AIOL: An Improved Orthogonal Lattice Algorithm for the General Approximate Common Divisor Problem

Yinxia Ran , Yun Pan , Licheng Wang * , Zhenfu Cao

Posted Date: 11 October 2023

doi: 10.20944/preprints202310.0698.v1

Keywords: General Approximate Common Divisors; Fully Homomorphic Encryption; Lattice Attack; Orthogonal Lattice

Article

AIOL: An Improved Orthogonal Lattice Algorithm for the General Approximate Common Divisor Problem

Yinxia Ran[1,2,3], Yun Pan[1], Licheng Wang[4,*] and Zhenfu Cao[3,5]

[1] Communication University of China (CUC), 1 Dingfuzhuang East Street, Beijing 100024, P.R. China.
[2] Longnan Teachers College (LNTC), 34 Longnan Road, Longnan 742500, P.R. China.
[3] Zhejiang Lab, Kechuang Avenue, Zhongtai Sub-District, Hangzhou 311121, P.R. China.
[4] Beijing insititute of technology (BIT), 5 Zhongguancun South Street, Beijing 100081, P.R. China.
[5] East China Normal University (ECNU), 3663, North Zhongshan Road, Shanghai 200062, P.R. China.
* Correspondence: lcwang@bit.edu.cn

Abstract: The security of several fully homomorphic encryption (FHE) schemes depends on the intractability assumption of the approximate common divisor (ACD) problem over integers. Subsequent efforts on solving the ACD problem as well as its variants were also developed during the past decade. In this paper, an improved orthogonal lattice (OL) based algorithm, AIOL, is proposed to solve the general approximate common divisor (GACD) problem. The conditions for ensuring the feasibility of AIOL are also presented. Compared to the Ding-Tao's OL algorithm, the well-know LLL reduction is used only once in AIOL, and when the error vector $\mathbf{r}$ is recovered in AIOL, the possible difference between the restored and the true value of p is given. The experimental comparisons towards the Ding-Tao's algorithm and ours are also provided for validating our improvements.

Keywords: general approximate common divisors; fully homomorphic encryption; lattice attack; orthogonal lattice

1. Introduction

BACKGROUND. The approximate common divisor (ACD) problem was firstly studied by Howgrave-Graham [1]. Further interest in this problem was inspired by the proposal of fully homomorphic encryption (FHE) due to Van Dijk et al. [2], as well as cryptographic constructions proposed subsequently [3–5]. The security of these cryptosystems depends on the hardness assumption of the ACD problem and its variants.

The ACD problem is usually formulated in two ways: the problem of general approximate common divisor (GACD) and the problem of partial approximate common divisor (PACD). Both of them take as inputs polynomially many samples $x_i = pq_i + r_i$ with sufficiently small but non-zero r_i, and aim to work out the *hidden common divisor* p, while the latter is given an additional *exact* sample $x_0 = pq_0$ (i.e. $r_0 = 0$). Intuitively, the PACD problem is easier than GACD, considering that one can work out p directly if he/she knows the factorization of the additional sample x_0, whereas the capability of integer factorization has no direct impact on the GACD problem. However, Van Dijk et al. pointed out that at present there is no PACD algorithm that does not work for GACD [2]. And the usefulness of PACD has been demonstrated by a much more efficient construction of FHE scheme [3] of which the security is proved relied on PACD, rather than on GACD.

The original papers [1,2] presented a few possible lattice attacks on the GACD problem, including orthogonal lattices (OL) method, simultaneous diophantine approximation (SDA) method, and multivariate polynomial equations (MP) method. Further cryptanalytic work was done by [4,6–14]. Among these work, the OL algorithm due to Ding and Tao [8] is ingenious by using the well-know LLL algorithm for twice to accurately recover the error vector $\mathbf{r}$. After mapping the given GACD instances into a lattice $\mathcal{L}$, the first calling LLL is to find suitable $t - z (z = 1, 2)$ short vectors $\mathbf{u}_i (i = 1, 2, \cdots, t - z)$ for establishing the equations

$$\mathbf{U} \cdot \mathbf{x} = \mathbf{U} \cdot \mathbf{r} \quad \text{for} \quad \mathbf{U} = [\mathbf{u}_1 | \cdots | \mathbf{u}_{t-z}], \tag{1}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

where $\mathbf{u_i} = (u_{i1}, u_{i2}, \cdots, u_{it})$. Then, a new lattice $\mathcal{L}'$ is constructed by using the base vectors of the solution space of (1), and the second calling LLL is to recover the error vector $\mathbf{r}$ accurately. With knowing $\mathbf{r}$ it is very easy to recover p even for a primary school student, say by using the extended Euclidean algorithm. According to Ding and Tao [8], it is an *amazing* thing that why the first calling LLL over $\mathcal{L}$ should give solutions for (1), and they claimed that a theoretical proof would be a very significant result. Another merit of the Ding-Tao's method is that the setting on the related parameters is simple and this makes the implementation of OL attacks towards GACD-based cryptosystems very easy in practice. For example, the lower bound of the number of samples t depends only on γ, and the length of the short vector $\mathbf{v}$ depends only on t and γ.

MOTIVATION AND CONTRIBUTIONS. With further experiments on the Ding-Tao's algorithm, we find that the actual effect of the algorithm is *better* than they claimed. In particular, we realize that the conditions $\rho < \eta/2$ and $t \geq (4\gamma)^{1/3}$ could be relaxed and merged, and the second calling LLL could also be saved. Moreover, we find that even for failure executions of the Ding-Tao's algorithm, there is a high probability that the recovered p differs from the actual value by only 1 or very small numbers. Therefore, our motivation in this work is to propose an improved OL algorithm to reduce both space and time costs for solving the GACD problem. Our main contributions are summarized as follows:

- First, we modify the range of parameters N, t and $\|\mathbf{v}\|$ in the Ding-Tao's algorithm, so that we need to build lattice and call LLL for only once, and the success rate for recovering p reaches 100%, under the merged condition

$$t \geq \max\left\{4, \frac{5}{3}\left\lceil \eta - \rho - \sqrt{(\eta-\rho)^2 - 1.2(\gamma+\rho)}\right\rceil\right\}. \tag{2}$$

  Note that (2) also implies $\rho \leq \eta + 0.6 - \sqrt{1.2(\eta+\gamma+3)}$, no matter whether $\rho < \eta/2$ holds.
- Second, based on the above modification, we give a proof on why in our algorithm AIOL, the only once calling LLL will give us solutions for (1). This can be viewed as a theoretical answer towards the Ding-Tao's *amazing* question;
- Third, we give the possible differences between the recovered of p and the actual hidden common divisor when the error vector $\mathbf{r}$ is recovered. Knowing these differences is in turn helpful for recovering p, and thus expanding the scope of OL attacks.

ROADMAP. The remained contents are organized as follows: In Section 2, the formal definitions of the problems of GACD and PACD are given, and the lattice concepts and the LLL algorithm are introduced beriefly; In Section 3, the orthogonal lattice based approach, including our improvements, for GACD are explored and developed in detail; Experiments and comparisons as well as related discussions are presented in Section 4; Finally, concluding remarks are given in Section 5.

2. Preliminaries

Throughout this paper, we make the following agreeent on notations: Capital boldface letters denote matrices, e.g. $\mathbf{A}$, while lowercase bold letters denote vectors e.g. $\mathbf{a}$; Let $(\cdot,\cdot)$ and $\|\cdot\|$ be the inner product and the l_2 Euclidean length respectively, and $\mathbf{A^T}$ denote the transpose of matrix $\mathbf{A}$; The logarithmic notation log always takes 2 as the base, while $\lceil r \rceil$ denotes the smallest integer not less r.

Definition 1 (ACD Distribution). *Given $\gamma, \eta, \rho \in \mathbb{N}$, let p be an η-bit odd integer, the ACD distribution, $D_{\gamma,\rho}(p)$, is an efficiently sampleable distribution define as follows:*

$$D_{\gamma,\rho}(p) = \{pq + r | q \leftarrow \mathbb{Z} \cap (0, 2^\gamma/p), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)\}. \tag{3}$$

Definition 2 (GACD Problem). *Given access to an ACD distribution $D_{\gamma,\rho}(p)$ as a oracle, the objective of the general approximate common divisor (GACD) problem is to find p.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Definition 3 (PACD Problem). *Given access to an ACD distribution $D_{\gamma,\rho}(p)$ as a oracle, with the restriction that the first output of $D_{\gamma,\rho}(p)$ is $x_0 = pq_0$ for some $q_0 \leftarrow \mathbb{Z} \cap (0, 2^\gamma / p)$, the objective of the partial approximate common divisor (PACD) problem is to find p.*

Remark 1. *Apparently, a PACD instance is a GACD by coincidence only with the probability that is negligible with respect to ρ.*

Definition 4 ($\delta-$LLL reduction basis). *Given a lattice basis $\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_n})$, the corresponding Gram-Schmidt basis $\mathbf{B}^* = (\mathbf{b_1^*}, \cdots, \mathbf{b_n^*})$, $\mathbf{B}$ is a reduced basis if and only if the following two conditions are satisfied:*

(1) (Size condition) $\mu_{i,j} = \dfrac{(\mathbf{b_i}, \mathbf{b_j^*})}{\|\mathbf{b_j^*}\|^2} \leq 1/2$, *for all* $1 \leq j < i \leq n$;

(2) (Lovász condition) $\|\mathbf{b_i^*}\|^2 \geq (\delta - \mu_{i,i-1}^2)\|\mathbf{b_{i-1}^*}\|^2$, *for all* $1 < i \leq n$, *where* $1/4 < \delta < 1$.

Definition 5 (Geometric Series Assumption [15]). *Given Gram-Schmidt basis $(\mathbf{b_1^*}, \cdots, \mathbf{b_n^*})$,*

$$\frac{\|\mathbf{b_i^*}\|}{\|\mathbf{b_1}\|} = \theta^{i-1}, \tag{4}$$

for $i = 1, 2, \cdots, n$, where $3/4 \leq \theta < 1$ is called GSA constant.

The Geometric Series Assumption (GSA) means the length of Gram-Schmidt basis $\|\mathbf{b_i^*}\|$ with LLL reduction decays geometrically with quotient θ and indicates

$$\|\mathbf{b_i^*}\| \leq \|\mathbf{b_1}\| (i = 1, 2, \cdots, n). \tag{5}$$

Theorem 1. *[16] Given a LLL reduction lattice basis $\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_n})$, $(\mathbf{b_1^*}, \cdots, \mathbf{b_n^*})$ is the corresponding Gram-schmidt basis. The following results hold:*

(1) $\|\mathbf{b_1}\| \leq \alpha^{\frac{n-1}{4}} |\det(\mathbf{B})^{\frac{1}{n}}|$;

(2) $\|\mathbf{b_j^*}\| \leq \alpha^{\frac{(i-j)}{2}} \|\mathbf{b_i^*}\|$, *for* $1 \leq j < i \leq n$;

(3) $\|\mathbf{b_j}\| \leq \alpha^{\frac{(i-1)}{2}} \|\mathbf{b_i^*}\|$, *for* $1 \leq j < i \leq n$;

where $\alpha = \dfrac{1}{\delta - \frac{1}{4}}$, *$\delta$ is the parameter in the Definition 4.*

3. Orthogonal Lattice (OL) based approach

3.1. The basic idea of OL algorithm

Nguyen and Stern [17] have demonstrated the usefulness of the orthogonal lattice in cryptanalysis, and this has been used in several ways to attack the ACD problem. The idea is to find $\mathbf{u} = (u_1, u_2, \cdots, u_t) \in \mathcal{L}^\perp(\mathbf{q}, \mathbf{r})$ that is orthogonal to both $\mathbf{q} = (q_1, q_2, \cdots, q_t)$ and $\mathbf{r} = (r_1, r_2, \cdots, r_t)$. Since $x_i = pq_i + r_i$, $\mathbf{x} = (x_1, x_2, \cdots, x_t)$ is orthogonal to $\mathbf{u}$. The task is to find $t-1$ linearly independent vectors $\mathbf{u}$ shorter than any vector in $\mathcal{L}^\perp(\mathbf{x})$ to recover $\mathbf{q}$, $\mathbf{r}$ and therefore p.

Based on the idea of Nguyen and Stern, the current idea is to find $t - z (z = 1, 2)$ linearly independent vectors $\mathbf{u}$ only orthogonal to $\mathbf{q}$. The core steps of the current OL algorithm include the following two steps:

First, find $t - z (z = 1, 2)$ linearly independent vectors $\mathbf{u}$ orthogonal to $\mathbf{q}$, that is,

$$\sum_{i=1}^{t} u_i \cdot q_i = 0 \tag{6}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

then establish and solve indefinite equations (1).

Second, find small positive integer solutions to (15). At present, the common way to find the small solutions is to construct the lattice $\mathcal{L}'$ with basis matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{d_0} \\ \mathbf{d_1} \\ \vdots \\ \mathbf{d_z} \end{pmatrix} \tag{7}$$

and then employ the LLL algorithm to reduce the basis matrix $\mathbf{B}$, with the hope of that the first output is the vector $\mathbf{r}$. However, at present, what can meet this expectation are experimental conditions, and there is still a lack of theory.

Let the general solution formula of (15) be

$$\mathbf{d} = \mathbf{d_0} + t_1\mathbf{d_1} + \cdots + \mathbf{d_z} \tag{8}$$

where $\mathbf{d_0}$ is a special solution of (15), $t_1, \cdots, t_z$ are integers, $\mathbf{d_1}, \cdots, \mathbf{d_z}$ is a basis of integer solution space for the corresponding homogeneous linear equations.

Let $\mathbf{d}' \in \mathcal{L}'$, then

$$\mathbf{d}' = k_0\mathbf{d_0} + k_1\mathbf{d_1} + \cdots + k_z\mathbf{d_z} \tag{9}$$

where $k_0, k_1, \cdots, k_z$ are integers. Obviously, when $k_0 = 1$, (9) = (8). Reduce the lattice $\mathbf{B}$ to $\mathbf{B}'$:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{d}'_0 \\ \mathbf{d}'_1 \\ \vdots \\ \mathbf{d}'_z \end{pmatrix}. \tag{10}$$

To facilitate finding $\mathbf{r}$, consider the explicit vectors $\mathbf{d}'_0, \mathbf{d}'_1, \cdots, \mathbf{d}'_z$. It's easy to deduce that only one of them is the solution to (15).

Let $\mathbf{d}'_i$ is the solution to (15), and if $\mathbf{d}'_i = \mathbf{d}'_0$, then $\mathbf{d}'_0$ is probably equal to $\mathbf{r}$. With this in mind, Ding and Tao [8] found the conditions that the algorithm can work well (theoretically not proved):

$$\rho < \frac{\eta}{2} \quad \text{and} \quad t \geq (4\gamma)^{1/3}. \tag{11}$$

In addition, if $\mathbf{d}'_i \neq \mathbf{d_0}$, we find an interesting thing that the recovery value p' is only 1 or a very small number different from the true value p in many cases of our experiment. And our experiments lead to the following general conclusions between p and p':

Let $\mathbf{d}'_i = (u_{i1}, u_{i2}, \cdots, u_{it}) \neq \mathbf{d}'_0, d_{ru} = \gcd(r_1 - u_{i1}, r_2 - u_{i2}, \cdots, r_t - u_{it})$, then

$$p' - p = d_{ru}, \tag{12}$$

where p' is the recovered value of p. So, if $\mathbf{d}'_i \neq \mathbf{d_0}$, using vector $\mathbf{d}'_i$, p' can be restored. And since d_{ru} is bounded, p can be restored by p'.

In summary, one of the outputs $\mathbf{d_1}, \cdots, \mathbf{d_z}$ generated by the LLL algorithm can be used to recover $\mathbf{r}$ under the appropriate conditions.

3.2. Our Proposal

In this part, an improved OL algorithm, AIOL, is described in detail.

Through the proof in the next section, it can be seen that when the condition

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Algorithm 1 AIOL: An improved OL algorithm for GACD

Input: The GACD parameters $\gamma, \eta, \rho \in \mathbb{N}$, and t ACD samples $\{x_1, \cdots, x_t\} \xleftarrow{\$} D_{\gamma,\rho}(p)$, with t satisfying

$$t \geq \max\left\{4, \left\lceil \frac{5}{3}\left(\eta - \rho - \sqrt{(\eta-\rho)^2 - 1.2(\gamma+\rho)}\right)\right\rceil\right\}. \tag{13}$$

Output: The approximate greatest common divisor p.

1. Randomly choose $N \in (2^{\gamma+\eta-1}, 2^{\gamma+\eta})$. And then constract a lattice $\mathcal{L}$ with the basis

$$B = \begin{pmatrix} 1 & & & x_1 \\ & 1 & & x_2 \\ & & \ddots & \vdots \\ & & 1 & x_t \\ & & & N \end{pmatrix}. \tag{14}$$

2. Reduce lattice $\mathcal{L}$ by calling the LLL algorithm with $\delta = 3/4$. Let the reduced basis be $\mathbf{V} = [\mathbf{v_1}|\cdots|\mathbf{v_{t+1}}]$, where $\mathbf{v_i} = (u_{i1}, \cdots, u_{it}, v_{i(t+1)}), (i = 1, 2, \cdots, t+1)$.

3. Collect short vectors from $\mathbf{V}$ so that $\|\mathbf{v_i}\| < 2^{\eta-\rho-2-\log\sqrt{t}}$, $(i = 1, 2, \cdots, t-z)$, where $z = 1, 2$. And then, solve the following Diophantine equations with t unknowns $r_1, \cdots, r_t$:

$$\sum_{j=1}^{t} u_{ij} \cdot r_i = \sum_{j=1}^{t} u_{ij} \cdot x_i (i = 1, \cdots, t-z). \tag{15}$$

4. Rewrite the integer solutions of (15) as follows:

$$\mathbf{d} = \mathbf{d_0} + t_1\mathbf{d_1} + \cdots + t_z\mathbf{d_z}, \tag{16}$$

where $\mathbf{d_0}$ is a special solution of the Diophantine equations, $t_1, \cdots, t_z$ are integers, $\mathbf{d_1}, \cdots, \mathbf{d_z}$ is a basis of integer solution space for the corresponding homogeneous linear equations.

5. Let $\mathbf{r} = \mathbf{d_0}$.

6. Compute $p = \gcd(x_1 - r_1, x_2 - r_2)$.

$$(\eta - \rho)^2 \geq 1.2(\gamma + \rho) \tag{17}$$

or equivelently

$$\rho \leq \eta + 0.6 - \sqrt{1.2(\eta + \gamma + 3)} \tag{18}$$

holds, the algorithm AIOL will successfully recover p.

3.3. The proof of AIOL algorithm

Lemma 1. *For $\forall\, \mathbf{v} \in \mathcal{L}$, if $\|\mathbf{v}\| < 2^{\eta-\rho-2-log\sqrt{t}}$, then the equation (15) holds.*

Proof. Let $\mathbf{v} = (u_1, u_2, \cdots, u_t, \sum_{i=1}^{t} u_i x_i + u_{t+1}N)$, $M = 2^{\eta-\rho-2-\log\sqrt{t}}$, then

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{t} u_i^2 + \left(\sum_{i=1}^{t} u_i x_i + u_{t+1}N\right)^2} < M. \tag{19}$$

Thus

$$|u_i| < M, \left|\sum_{i=1}^{t} u_i x_i + u_{t+1}N\right| < M (1 \leq i \leq t). \tag{20}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Since $2^{\gamma+\eta-1} \leq N \leq 2^{\gamma+\eta}$,

$$\begin{aligned}\left|\sum_{i=1}^{t} u_i x_i\right| &\leq 2^{\gamma}\sqrt{t}\cdot\|\mathbf{u}\| \\ &\leq 2^{\gamma}\sqrt{t}\cdot\|\mathbf{v}\| \\ &\leq 2^{\gamma}\sqrt{t}\cdot 2^{\eta-\rho-2-\log\sqrt{t}} \\ &= 2^{\gamma+\eta-\rho-2} < N/2.\end{aligned} \tag{21}$$

Therefore, there is no modular N operation and $u_{t+1} = 0$. So $\mathbf{v} = (u_1, u_2, \cdots, u_t, \sum_{i=1}^{t} u_i x_i)$.

We also have

$$\left|\sum_{i=1}^{t} u_i r_i\right| \leq 2^{\rho}\sqrt{t}\cdot\|\mathbf{v}\| \leq 2^{\eta-2}. \tag{22}$$

To prove that (6) holds, suppose $\left|\sum_{i=1}^{t} u_i q_i\right| \neq 0$, so

$$p\left|\sum_{i=1}^{t} u_i q_i\right| \geq p \geq 2^{\eta-1} \tag{23}$$

$$\left|\sum_{i=1}^{t} u_i x_i\right| = \left|p\sum_{i=1}^{t} u_i q_i + \sum_{i=1}^{t} u_i r_i\right| \geq p\left|\sum_{i=1}^{t} u_i q_i\right| - \left|\sum_{i=1}^{t} u_i r_i\right| \geq 2^{\eta-1} - 2^{\eta-2} = 2^{\eta-2}, \tag{24}$$

but

$$\left|\sum_{i=1}^{t} u_i x_i + u_{t+1}N\right| = \left|\sum_{i=1}^{t} u_i x_i\right| < M = 2^{\eta-\rho-2-\log\sqrt{t}} < 2^{\eta-2}. \tag{25}$$

This is a contradiction. The equations (6) and (15) hold. Then Lemma 1 holds. □

Lemma 2. *If the number t of samples satisfies*

$$(4/3)^{(3t-2)/4}\cdot 2^{(\gamma+\eta)/(t+1)} \leq 2^{\eta-\rho-2-\log\sqrt{t}}, \tag{26}$$

then LLL reduction basis vectors is valid for the construction of equation (15).

Proof. For the $(t-1)$-th LLL reduction basis vector $\mathbf{v_{t-1}}$, whose length can be estimated

$$\begin{aligned}\|\mathbf{v_{t-1}}\| &\leq \alpha^{(t-1)/2}\|\mathbf{v_t^*}\| \\ &= (4/3)^{(t-1)/2}\|\mathbf{v_t^*}\|(\alpha = 4/3) \\ &\leq (4/3)^{(t-1)/2}\|\mathbf{v_1}\|(by(5)) \\ &\leq (4/3)^{(t-1)/2}\cdot(4/3)^{t/4}\cdot|\mathbf{B}|^{1/(t+1)}(by\ Theorem\ 1) \\ &\leq (4/3)^{(3t-2)/4}\cdot 2^{(\gamma+\eta)/(t+1)}.\end{aligned} \tag{27}$$

According to Lemma 1, equation (15) holds when (26) is true. □

Based on the above two lemmas, the following theorem can be obtained.

Theorem 2. *When GACD parameters satisfy (17) or (18) and the number of samples satisfy (13), then we construct the equation (15) to be true.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Proof. From the condition (26), the length of LLL reduction basis vectors satisfies Lemma 2, then LLL reduction basis vectors is valid for the constrction of the equation (15). Combined the above two lemmas, we simiplify the inequation and ignore some small terms to get the following bound of sample numbers t. The specific process is as follows: take the logarithm base 2 on both sides of (26) to obtain:

$$\frac{3t-2}{4}\log\frac{4}{3}+\frac{\gamma+\eta}{t+1}\leq\eta-\rho-2-\log\sqrt{t}. \tag{28}$$

Remove some smaller items of (28), $\log\frac{4}{3}\approx 0.4$, we have

$$0.3t+\frac{\gamma+\eta}{t+1}\leq\eta-\rho, \tag{29}$$

Sort out the formula (29), we get

$$0.3t^2-(\eta-\rho-0.3)t+(\gamma+\eta)\leq 0, \tag{30}$$

then

$$0.3t^2-(\eta-\rho)t+(\gamma+\eta)\leq 0, \tag{31}$$

By solving the inequality (31), we can get

$$t\geq\frac{5}{3}\left(\eta-\rho-\sqrt{(\eta-\rho)^2-1.2(\gamma+\rho)}\right). \tag{32}$$

In summary, when conditions (18) and (32) hold, the algorithm can recover p successfully. Note that the condition $t\geq 4$ comes from the third step of AIOL where we need to collect at least 2 short vectors for building the required Diophantine equations. Then the condition (13) is true. Hence,Theorem 2 holds. □

4. Experiments and Comparisons

In this section, we conduct experiments towards our algorithm AIOL, as well as the Ding-Tao's algorithm. The experimental environment is specified as below: Intel(R) Core (TM) Processor i5-1235U CPU (1.30 GHz) with 16 GB of memory, Windows 10 OS, and Maple 2021 coding language.

The experiments are organized as two steps. Firstly, to test the effects on relaxation on conditions of ρ and t, we adopt the following settings on the related parameters:

- Fix $\eta=160$;
- Let $\gamma=300,400,500,1000,1500,2000$ respectively;
- For each case of setting on γ, running the Ding-Tao's algorithm and our AIOL algorithm for 100 times for different ρ (resp. t) around the upper (resp. lower) bound of ρ (resp. t) given by the Ding-Tao's condition (11) and our condition (13,18), respectively.
- Then, for each case, collect the success rate for recovering the hidden common divisor p, as well as the maximal ρ (resp. the minimal t) that enables the related algorithms work. These results are summaried in Table 1, where the symbol '–' indicates that in this case the related failed to work out.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Table 1. Experiments and Comparisons: Conditions on ρ, t and Success rate ($\eta = 160$).

γ	Ding-Tao					AIOL				
	ρ (11)	$\rho_{\max}$	t (11)	t_{min}	succ %	ρ (18)	$\rho_{\max}$	t (13)	$t_{\min}$	succ %
300	79	103	11	11	82%	137	137	35	17	100%
400	79	91	12	12	87%	134	134	34	19	100%
500	79	80	13	13	90%	131	132	39	23	100%
1000	79	30	16	16	89%	122	123	54	33	100%
1500	79	–	19	–	–	115	115	60	40	100%
2000	79	–	21	–	–	109	109	72	46	100%

From Table 1, we can see that:

- The overall success rate of our algorithm is 100%, which is observably higher than that of in the Ding-Tao's algorithm, under the same settings on η, γ and a similar scale of t. Moreover, even for bigger settings on ρ in AIOL, the success rates are still higher than those of obtained by the Ding-Tao's algorithm for the smaller settings on ρ. [1]
- The condition on ρ given by the Ding-Tao's condition (11) is *irrelevant*, considering for $\gamma = 300$ and $\gamma = 1000$, the maximal values of ρ for ensuring the Ding-Tao's a high success rate are 103 and 30, respectively. They are respectively either observably bigger or smaller than the given bound $79 < \eta/2$.
- The condition on ρ given by AIOL is relaxed to the case of $\rho > \eta/2$. And this condition is *tight in the sense that* for all these cases, the the maximal values of ρ for ensuring AIOL success are almost same with the bound given by (18).
- The condition on t given by the Ding-Tao's condition (11) is rigorous *in the sense that* for even small t, our tests on the Ding-Tao's algorithm failed, whereas the condition on t given by (13) in AIOL is *loose* since for even small t, our algorithm still works well. At present, we have no idea to give a tight bound on choicing t for the AIOL algorithm.

Secondly, to test the scalability, as well as the speed, we adopt the settings according to experiments given by Ding and Tao in [8]. That is,

- Fix $\eta = 1000$ and $\rho = 450$;
- Let $\gamma = 5000, 10000, 15000, 20000$ respectively;
- In Ding-Tao's algorithm, let $t = 18, 40, 59, 85$ respectively, according to what was given in [8], while in our algorithm AIOL, t is set to $10, 20, 29, 38$ respectively – calculated according to the condition given by (13).
- Then, for each case, we run Ding-Tao's algorithm and our algorithm respectively, and then collect the running time for getting correct results in Table 2. (Note that for conveniences doing comparisons, partial data on the running time in Ding-Tao's paper [8] is referenced here.)

[1] Intuitively, the bigger ρ, the more errors involved in the given ACD samples, and this in turn means the harder for solving the given GACD instances.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 October 2023 doi:10.20944/preprints202310.0698.v1

Table 2. Experiments and Comparisons: Scalability and Speed ($\eta = 1000$).

γ	Ding-Tao				AIOL		
	ρ	t	time (s)[1]	time (s)[2]	ρ	$t(13)$	time (s)[2]
5000	450	18	2.386	207.09	450	10	40.15
10000	450	40	91.447	7436.85	450	20	1162.34
15000	450	59	749.179	61767.34	450	29	3793.10
20000	450	85	4245.879	141303.98	450	38	32651.77

[1] Running time in MAGMA according to [8]. [2] Running time in MAPLE according to our experiments.

From Table 2, we can see that:

- Both our algorithm AIOL and the Ding-Tao's algorithm have good performance in scalability in the parameter γ. Moreover, the AIOL algorithm can find the correct solutions with even smaller t, this in turn means less space cost for storing the ACD samples.
- With the sample computational environments (i.e. MAPLE coding on an Intel i5 CPU with 1.30 GHz clocks), our AIOL algorithm runs much quicker than the Ding-Tao's algorithm, under the same settings on γ, η and ρ.
- Ding-Tao's tests given in [8] (i.e. MAGMA coding on two Quad-Core Intel Processor Q9400 CPUs with 2.66 GHz clocks) are much quicker than our tests on both AIOL and the Ding-Tao's algorithm. We think that this might be mainly attributed to the differences of computational environments.

5. Conclusions

The interest in the general approximate common divisor (GACD) problem is excited by the possibility of building fully homomorphic encryptions over integers, though many such kind of cryptographic constructions were broken. In fact, from an even abstract point, the GACD problem can be viewed as a *learning-with-error (LWE)* version of the greatest common divisor (GCD) problem over the 1-dimension lattice $\mathbb{Z}$. Although we know that all lattice problems are easy for low dimensions, more efforts are still needed to tackle the GACD problem. In this paper, an improved orthogonal lattice algorithm, AIOL, is proposed for solving GACD. Compared with the Ding-Tao's OL method, the parameter conditions for suit for AIOL is relaxed, and the experiments show that the success rate of AIOL is enhanced observably.

Author Contributions: Conceptualization, Ran Y., Wang L. and Cao Z.; Methodology, Ran Y., Wang L. and Cao Z.; Validation, Pan Y. and Wang L.; Writing—original draft preparation, Ran Y.; Writing—review and editing, Ran Y., Wang L.; Code implementation, Ran Y. and Wang L.; Supervision and project administration, Pan Y.

Funding: This research is partially supported by the National Natural Science Foundation of China (NSFC) (62272040).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. N. Howgrave-Graham. Approximate integer common divisors. Cryptography and Lattices. *Springer Berlin Heidelberg*, **2001**: 51–66.
2. M. Van Dijk, C.Gentry, S. Halevi, V. Vaikuntanathan, Fully homomorphic encryption over the integers, in: H. Gilbert (ed.), *Advances in Cryptology–EUROCRYPT 2010*, Lecture Notes in Comput. Sci. Springer, Berlin, Heidelberg, **2010**, 6110: 24–43.
3. J. S. Coron, A. Mandal, D. Naccache, M. Tibouchi, Fully homomorphic encryption over the integers with shorter public keys, in: P. Rogaway (ed.), *Advances in Cryptology-CRYPTO 2011*, Lecture Notes in Comput. Sci, Springer, Berlin, Heidelberg, **2011**, 6841: 487–504.

4. J. S. Coron, D. Naccache, M. Tibouchi. Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers. *In D. Pointcheval and T. Johansson (ed.), EUROCRYPT'12*, Springer LNCS, **2012**, 7237: 446–464.
5. J. H. Cheon, D. Stehlé. Fully Homomorphic Encryption over the Integers Revisited. In E. Oswald and M. Fischlin (eds.), *EUROCRYPT'15, Springer LNCS*, **2015**, 9056: 513-536.
6. Y. Chen, P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully homomorphic encryption challenges over the integers. *Advances in Cryptology-EUROCRYPT 2012*. Springer Berlin Heidelberg, **2012**: 502–519.
7. H. Cohn, N. Heninger. Approximate common divisors via lattices. In *proceedings of ANTS X, vol. 1 of The Open Book Series*, **2013**: 271–293.
8. J. Ding, C. Tao. A New Algorithm for Solving the General Approximate Common Divisors Problem and Cryptanalysis of the FHE Based on the GACD problem. *Cryptology ePrint Archive*, Report 2014/042, **2014**.
9. S. Gebregiyorgis. Algorithms for the Elliptic Curve Discrete Logarithm Problem and the Approximate Common Divisor Problem. PhD thesis, *The University of Auckland*, Auckland, New Zealand, **2016**.
10. S. Galbraith, S. Gebregiyorgis, S. Murphy. Algorithms for the approximate common divisor problem. *LMS Journal of Computation and Mathematics*. 19(A), **2016**.: 58-72.
11. Xiaoling Yu, Yuntao Wang, Chungen Xu, Tsuyoshi Takagi. Studying the Bounds on Required Samples Numbers for Solving the General Approximate Common Divisors Problem. *2018 5th International Conference on Information Science and Control Engineering*, http://dx, dio. org/10-1109/ICISCE. **2018**. 00117.
12. J. Xu, S. Sarkar, L. Hu, Revisiting orthogonal lattice attacks on approximate common divisor problems and their applications. *Cryptology ePrint Archive*, **2018**.
13. J. H. Cheon, W. Cho, M. Hhan, Algorithms for CRT-variant of approximate greatest common divisor problem. *Journal of Mathematical Cryptology*, **2020**, 14(1): 397–413.
14. W. Cho, J. Kim, C. Lee. Extension of simultaneous Diophantine approximation algorithm for partial approximate common divisor variants. *IET Information Security*, **2021**, 15(6): 417–427.
15. Claus-Peter Schnorr. Lattice reduction by random sampling and birthday methods. In STACS 2003, *20th Annual Symposium on Theoretical Aspects of Computer Science*, Berlin, Germany, February 27–March 1, Proceedings, **2003**: 145-156.
16. J. Hoffstein, J. Pipher, and J. H. Silverman. An Introduction to Mathematical Cryptography. *Springer Publishing Company*, 2nd edition, **2014**.
17. P. Q. Nguyen and Jacques Stern. The Two Faces of Lattices in Cryptology. In *J. Silverman (ed.), Cryptography and Lattices, Springer LNCS 2146*, **2001**: 146–180.