, c)$ and $Evaluate(pk, C, (c_1, c_2, \cdots, c_t))$.

KeyGen(λ). Generate a random odd integer p with size η bits. For $0 \leq i \leq \tau$, sample $x_i \leftarrow D_{\gamma,\rho}(p)$. Retable so that x_0 is the largest. Restart unless x_0 is odd and $[x_0]_p$ is even. The public key is $pk = \{x_0, x_1, x_2, \cdots x_{\tau}\}$ and the secret key is $sk = p$.

Encrypt $(pk, m \in \{0,1\})$: Choose a random subset $S \subseteq \{1, 2, \cdots, \tau\}$ and a random integer r in $(-2^{\rho'}, 2^{\rho'})$, output $c = [m + 2r + 2\sum_{i \in S} x_i]_{x_0}$.

Evaluate $(pk, C, (c_1, c_2, \cdots, c_t))$. Given the (binary) circuit C with t inputs, and t ciphertexts c_i, apply the addition and multiplication gates of C to the ciphertexts, perform all the operations over the integers, and return the resulting integer.

Decrypt(sk, c). Output $m' = (c \bmod p) \bmod 2$. Note that since p is odd, we can replace the decryption with the formula $m' = (c \bmod 2) \oplus \lfloor c / p \rceil \bmod 2$.

To satisfy these constraints the following parameter set is suggested in [4]: $\rho = \lambda, \rho' = 2\lambda, \quad \eta = O(\lambda^2), \gamma = O(\lambda^5), \tau = \gamma + \lambda = O(\lambda^5)$. The public key size is $O(\lambda^{10})$. In practice, the size of the value x_i should be at least $\gamma = 2^{23}$ bits to prevent lattice attacks. Then, the public key size is at least 2^{46} bits, which is too large for any practical system.

Note that the scheme is a SWHE scheme and it is semantically secure under the AGCD assumption [3].

We recall that the noise of ciphertexts c is equal to $c \bmod p$. The decryption function outputs the correct plaintext if the ciphertext has noise at most $p/2$. However, the ciphertext outputted by a permitted circuit in $Evaluate(pk, C, (c_1, c_2, \cdots, c_t))$ algorithm has noise at most $2^{\eta-4} < p/8$, so the bound $2^{\eta-4} < p/2$ would suffice for correct decryption. Suppose the circuit C is represented by a multivariate polynomial $f(x_1, x_2, \cdots, x_t)$, let d be its degree, then the degree d of the evaluate polynomial f must satisfy the following Equation (1), where $\left|\vec{f}\right|$ is the l_1 norm of the coefficient vector of f which is permitted polynomials (permitted circuits).

$$d \le \frac{\eta - 4 - \log\left|\vec{f}\right|}{\rho' + 2} \tag{1}$$

In fact, too many multiplications or additions operated on ciphertexts make the noise greatly grow, which makes the decryption function no longer output the correct plaintext. The ciphertexts also blow up in size. This SWHE scheme is not fully homomorphic, but Dijk, *et al.*, use Gentry's bootstrapping techniques to turn it into a FHE scheme in [3].

3.2. Our SWHE Schemes over the Integers

In this section, we precisely describe four algorithms ($KeyGen$, $Encrypt$, $Decrypt$, $Evaluate$) in each our SWHE scheme with shorter public keys. Our two schemes are denoted by ε^* and ε^{**} respectively. Our schemes can reduce the public key size of the SWHE schemes from $O(\lambda^{10})$ down to $O(\lambda^5 \log \lambda)$ and $O(\lambda^5)$ respectively. At the same time, x_0 does not effect the correct decryption for ciphertext as long as x_0 is odd and $[x_0]_p$ is even, thus we let x_0 to be initialized by an assignment in the public keys generation process, which can reduce the computation cost.

3.2.1. Two SWHE Schemes

3.2.1.1. Scheme One

KeyGen(λ)

(1) Generate a random odd integer p of size η bits. Let x_0 be odd and $x_0 = pq + 2r$, where q is a random integer of size λ^3 bits and r is a random integer in $[-2^{\rho}, 2^{\rho})$.

(2) Generate integers x_i for $1 \le i \le \log \tau$, let $x_i = pq_i + r_i$, where q_i is a random integer in $[0, 2^{\gamma - \log(5\log\lambda)}/p)$ and r_i is a random integer in $[-2^{\rho - \log(5\log\lambda)}, 2^{\rho - \log(5\log\lambda)})$.

(3) The public key is $pk = \{x_0, x_1, x_2, \cdots x_{\log\tau}\}$ and the secret key is $sk = p$.

Encrypt $(pk, m \in \{0,1\})$. Choose a random subset $S \subseteq \{1,2,3,\cdots,\tau\}$ and a random integer r in $(-2^{\rho'}, 2^{\rho'})$, and output $c = [m + 2r + 2\sum_{i \in S} y_i]_{x_0}$, where y_i is the sum of all elements of each combination in τ different combinations, the τ different combinations are generated from the $\log \tau$ different numbers $\{x_1, x_2, \cdots x_{\log \tau}\}$.

Evaluate and **Decrypt** are the same as in the original scheme ε.

3.2.1.2. Scheme Two

KeyGen (λ)

(1) Generate a random odd integer p of size η bits. Let $x = pq + r$ where q is a random integer in $[2^{\rho}, 2^{\gamma} / p)$ and r is a random integer in $[-2^{\rho-1}, 2^{\rho-1})$. Let x_0 is odd and $x_0 = pq + 2r$, where q is a random integer of size λ^3 bits and r is a random integer in $[-2^{\rho}, 2^{\rho})$.

(2) Generate integers x_i for $1 \le i \le \log \tau$, let $x_i = pq_i + r_i$, where q_i is a random integer in $[0, 2^{\rho - \log(5\log\lambda)})$ and r_i is a random integer in $[-2^{\rho-1-\log(5\log\lambda)}, 2^{\rho-1-\log(5\log\lambda)})$.

(3) The public key is $pk = \{x_0, x, x_1, x_2, \cdots x_{\log \tau}\}$ and the secret key is $sk = p$.

Encrypt $(pk, m \in \{0,1\})$. Choose a random subset $S \subseteq \{1,2,3,\cdots,\tau\}$ and a random integer r in $(-2^{\rho'}, 2^{\rho'})$, and output $c = [m + 2r + 2\sum_{i \in S} (x - y_i)]_{x_0}$ where y_i is the sum of all elements of each combination in τ different combinations, the τ different combinations are generated from the $\log \tau$ different numbers $\{x_1, x_2, \cdots x_{\log \tau}\}$.

Evaluate and **Decrypt** are the same as in the original scheme ε.

To satisfy these constraint conditions of the parameters, we can still take $\rho = \lambda$, $\eta = O(\lambda^2)$, $\rho' = 2\lambda$, $\gamma = O(\lambda^5)$ and $\tau = O(\lambda^5)$ as in the original scheme [3]. In fact, y_i in the scheme ε^* and $x - y_i$ in the scheme ε^{**} are equivalent to the public key x_i in the scheme ε. Hence the public key size becomes $O(\lambda^5 \log \lambda)$ in the scheme one and $O(\lambda^5)$ in the scheme two instead of $O(\lambda^{10})$ in the DGHV scheme.

3.2.2. Correctness

Lemma 3.1 Two schemes are correct.

Proof. The correctness of the scheme one is obvious. We only prove the scheme two. Given ciphertext c and the secret key p, it is easy to verify that $\sum_{i \in S} y_i$ has general form $\sum_{i \in S} y_i = pq_1 + r_1$. To decrypt c, one computes:

$$
\begin{aligned}
&(c \bmod p) \bmod 2 \\
&= [[m + 2r + 2\sum_{i \in S} (x - y_i)]_{x_0}]_p \bmod 2 \\
&= [m + 2r + 2|S|x - 2\sum_{i \in S} y_i - kx_0]_p \bmod 2 \\
&= [m + 2r + 2|S|(pq_2 + r_2) - 2(pq_1 + r_1) - k(pq_3 + 2r_3)]_p \bmod 2 \\
&= [m + 2(r + |S|r_2 - r_1 - kr_3) + p(2|S|q_2 - 2q_1 - kq_3)]_p \bmod 2 \\
&= m
\end{aligned}
\tag{2}
$$

Recall that here $c \bmod p$ is an integer in $(-p/2, p/2]$. It requires the absolute value of the noise that $m + 2(r + |S| r_2 - r_1 - kr_3)$ at most $p/2$, one can perform decryption correctly (make the equation (2) true).

It is clear that two schemes have homomorphic properties for two operations (addition and multiplication). But the noise will grow greatly along with the operations on ciphertext. Eventually the noise is so great that it is not possible to decrypt. The noise is linear increase and quadratic growth with additions and multiplications respectively. Thus, the evaluate effect of our two schemes is mainly the circuit depth (polynomial degree). To decrypt the ciphertext outputted by the circuit C correctly, it requires the degree of the evaluative polynomial satisfies the inequality (1).

3.2.3. Security

As a matter of fact, y_i in the scheme ε^* and $x - y_i$ in the scheme ε^{**} are equivalent to the public key x_i in the scheme ε. So, we can reduce the security of our two schemes to the hardness of the approximate GCD problem as the original scheme ε.

Lemma 3.2 y_i (in the scheme ε^*) and $x - y_i$ (in the scheme ε^{**}) are equivalent to the public key x_i in the scheme ε.

Proof. In the scheme ε^*, y_i is the sum of all elements of each combination in τ different combinations, the τ different combinations are generated from the $\log \tau$ different numbers $\{x_1, x_2, \cdots x_{\log \tau}\}$.

$$x_i = pq_i + r_i \quad 1 \le i \le \log \tau$$

where q_i is a random integer in $[0, 2^{\gamma - \log(5\log\lambda)}/\mathrm{p})$ and r_i is a random integer in $[-2^{\rho - \log(5\log\lambda)}, 2^{\rho - \log(5\log\lambda)})$. We obtain:

$$y_i = pq + r, q \in [0, 2^{\gamma}/p) \text{ and } r \in [-2^{\rho}, 2^{\rho}).$$

Therefore, y_i is consistent with the public key $x_i = pq + r$, where $q \in [0, 2^{\gamma}/p)$ and $r \in [-2^{\rho}, 2^{\rho})$) in the scheme ε.

In the scheme ε^{**}:

$$x - y_i = p(q_1 - q_2) + (r_1 - r_2),$$

where $q_1 \in Z \cap [2^{\rho}, 2^{\gamma}/p)$, $r_1 \in Z \cap [-2^{\rho-1}, 2^{\rho-1})$, $q_2 \in Z \cap [0, 2^{\rho})$, and $r_2 \in Z \cap [-2^{\rho-1}, 2^{\rho-1})$, we obtain:

$$(q_1 - q_2) \in [0, 2^{\gamma}/p) \text{ and } (r_1 - r_2) \in [-2^{\rho}, 2^{\rho}).$$

Therefore, $x - y_i$ is consistent with the public key $x_i = pq + r$, where $q \in [0, 2^{\gamma}/p)$ and $r \in [-2^{\rho}, 2^{\rho})$ in the scheme ε.

The two improved schemes are semantically secure and we do not change the security of the original scheme ε. Similarly, we can reduce any attack problem for our schemes to the difficulty of solving AGCD problem, the proof is the same strategy as [3].

We now compare our improved schemes with two existing schemes [3, 4] in the following Table 1.

Table 1. The Comparison of Four Schemes

Schemes	Security level	Based on the difficult problem	Public key sizes	Secret key sizes
[3]	IND-CPA	AGCD	$O(\lambda^{10})$	$O(\lambda^{2})$
[4]	IND-CPA	PACD(no security)	$O(\lambda^{7})$	$O(\lambda^{2})$
Our scheme one	IND-CPA	AGCD	$O(\lambda^{5}\log\lambda)$	$O(\lambda^{2})$
Our scheme two	IND-CPA	AGCD	$O(\lambda^{5})$	$O(\lambda^{2})$

4. Making the Scheme Fully Homomorphic

We can make the two improved SWHE schemes fully homomorphic with Gentry's approach [3]. Subsequently, we set our improved scheme two as an example to construct fully homomorphic encryption scheme from the SWHE scheme ε^{**} (our improved scheme two) that is bootstrappable.

According inequality (1) and the settings of parameters in §3.1, the degree of the evaluative polynomial (permitted circuits) is at most $\alpha\lambda\log^{2}\lambda$. However, our decryption function is more complex, due to $\lfloor c/p \rceil$ has high computational complexity so that the decryption function does not belong to the permitted function of the scheme ε^{**}. We use Gentry's approach [14] to reduce the degree of polynomials required on operations by compressing the decryption function (circuit), and introducing another hardness assumption, namely the sparse subset sum problem (SSSP), which can make the SWHE scheme ε^{**} bootstrapped. Then we obtain a FHE scheme by bootstrap transformations which can handle Boolean circuits of any depth.

4.1. Compressing the Decryption Circuit

We follow the description of [3]. Three more parameters κ, Θ and θ, which are functions of λ. Concretely, one uses $\kappa = \gamma\eta/\rho', \Theta = \omega(\kappa\log\lambda)$ and $\theta = \lambda$.

KeyGen. Generate $sk^{*} = p$ and pk^{*} as before. Set $x_p = \lfloor 2^{\kappa}/p \rceil$, choose at random a Θ-bit vector with hamming weight θ, $\mathbf{s} = <s_1, s_2, \cdots, s_\Theta>$, and let $S = \{i : s_i = 1\}$. Choose at random integers $u_i \in [0, 2^{\kappa+1})$ for $i = 1, 2, \cdots, \Theta$, subject to the condition that $\sum_{i\in S} u_i = x_p \pmod{2^{\kappa+1}}$. Set $\boldsymbol{y} = <y_1, y_2, \cdots, y_\Theta>$ and $y_i = u_i/2^{\kappa}$. Hence each y_i is a positive number smaller than two, with κ bits of precision after the binary point. Also, $[\sum_{i\in S} y_i]_2 = 1/p - \Delta_p$ for some $|\Delta_p| < 2^{-\kappa}$. Output the secret key $sk = \boldsymbol{s}$ and public key $pk = (pk^{*}, \boldsymbol{y})$.

Encrypt and **Evaluate**. Generate a ciphertext c^{*} as before. Then for $i \in \{1, 2, \cdots, \Theta\}$, set $z_i = c^{*} y_i \bmod 2$, keeping only $n = \lceil \log\theta \rceil + 3$ bits of precision after the binary point for z_i. Output $(c^{*}, <z_1, z_2, \cdots, z_\Theta>)$.

Decrypt. Output $m' = (c^{*} - \lfloor \sum_{i\in S} s_i z_i \rceil)$ mod2.

4.2. Bootstrapping

The complexity of the decryption circuit is still large. In order to further reduce the complexity of the decryption circuit, we perform three steps as below to implement the decryption algorithm as [3].

Step1. For $i \in \{1, 2, \cdots, \Theta\}$, set $a_i = s_i z_i$. The a_i's are still rational numbers in [0, 2), given in binary representation with $n = \lceil \log \theta \rceil + 3$ bits of precision after the binary point.

Step2. From the Θ rational numbers $\{a_i\}_{i=1}^{\Theta}$, generate other $n+1$ rational numbers $\{w_j\}_{j=0}^{n}$ with less than n bits of precision, such that $\sum_j w_j = \sum_i a_i \bmod 2$.

Step 3. Output $(c^* - \lfloor \sum_j w_j \rceil)$ mod2

If we want to implement three steps above, it requires that the degree of the polynomials in the first step is two, the degree of polynomials in the second step is at most θ, and the degree of the polynomial in the third step is at most $32(n+1)^{1/\log(3/2)} < 32(\lceil \log \theta \rceil + 4)^{1.71} < 32 \log^2 \theta$. Therefore, the total degree of the decryption circuit is bounded by $2 \cdot \theta \cdot 32 \log^2 \theta = 64\theta \log^2 \theta$, and since we are using $\theta = \lambda$ and the degree is at most $64\lambda \log^2 \lambda$.

It follows that the augmented decryption circuits $D_{\varepsilon^{**}}(\lambda)$ (Definition 2.3) can be expressed as polynomials of degree at most $128\lambda \log^2 \lambda$. In order to satisfy the inequality (1), setting $\eta = \rho' \cdot 128\lambda \log^2 \lambda$, we can get $D_{\varepsilon}(\lambda) \subseteq C_{\varepsilon}(\lambda)$, thus making the scheme bootstrapped. Then we obtain a FHE scheme by using bootstrap transformations which can handle Boolean circuits of any depth.

5. Conclusion

In this paper, we described our two somewhat homomorphic encryption schemes with shorter public keys on the basis of the DGHV scheme, and then invoked Gentry's bootstrapping technique to obtain fully homomorphic public key encryption schemes. Our schemes were still semantically secure under the AGCD problem and the SSSP [15] assumption. Compared with the Dijk, *et al.,* scheme [3] and Coron, *et al.*, scheme [4], our schemes were more efficient and had smaller public key size in $O(\lambda^5 \log \lambda)$ and $O(\lambda^5)$ respectively.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grant No. 11271003, the National Research Foundation for the Doctoral Programme of Higher Education of China under Grant No. 20134410110003, High Level Talents Project of Guangdong, Guangdong Provincial Natural Science Foundation under Grant No. S2012010009950, the Project of Department of Education of Guangdong Province under Grant No 2013KJCX0146, and the Natural Science Foundation of Bureau of Education of Guangzhou under Grant No. 2012A004

References

[1] R. L. Rivest, L. Adleman and M. L. Dertouzos, "On Data Banks and Privacy Homomorphism", Foundations of Secure Computation, Academic Press, New York, **(1978)**, pp. 452-473.

[2] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattice", Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, **(2009)** May 31 - June 2, Bethesda, USA.
[3] M. V. Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, "Fully homomorphic encryption over the integers", Proceedings of the ASIACRYPT'2010, **(2010)** December 5-9, Clarke Quay, Singapore.
[4] J.-S. Coron, A. Mandal, D. Naccache and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys", Proceedings of the Crypto 2011, **(2011)** August 14–18, Santa Barbara, USA.
[5] Y. Chen and P. Q. Nguyen, "Faster algorithms for approximate common divisors: Breaking fullyhomomorphic-encryption challenges over the integers", Cryptology ePrint Archive, Report 2011/436, **(2011)**, http://eprint.iacr.org/2011/436.
[6] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits", Proceedings of the FOCS 2011, **(2011)** October 23-25, Palm Springs, USA.
[7] C. A. Melchor, G. Castagnos and P. Gaborit, "Lattice-based Homomorphic Encryption of Vector Spaces", In proceeding of: Information Theory, 2008. ISIT 2008. IEEE International Symposium on, **(2008)** July 6-11, Toronto, Canada.
[8] J. S. Coron, D. Naccache and M. Tibouchi, "Public-key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers", Proceedings of the Eurocrypto 2012, **(2012)** April 15-19, Cambridge, United Kingdom.
[9] Y. Chen and P. Q. Nguyen, "Faster Algorithms for Approximate Common Divisors: Breaking Fully Homomorphic Encryption Challenges over the Integers", Proceedings of the Eurocrypto 2012, **(2012)** April 15-19, Cambridge, United Kingdom.
[10] K. M. Chung, Y. T. Kalai and S. Vadhan, "Improved Delegation of Computation Using Fully Homomorphic Encryption", Proceedings of the 30th Annual Conference on Advances in Cryptology, **(2010)**, Berlin, Germany.
[11] P. Scholl and N. P. Smart, "Improved Key Generation for Gentry's Fully Homomorphic Encryption Scheme", http://eprint.iacr.org/2011/471.
[12] J. Kim, M. S. Lee, A. Yun and J. H. Cheon, "CRT-based fully homomorphic encryption over the integers", Cryptology Print Archive, Report 2013/057, **(2013)**, http://eprint.iacr.org.
[13] J. S. Coron, T. Lepoint, M. Tibouchi, *et al.*, "Batch fully homomorphic encryption over the integers", In EUROCRYPT 2013, **(2013)** May 26-30, Athens, Greece.
[14] C. Gentry, "A fully homomorphic encryption scheme [D/OL]", Stanford University, **(2009),** http: // crypto.stanford.edu/craig.
[15] P. Q. Nguyen and J. Stern, "Adapting density attacks to low weight knapsacks", Proceedings of Asiacrypt'05, **(2005)** December 4-8, Madras, India.

Authors

Can Xiang, she is a PhD candidate in Guangzhou University of China. She obtained her Master's degree in Hainan Normal University of China. Her research interests include cryptography, information security and cloud computing.

Chunming Tang, he was born in 1972. He received the Ph. D. degree in Academy of Mathematics and Systems Science from Chinese Academy of Sciences, Beijing city, in 2004. Currently, he is a Professor at Guangzhou University. His research interests include Cryptography, Information Security, and Cloud Computing.

