Fully Homomorphic Encryption Based on Approximate Matrix GCD

Gu Chunsheng
School of Computer Engineering
Jiangsu Teachers University of Technology
Changzhou, China, 213001
guchunsheng@gmail.com
November 27, 2011

Abstract: We first introduce approximate matrix GCD problem (AMGCD), and construct public key encryption schemes based on AMGCD. Then, we define a variant of AMGCD and design a new fully homomorphic encryption scheme (FHE) based on the variant AMGCD, whose security depends on the hardness assumption of the variant AMGCD problem.
Keywords: Fully Homomorphic Encryption, Approximate Matrix GCD, Learning with Error, Approximate GCD

1. Introduction

We construct a new fully homomorphic encryption schemes, which are based on the trapdoor function of an approximate matrix GCD over the integers. Let n a security parameter, $p=\prod_{i=1}^{n} p_i$ with $p_i, i=1,...,n$ positive integers, $A\in\mathbb{Z}_p^{n\times n}$ where $A=S\times diag(p/p_1, p/p_2,..., p/p_n)\times S^{-1}$ such that $S\times S^{-1}=I\in\mathbb{Z}_p^{n\times n}$. The public key is a list matrices $pk=\{B_i=(A\times R_i+2E_i\}_{i=0}^{\tau}$, where $R_i\in\mathbb{Z}_p^{n\times n}$ is an uniformly random matrix and $S\times R_i\times S^{-1}$ is a non-diagonalizable matrix, and $E_i=S\times diag(e_1,e_2,...,e_n)\times S^{-1}\in\mathbb{Z}_p^{n\times n}$ with $|e_i|=2^{O(n)}$. The secret key is $sk=(p,A,S)$. Assume $m\in\{0,1\}$ an integer, its ciphertext is evaluated as $C=\sum_{i=0}^{\tau} k_i\bullet B_i+m\bullet I$, where $k_i\in\{0,1\}$ and I is the identity matrix. To compute addition/multiplication of the integers in ciphertexts, the scheme simply adds/multiplies the ciphertexts as the addition/multiplication over $\mathbb{Z}^{n\times n}$. To decrypt a ciphertext C, one computes $P=\left[\left[S^{-1}\times(T\times C \bmod p)\times S\right]_p\right]_2$, where $T=p\bullet A^{-1}$. Now one can compute the

plaintext $m = P(1,1)$ from the first row and the first column $P(1,1)$ of P. Recall that here $[z]_p$ is an integer in $(-p/2, p/2)$.

Notice that here p is a part of the secret key for our scheme, and is not included in the public key.

1.1 Our Contribution

Our first contribution is to design new trapdoor function based on (approximate) matrix GCD problem (MGCD). We think that this new trapdoor is independent of interest.
Our second contribution is to construct a new fully homomorphic encryption, whose security relies upon the hardness of the variant approximate MGCD problem.

1.2 Related work

Rivest, Adleman, and Dertouzos [RAD78] first investigated a privacy homomorphism, which now is called the fully homomorphic encryption (FHE). After then, many researchers [BGN05, ACG08, SYY99, Yao82] have worked at this open problem. Until 2009, Gentry [Gen09] constructed the first fully homomorphic encryption using ideal lattice. In Gentry's scheme, the public key is approximately n^7 bits, the computation per gate costs $O(n^6)$ operations. Smart and Vercauteren [SV10] presented a fully homomorphic encryption scheme with both relatively small key $O(n^3)$ bits , ciphertext size $O(n^{1.5})$ bits and computation per gate at least $O(n^3)$ operations, which is in some sense a specialization and optimization of Gentry's scheme. Dijk, Gentry, Halevi, and Vaikuntanathan [vDGHV10] proposed a simple fully homomorphic encryption scheme over the integers, whose security depends on the hardness of finding an approximate integer GCD. Stehle and Steinfeld [SS10] improved Gentry's fully homomorphic scheme and obtained to a faster fully homomorphic scheme, with $O(n^{3.5})$ bits complexity per elementary binary addition/multiplication gate, but the hardness assumption of the security of the scheme in [SS10] is stronger than that in [Gen09]. Brakerski and Vaikuntanathan[BV11a,BV11b] respectively constructed Ring-LWE/LWE-based fully homomorphic encryptions. Gentry and Halevi [GH11b] designed a new fully homomorphic encryption by replacing the hardness assumption of SSSP with the hardness assumption of Diffie-Hellman.
Recently, Brakerski and Gentry and Vaikuntanathan [BGV11] presented a radically new approach to leveled fully homomorphic encryption (FHE) without Gentry's bootstrapping procedure.

1.3 Outline

Section 2 recalls the notations and the definitions of (approximate) MGCD problem. Section 3 designs new trapdoor functions and describe how to construct public key scheme based on new trapdoor functions. Section 4 gives a new fully homomorphic encryption scheme. Section 5 gives its security assumption. Section 6 concludes this paper and gives some open problems.

2. Preliminaries

2.1 Notations

Let n be a security parameter, and $[n]$ a set of integers $\{0,1,...,n\}$. Let $p=\prod_{i=1}^{n} p_i$ with p_i primes. Let $\mathbb{Z}$ be integer set, $\mathbb{Z}_p=\mathbb{Z}/p\mathbb{Z}$. We refer to $A\in\mathbb{Z}^{n\times n}$ as a $n\times n$ matrix, A^t the transpose matrix of A, and A^{-1} the inverse matrix of A. We will refer to I as the identity matrix over $\mathbb{Z}^{n\times n}$, M as the "1" matrix.

Given $u\in\mathbb{Z}^n$, we refer to $rot(u)=(-u_{n-1},u_0,...,u_{n-2})^T$ as the cyclic rotation of u, and $Rot(u)=(u,rot(u),...,rot^{n-1}(u))^T$ as the circulant matrix of u.

2.2 Matrix GCD and Approximate MGCD

The lattice problem is a natural generalization of the greatest common divisor of two integers. Given integers m,n, one can compute the first minimum of the lattice $m\mathbb{Z}+n\mathbb{Z}$ spanned by m and n by using the classical Euclidean algorithm. However, there is no known efficient algorithm that finds the shortest vector for the general lattice problem. In this paper, we will generalize the GCD problem to matrix GCD problem.

Definition 2.1. (Lattice): Given n linearly independent vectors $b_1,b_2,...,b_m\in\mathbb{R}^n$, the lattice is equal to the set $L(b_1,b_2,...,b_m)=\{\sum_{i=1}^{m} x_i\bullet b_i, x_i\in\mathbb{Z}\}$ of all integer linear combinations of the b_i's. We also denote by matrix B the b_i's. In this paper, we only consider the lattice over the integers, i.e., $b_i\in\mathbb{Z}^n$.

Currently, the computational hard problem over the general lattice is mainly the shortest vector problem and the closest vector problem.

Definition 2.2 ([SV10, GH11] Small Principal Ideal Problem (SPIP)). Given a principal

ideal π in either two elements (p,α) or HNF representation, compute a small generator of the ideal.

Definition 2.3. (Learning With Error (LWE) [Reg09]). Let n, p be integers related to security parameter λ, and χ a distribution over $\mathbb{Z}_p$. Given a list samples (b_i, r_i) of the distribution $D_{n,p,\chi}$ over $\mathbb{Z}_p^{n+1}$ such that $a \leftarrow \mathbb{Z}_p^n$, $r_i \leftarrow \mathbb{Z}_p^n$, $e_i \leftarrow \chi$ and $b_i = < a, r_i > + e_i \bmod p$, the LWE problem $LWE_{n,p,\chi}$ is to distinguish the distribution $D_{n,p,\chi}$ from the uniform distribution over $\mathbb{Z}_p^{n+1}$.

Definition 2.4. ([vDGHV10] Approximate-GCD over the Integers (AGCD)). Given a list of approximate multiples of p: $\{b_i = a_i p + e_i : a_i \in Z_+, e_i \in Z, |e_i| < 2^{n-1}\}_{i=0}^{\tau}$, find p.

Definition 2.5. (Matrix GCD (MGCD)): Given matrices $B_i = AR_i \in \mathbb{Z}^{n\times n}, i = 1,2$, find the matrix $A \in \mathbb{Z}^{n\times n}$ such that $A \mid B_i$ and $\det(A) = \gcd(\det(B_1), \det(B_2))$.

Definition 2.6. (Approximate MGCD over the Integers): Given a list matrices $B_i = AR_i + E_i \in \mathbb{Z}^{n\times n}$, where $R_i, E_i \in \mathbb{Z}^{n\times n}$ with $\|E_i\|_\infty \le \beta$, find the matrix $A \in \mathbb{Z}^{n\times n}$ such that $\left\| B_i - A\lceil A^{-1}B_i + 0.5 \bullet M \rceil \right\|_\infty \le \beta$ with M all entries one.

Definition 2.7. (Approximate MGCD): Given a list matrices $B_i = AR_i + E_i \in \mathbb{Z}_p^{n\times n}$, where $R_i, E_i \in \mathbb{Z}^{n\times n}$ and $\det(A) = p$, find the matrix $A \in \mathbb{Z}^{n\times n}$ such that $\left\| p \bullet A^{-1}E_i \right\|_\infty < p/2$.

Since there is unimodular matrix over $\mathbb{Z}^{n\times n}$, one can not get an unique solution for the MGCD and approximate MGCD.

3. Trapdoor Functions Based on Approximate MGCD

In this section, we present two new trapdoor functions and describe how to construct public key scheme based on the approximate MGCD problem. In the following, we first give Lemma 3.1.

Lemma 3.1. Given an arbitrary matrix $T \in \mathbb{Z}^{n\times m}$ with $n \le m$, there is a polynomial-time algorithm that outputs a matrix $A \in \mathbb{Z}^{n\times m}$ such that $AT^t = p \bullet I$ with $p = \det(T^t T)$.

Proof: Let. We set $A = p \bullet (T \times T^t)^{-1} \times T$ by using the definition of dual lattice. It is easy to verify $AT^t = p \bullet I$.

To describe simplicity, we take $n = m$, $p = \det(T)$, $A = p \bullet T^{-1}$ throughout this paper. According to Lemma 3.1, we present two trapdoor functions as follows.

Trapdoor Function 1: Assume $T \in \mathbb{Z}^{n \times n}$ and $A = p \bullet T^{-1}$ with p odd integer. Given a list matrices $B_i = (AR_i + 2 \bullet E_i) \bmod p, i = 1, 2, ..., \tau = O(n)$ with $R_i, E_i \in \mathbb{Z}^{n \times n}$ such that $\left\| 2 \bullet TE_i \right\|_\infty < p/2$, find matrix A.

Trapdoor Function 2: Assume $T \in \mathbb{Z}^{n \times n}$ and $A = p \bullet T^{-1}$ with p odd integer. Given a list matrices $B_i = (AR_i + 2 \bullet E_i), i = 1, 2, ..., \tau = O(n)$ with $R_i, E_i \in \mathbb{Z}^{n \times n}$ such that $\left\| 2 \bullet TE_i \right\|_\infty < p/2$, find matrix A.

Public Key Encryption Scheme Based on the Trapdoor Function 1 (PKE-1). Assume that the public key is $B_i = (AR_i + 2E_i) \bmod p$ with $R_i, E_i \in \mathbb{Z}^{n \times n}$ such that $\left\| 2 \bullet TE_i \right\|_\infty < p/(2n\tau\beta)$, the secret key T. The plaintext matrix is $X \in \mathbb{Z}_2^{n \times n}$. One encrypts plaintext X as $C = (\sum_{i=1}^{\tau} K_i \times B_i + X) \bmod p$, where $K_i \in \mathbb{Z}^{n \times n}$ with $\left\| K_i \right\|_\infty \le \beta$. One decrypts ciphertext C as follows.

$$\left[\left[C \times T \right]_p \right]_2 \times \left[T \right]_2^{-1} = \left[\sum_{i=1}^{\tau} 2 \bullet K_i E_i T + XT \right]_2 \times \left[T \right]_2^{-1} = \left[XT \right]_2 \times \left[T \right]_2^{-1} = X .$$

Here $\left[T \right]_2^{-1}$ denotes the inverse matrix of T modulo 2. It is not difficult to verify the correctness of decryption algorithm.

Public Key Encryption Scheme Based on the Trapdoor Function 2 (PKE-2). Assume that the public key is $B_i = AR_i + 2E_i$ with $R_i, E_i \in \mathbb{Z}^{n \times n}$ such that $\left\| 2 \bullet TE_i \right\|_\infty < p/(2n\tau\beta)$, the secret key T. The plaintext matrix is $X \in \mathbb{Z}_2^{n \times n}$. One encrypts plaintext X as $C = \sum_{i=1}^{\tau} K_i \times B_i + X$, where $K_i \in \mathbb{Z}^{n \times n}$ with $\left\| K_i \right\|_\infty \le \beta$. One decrypts ciphertext C same as that of PKE-1.

4. Fully Homomorphic Encryption Scheme

To design fully homomorphic encryption, we cannot directly apply trapdoor function 2 in Section 3. We first need to give the following variant of approximate MGCD problem and its trapdoor function. Then, we use standard bootstrappable techniqure to implement a new fully homomorphic encryption scheme.

4.1 Variant of Trapdoor Function 2

For the above trapdoor function, the addition between ciphertexts is obvious. To support multiplication operation between ciphertexts, the matrices E and A need to commute, that is, $E \times A = A \times E$. According to 1.3.12 in [HG05], E and A commute if and only if they are simultaneously diagonalizable. Thus, we construct the variant of the trapdoor function 2 as follows.

Trapdoor Function 2': Assume $p = \prod_{i=1}^{n} p_i$ with p_i positive integers, $A = S \times diag(p_1, p_2, ..., p_n) \times S^{-1}$, where $S \in Z_p^{n \times n}$ is an random matrix such that $S \times S^{-1} = I \in \mathbb{Z}_p^{n \times n}$. Given a list matrices $\{B_i = A \times R_i + E_i\}_{i=0}^{\tau}$, where $R_i \in \mathbb{Z}_p^{n \times n}$ is an uniformly random matrix and $S \times R_i \times S^{-1}$ is a non-diagonalizable matrix, $E_i = S \times diag(e_1, e_2, ..., e_n) \times S^{-1} \in \mathbb{Z}^{n \times n}$, find the matrix $A \in \mathbb{Z}^{n \times n}$.

4.2 Fully Homomorphic Encryption Scheme

In this subsection, we first present a construct a fully homomorphic encryption scheme, then show its correctness and simply analyze its performance.

4.2.1 Construction

Key Generating Algorithm (KeyGen):

(1) According to Trapdoor Function 2', select a list random primes $p_i, i = 1, 2, ..., n$. Compute $p = \prod_{i=1}^{n} p_i$, $A = S \times diag(p / p_1, p / p_2, ..., p / p_n) \times S^{-1}$, where $S \in Z_p^{n \times n}$ is an random matrix such that $S \times S^{-1} = I \in \mathbb{Z}_p^{n \times n}$.

(2) Generate $k = O(\log p / n)$ groups, the *j*-th includes n^2 matrices $\{V_i = (A \times R_i + 2E_i) \bmod (p \bullet 2^{ni})\}_{i=1}^{n^2}$, where $R_i \in \mathbb{Z}_p^{n \times n}$ is an uniformly random matrix and $S \times R_i \times S^{-1}$ is a non-diagonalizable matrix, and $E_i = S \times diag(e_1, e_2, ..., e_n) \times S^{-1} \in \mathbb{Z}^{n \times n}$ such that $|e_i| = 2^{O(n)}$. The *j*-th group $\{V_i = A \times R_i + 2E_i\}_{i=1}^{n^2}$ consists of a vector basis V^j, Here each V_i is an n^2-tuple column vector.

(3) Generate a list matrices $\{B_i = (A \times R_i + 2 \bullet E_i) \bmod p\}_{i=0}^{\tau}$, where $R_i \in \mathbb{Z}_p^{n \times n}$ is an uniformly random matrix and $S \times R_i \times S^{-1}$ is a non-diagonalizable matrix, and $E_i = S \times diag(e_1, e_2, ..., e_n) \times S^{-1} \in \mathbb{Z}^{n \times n}$ such that $|e_i| = 2^{O(n)}$.

(4) Output the public key $pk = (V^j, j \in [k], B_i, i \in [\tau])$ and the secret key $sk = (p, A, S)$.

Encryption Algorithm (Enc). Given the public key pk and an integer $m \in \{0,1\}$. Evaluate ciphertext $C = (\sum_{i \in [\tau]} k_i \bullet B_i + m \bullet I) \bmod V^0$ where $k_i \in \{0,1\}$.

In this paper, when computing $\bmod V$, B_i is viewed as an n^2-tuple column vector.

Add Operation (Add). Given the public key pk and ciphertexts C_1, C_2, output new ciphertext $C = (C_1 + C_2) \bmod V^0$.

Multiplication Operation (Mul). Given the public key pk and ciphertexts C_1, C_2, output new ciphertext $C = (C_1 \times C_2) \bmod V^k \bmod V^{k-1} ... \bmod V^0$.

Decryption Algorithm (Dec). Given the secret key sk and ciphertext C, decipher $P = \left[\left[S^{-1} \times (T \times C \bmod p) \times S \right]_p \right]_2$ with $T = p \bullet A^{-1}$, and obtain the plaintext $m = P(1,1)$.

4.2.2 Correctness

Lemma 4.1. The above **Dec** algorithm is correct.

Proof. Given ciphertext C and the secret key sk, it is not difficult to verify that C has general form $C = A \times R + 2 \bullet E + m \bullet I$. To decrypt C, one evaluates

$$
\begin{aligned}
P &= \left[\left[S^{-1} \times (T \times C \bmod p) \times S \right]_p \right]_2 \\
&= \left[\left[S^{-1} \times (T \times (A \times R + 2E + m \bullet I) \bmod p) \times S \right]_p \right]_2 \\
&= \left[\left[S^{-1} \times (p \bullet A^{-1} \times (A \times R + 2E + m \bullet I) \bmod p) \times S \right]_p \right]_2 \\
&= \left[\left[S^{-1} \times (p \bullet A^{-1} \times (2E + m \bullet I) \bmod p) \times S \right]_p \right]_2 \\
&= \left[2 \bullet diag(p_1, p_2, ..., p_n) \times diag(e_1, e_2, ..., e_n) + diag(mp_1, mp_2, ..., mp_n) \right]_2 \\
&= \left[diag(mp_1, mp_2, ..., mp_n) \right]_2
\end{aligned}
$$

Notice that we here use $T = p \bullet A^{-1} = S \times diag(p_1, p_2, ..., p_n) \times S^{-1}$. Since p_1 is odd integer, we have $m = P(1,1)$. ■

Lemma 4.2. The **Add** and **Mul** algorithms in the above scheme are correct.

Proof. The correctness of the Add algorithm is obvious. We only prove the Mul algorithm. Before multiplication operation, we know that $C_j = A \times R_j + E_j + m_j \bullet I$, $j = 1,2$. Notice that in the following proof, we only consider the final computing result relative modulo p, although it is not in the public key.

$$\begin{aligned} C &= (A \times R_1 + 2E_1 + m_1 \bullet I) \times (A \times R_2 + 2E_2 + m_2 \bullet I) \bmod p \\ &= ((AR_1(AR_2 + 2E_2 + m_2 \bullet I) + (2E_1 + m_1 \bullet I)AR_2 + (2E_1 + m_1 \bullet I)(2E_2 + m_2 \bullet I) \bmod p \\ &= ((AR_1(AR_2 + 2E_2 + m_2 \bullet I) + A(2E_1 + m_1 \bullet I)R_2 + (2E_1 + m_1 \bullet I)(2E_2 + m_2 \bullet I) \bmod p \\ &= (AR + 2E + m_1 m_2 \bullet I) \bmod p \end{aligned}$$

where $R = R_1(AR_2 + 2E_2 + m_2 \bullet I) + (2E_1 + m_1 \bullet I)R_2$, $E = 4E_1E_2 + 2m_2 \bullet E_1 + 2m_1 \bullet E_2$. ■

Now, we use standard bootstrappable technique to implement fully homomorphic encryption.

4.2.3 Performance

The size of the public key $pk = (V, B_i, i \in [\tau])$ is $O(n^5)$, the size of the secret key $sk = (p, A, S)$ is $O(n^3)$. The expansion rate of ciphertext is $O(n^2)$. The running times of **Enc, Dec, Add, Mul** are respectively $O(n^4)$, $O(n^2 \log n)$, $O(n^2)$, and $O(n^4)$.

4.2.4 Optimization

- According to the above scheme, there exist two matrices (X, Y) for arbitrary ciphertext C such that $(X \times C \times Y) \bmod p = diag(c_1, c_2, ..., c_n)$. To avoid this attack, we hide modulo p in the scheme. However, this increases the size of the public key. So, we may choose two large primes p_1, q_1, and set the public key as $pk = (q = pq_1, B_i, i \in [\tau])$. Now, there does not exist (X, Y) such that $(X \times C \times Y) \bmod q = diag(c_1, c_2, ..., c_n)$ for arbitrary C. Thus, the security of the scheme depends on the hardness of factoring q.
- We may take small matrix dimension (e.g. a constant) and big integer p to decrease the expansion rate of ciphertext.

5. Security Assumption

Since the security of our scheme relies upon the hardness of approximate MGCD problem, which is a new problem. Currently, we do not know its concrete computational hardness. According to the hardness of LWE [Reg09] and the hardness of approximate GCD [vDGHV10], we have some confidence in the inherent hardness about the approximate matrix GCD problem and its variant problem. However, we now can not say anything about their security for our scheme. Further study is certainly a very important research direction. Thus, in the following we will apply the hardness assumption of approximate MGCD to support the security of our scheme.

Assumption 1 (MGCD). There is no polynomial-time algorithm that finds $A \in \mathbb{Z}^{n\times n}$ from a list matrices $B_i = AR_i \in \mathbb{Z}_p^{n\times n}$, where $R_i \in \mathbb{Z}^{n\times n}$ and $\det(A) = p$.

Assumption 2 (AMGCD). There is no polynomial-time algorithm that finds $A \in \mathbb{Z}^{n\times n}$ from a list matrices $B_i = AR_i + E_i \in \mathbb{Z}_p^{n\times n}$, where $R_i, E_i \in \mathbb{Z}^{n\times n}$ and $p \mid \det(A)$.

Assumption 3 (V-AMGCD). There is no polynomial-time algorithm that finds $A \in \mathbb{Z}^{n\times n}$ from a list matrices $\{B_i = (A \times R_i + E_i)\}_{i=0}^{\tau}$, where $R_i \in \mathbb{Z}_p^{n\times n}$ is an uniformly random matrix and $S \times R_i \times S^{-1}$ is a non-diagonalizable matrix, $A = S \times diag(p_1, p_2, ..., p_n) \times S^{-1}$, and $E_i = S \times diag(e_1, e_2, ..., e_n) \times S^{-1} \in \mathbb{Z}_p^{n\times n}$.

To optimize and implement FHE scheme, Smart and Vercauteren [GH10] and Gentry and Halevi [GH11a] constructed FHE based on the principal ideal lattice problem. We first reduce this problem to MGCD in the following.

Theorem 5.1. There exists a polynomial time algorithm which reduces the small principal ideal lattice problem to MGCD in Assumption 1.

Proof: According to the definition of [SV10, GH11], $v(x) \times s(x) = p \bmod (x^n + 1)$. Since α is a root of $f_n(x) = x^n + 1$ over modulo p, so we can factor $x^n + 1 = (x - \alpha) \bullet g(x) \bmod p$, where $g(x) = t(x) \bullet v(x) \bmod (x^n + 1)$. Now, we take $A = Rot(\vec{v})$, $R_1 = Rot(\vec{s})$, and $R_2 = Rot(\vec{t})$ where $\vec{v}, \vec{s}, \vec{t}$ respectively are coefficient vectors of $v(x), s(x), t(x)$. It is easy to verify $B_1 = A \times R_1 = diag(p, p, ..., p)$ and $B_2 = A \times R_2 = Rot(\vec{g})$. So, we reduce SPIP to MGCD. ■

Theorem 5.2. There exists a polynomial time algorithm which reduces AGCD to AMGCD in Assumption 2.

Proof: It is obvious. We only need to set $n = 1$ for AMGCD. ■

Theorem 5.3. There exists a polynomial time algorithm which reduces LWE to AMGCD in

Assumption 2.

Proof. Given a list sample (b_i, r_i) with $b_i =< a, r_i > + e_i \bmod p$. We first choose $2n$ samples (b_i, r_i) from the distribution $D_{n,p,\chi}$ of LWE, and randomly $a_j \in \mathbb{Z}_p^n, j = 2,...,n$. Then, we compute $b_{k,j} =< a_j, r_k > + e_{j,k}$, $j = 2,...,n; k = 1,...,2n$. Now, we set $R_i = (r_{1+(i-1)n}, r_{2+(i-1)n}, ..., r_{n+(i-1)n}), i = 1,2$, $A = (a, a_2, ..., a_n)^t$, and

$$B_1 = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ b_{1,2} & b_{2,2} & \cdots & b_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{1,n} & b_{2,n} & \cdots & b_{n,n} \end{pmatrix}, \quad B_2 = \begin{pmatrix} b_{n+1} & b_{n+2} & \cdots & b_{2n} \\ b_{n+1,2} & b_{n+2,2} & \cdots & b_{2n,2} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n+1,n} & b_{n+2,n} & \cdots & b_{2n,n} \end{pmatrix}.$$

It is not difficult to verify $B_i = AR_i + E_i, i = 1,2$. Since the probability what a random matrix A is satisfied for $p \mid \det(A)$ is $1/p$, and $p = n^{O(1)}$ in LWE. Thus, we have non-negligible probability to successfully reduce LWE to AMGCD.■

Theorem 5.4. Suppose Assumption 2 holds, then breaking PKE is hard.

Theorem 5.5. Suppose Assumption 3 holds, then breaking FHE is hard.

6. Conclusion and Open Problem

In this paper, we have constructed a new fully homomorphic encryption scheme, whose security depends on the hardness assumptions of the approximate MGCD problem.

This paper raises some interesting open problems. First, the security of our FHE is based on the hardness of variant approximate MGCD problem, which is a new problem. For this (variant) AMGCD, we do currently not know its computational hardness. It would be most desirable to reduce the SVP/CVP problem in lattice to the (variant) approximate MGCD problem. In addition, we hope to build the relationship between the (variant) approximate MGCD problem and the LWE problem [Reg09].

References

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In Proc. of STOC 1996, pages 99-108, 1996.

[AP09] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS*, pages 75–86, 2009.

[ACG08] C. Aguilar Melchor, G. Castagnos, and G. Gaborit. Lattice-based homomorphic

encryption of vector spaces. In IEEE International Symposium on Information Theory, ISIT'2008, pages 1858-1862, 2008.
[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. Lecture Notes in Computer Science, 2005, Volume 3378, pages 325-341, 2005.
[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In CRYPTO, 2011. To appear.
[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. ePrint Archive: Report 2011/344: http://eprint.iacr.org/2011/344.
[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Proc. of Eurocrypt, volume 6110 of LNCS, pages 24-43. Springer, 2010.
[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169-178, 2009.
[Gen11] C. Gentry. Fully Homomorphic Encryption without Bootstrapping. ePrint Archive: Report 2011/279: http://eprint.iacr.org/2011/277.
[GH11a] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, Advances in Cryptology — EUROCRYPT 2011, volume 6632 of Lecture Notes in Computer Science, pages 129–148, Berlin, Heidelberg, New York, 2011. Springer Verlag. Cryptology ePrint Archive: Report 2010/520: http://eprint.iacr.org/2010/520.
[GH11b] C. Gentry and S. Halevi, Fully homomorphic encryption without squashing using depth-3 arithmetic circuits, Cryptology ePrint Archive, Report 2011/279.
[GHV10] C. Gentry and S. Halevi and V. Vaikuntanathan. A Simple BGN-type Cryptosystem from LWE. In Proc. of Eurocrypt, volume 6110, pages 506-522, 2010.
[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proc. of STOC, pages 197-206, 2008.
[HG05] Roger A. Horn, Charles R. Johnson. Matrix Analysis, Cambridge University Press, 2005.
[LNV11] Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? Manuscript at http://www.codeproject.com/News/15443/Can-Homomorphic-Encryption-be-Practical.aspx, 2011.
[LPR10] V. Lyubashevsky and C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. In Proc. of Eurocrypt, volume 6110, pages 1–23, 2010.
[Mic07] D. Micciancio Generalized compact knapsaks, cyclic lattices, and efficient one-way functions. Computational Complexity, 16(4):365-411.
[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussion measures. SIAM Journal Computing, 37(1):267-302, 2007.
[NS06] P.Q. Nguyen and D. Stehle, LLL on the average, proc. Of ANTS VII, 2006, LNCS 4076, pp. 238-256.
[Reg09] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), 56(6), pages 1-40, 2009.

[Sho09] V. Shoup. NTL: A Library for doing Number Theory. http://shoup.net/ntl/, Version 5.5.2, 2009.
[SS10] D. Stehle and R. Steinfeld. Faster Fully Homomorphic Encryption. Cryptology ePrint Archive: Report 2010/299: http://eprint.iacr.org/2010/299.
[SV10] N. P. Smart and F. Vercauteren Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Lecture Notes in Computer Science, 2010, Volume 6056/2010, 420-443.
[SYY99] T. Sander, A. Young, and M. Yung. Non-interactive CryptoComputing for NC1. In 40th Annual Symposium on Foundations of Computer Science, pages 554{567. IEEE, 1999.
[RAD78] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, pages 169-180, 1978.
[Yao82] A. C. Yao. Protocols for secure computations (extended abstract). In 23rd Annual Symposium on Foundations of Computer Science (FOCS '82), pages 160-164. IEEE, 1982.