Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Alogrithms for the ACD Problem and Cryptanalysis of ACD-based FHE schemes, Revisited

Yinxia Ran[1,2], Yun Pan[1] and Licheng Wang[3,*]

[1] Communication University of China (CUC), 1 Dingfuzhuang East Street, Beijing 100024, China.
[2] Longnan Teachers College (LNTC), 34 Longnan Road, Longnan 742500, China.
[3] Beijing insititute of technology (BIT), 5 Zhongguancun South Street, Beijing 100081, China.
[*] Correspondence: lcwang@bit.edu.cn

Abstract: The security of several full homomorphic encryption (FHE) schemes depends on the hardness of the approximate common divisor (ACD) problem. The analysis of attack and defense against the system is one of the frontiers of cryptography research. In this paper, the performance of existing algorithms, including orthogonal lattice, simultaneous diophantine approximation, multivariate polynomial and sample pre-processing are reviewed and analyzed for solving the ACD problem. Orthogonal lattice (OL) algorithms are divided into two categories (OL-$\wedge$ and OL-$\vee$) for the first time. And an improved algorithm of OL-$\vee$ is presented to solve the GACD problem. This new algorithm works well in polynomial time if the parameter satisfies certain conditions. Compared with Ding and Tao's OL algorithm, the lattice reduction algorithm is used only once, and when the error vector $\mathbf{r}$ is recovered in Ding et al.'s OL algorithm, the possible difference between the restored and the true value of p is given. It is helpful to expand the scope of OL attacks.

Keywords: Approximate common divisors; Fully homomorphic encryption; lattice attack, orthogonal lattice.

1. Introduction

It is well known that the Greatest Common Divisor (GCD) problem has been widely and deeply studied. Euclid algorithm is a classical algorithm for solving the GCD problem, which is called by Knuth [4] as the ancestor of all GCD algorithms. In the last two decades, many improvements to the GCD algorithms have been proposed. And these algorithms have a wide range of applications in computational algebra and cryptography. However, the approximate common divisor (ACD) problem remains a number theory problem. The ACD problem was first studied by Howgrave-Graham [5]. Further interest in this problem was proposed by the homomorphic encryption (FHE) scheme of Van Dijk et al. [16] and its variants [19,24,34]. The security of these cryptosystems depends on the hardness assumption of the ACD problem and its variants.

Fix $\gamma, \eta, \rho \in \mathbb{N}^*$, let p be an η-bit odd integer, define the efficiently sampleable distribution $D_{\gamma,\rho}(p)$ as

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

The ACD problem is usually formulated in two ways: general approximate common divisor (GACD) problem and partial approximate common divisor (PACD) problem. Given polynomially many samples $x_i = pq_i + r_i$ from $D_{\gamma,\rho}(p)(r_i \neq 0$ for all i), to calculate p, which is the GACD problem. Given polynomially many samples $x_i = pq_i + r_i$ from $D_{\gamma,\rho}(p)$, as well as a sample $x_0 = pq_0$ for uniformly chosen $q_0 \in \mathbb{Z} \cap [0, 2^{\gamma}/p)$, to compute p, which is the PACD problem.

By definition, PACD cannot be harder than GACD, and intuitively it seems that it should be easier than GACD. However, Van Dijk et al.] mentioned that there was

© 2023 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

no PACD algorithm that did not work for GACD. And the usefulness of PACD was demonstrated by the construction [19], where a much more efficient variant of the scheme [16] was built, whose security relied on PACD rather than GACD. Thus, it is very important to get whether or not PACD is actually easier than GACD. Moreover, ACD problems were divided into computational and decision versions. Coron et al. [32] pointed out that the two versions are equivalent. By definition, PACD cannot be harder than GACD, and intuitively it seems that it should be easier than GACD. However, Van Dijk et al. [16] mentioned that there was no PACD algorithm that did not work for GACD. And the usefulness of PACD was demonstrated by the construction [19], where a much more efficient variant of the scheme [16] was built, whose security relied on PACD rather than GACD. Thus, it is very important to get whether or not PACD is actually easier than GACD. Moreover, ACD problems were divided into computational and decision versions. Coron et al. [32] pointed out that the two versions are equivalent.

The variant problems based on ACD mainly include CRT-ACD and CS-ACD. Cheon et al. [26] and Lepoint [33] call the problem of computing $p_1, \cdots, p_l$ from the public key the CRT-ACD problem. Cheon and Stehle [34] proposed a FHE scheme whose parameters are set to $(\rho, \eta, \gamma) = (\lambda, \lambda + \log\lambda, \Omega(d^2\lambda\log\lambda))$, Where d is the depth of the circuit for homomorphism computation. The problem corresponding to this set of parameters is called CS-ACD. Despite the utility of the two variants, the algorithms that secures its security foundation have not been probed well enough.

The original papers [5,16] presented a few possible lattice attacks on this problem, including orthogonal lattices (OL), simultaneous diophantine approximation (SDA) and multivariate polynomial equations (MP). Further cryptanalytic work was done by [23–25,30,31,36,37,41–44]. This paper surveys and compares the known lattice algorithms for the ACD problem.

Our main contribution is to propose an improved algorithm of OL-$\vee$ to reduce both space and time costs for solving the GACD problem. Another contribution is to give the possible difference between the restored and the true value of p when $\mathbf{r}$ is recovered, which is helpful to expand the scope of OL attacks. Our third contribution is to analyze the application range and performance of SDA, OL, MP algorithms and pre-processing. These work is very helpful for cryptographic algorithm attacks to achieve better results.

2. Preliminaries

Throughout this paper, capital boldface letters denote matrices, e.g. $\mathbf{A}$, lowercase bold letters denote vectors e.g. $\mathbf{a}$. Let $\langle\cdot,\cdot\rangle$, $\|\cdot\|$ be the inner product and the l_2 Euclidean length respectively. $\mathbf{A^T}$ denote the transpose of matrix $\mathbf{A}$. The notation log refers to the base-2 logarithm. And $\lfloor r \rfloor$ denotes the largest integer not more than real number r.

2.1. Lattice

Definition 1 (Lattice)**.** *A rank-t lattice $\mathcal{L}$ in the $\mathbb{R}^n$ is spanned by t linearly independent vectors*

$$\mathcal{L} = \{\sum_{i=1}^{t} u_i \mathbf{b_i} | u_i \in \mathbb{Z}\},$$

where $\{\mathbf{b_1}, \cdots, \mathbf{b_t}\}$ is a basis for $\mathcal{L}$ and

$$\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_t})$$

is the corresponding basis matrix. The rank or dimension and determinant of $\mathcal{L}$ are respectively denoted as $\dim(\mathcal{L}) = n$ *and*

$$\det(\mathcal{L}) = \sqrt{|\det(\mathbf{BB^T})|}.$$

If $\mathbf{B}$ is a square matrix, then

$$\det(\mathcal{L}) = |\det(\mathbf{B})|.$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

In addition, when a set of column vectors $\mathbf{u} \subset \mathbb{Z}^n$ is given, the orthogonal lattice is defined as $\mathcal{L}^{\perp}(\mathbf{u}) = \{\mathbf{v} \in \mathbb{Z}^n | (\mathbf{v}, \mathbf{u}) = 0\}$.

When $t = n$, the lattice $\mathcal{L}$ is called a full rank lattice. In fact, it is only necessary to consider the full rank lattice in Euclidean space. Therefore, the lattices mentioned below are full rank lattice.

The length of the shortest non-zero vector, denoted by λ_1, is a very important parameter of a lattice. It is expressed as the radius of the smallest zero-centered ball containing a non-zero linearly independent lattice vector. Generally, successive minima was defined as follows:

Definition 2 (Successive minima). *Let $\mathcal{L}$ be a lattice of rank t. For $1 \leq i \leq t$, the i-th successive minima is defined as*

$$\lambda_i(\mathcal{L}) = inf\{r | \dim(span(\mathcal{L} \cap B_n(0,r))) \geq i\},$$

where $B_n(0,r) = \{x \in \mathbb{R}^n | \, \|x\| \leq r\}$ is a ball centered at the origin.

For the shortest vector of a random lattice, the Gaussian hypothesis is expressed as follows:
Gaussian Heuristic (First Minima). Let $\mathcal{L}$ be a full rank lattice in $\mathbb{R}^t$, then the length of the shortest non-zero vector λ_1 in $\mathcal{L}$ is estimated by

$$\lambda_1 = \sqrt{\frac{t}{2\pi e}} (\det \mathcal{L})^{1/t}. \tag{1}$$

2.2. Lattice basis reduction

Lattice reduction algorithm is employed to transform a lattice basis to another basis, and the latter one is nearly orthongral with each other and relatively shorter. As far, the LLL algorithm [1] and the BKZ algorithm [17] are well-known lattice reduction algorithms.

Definition 3 (Gram Schmidt Orthogonalization). *Given a sequence of t linearly independent vectors $\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_t}$, the Gram Schmidt orthogonalization is the sequence of vectors $\mathbf{b_1^*}, \mathbf{b_2^*}, \cdots, \mathbf{b_t^*}$ defined by*

$$\mathbf{b_i^*} = \mathbf{b_i} - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b_j^*},$$

where

$$\mu_{i,j} = \frac{(\mathbf{b_i}, \mathbf{b_j^*})}{(\mathbf{b_j^*}, \mathbf{b_j^*})}, 1 \leq j < i \leq t.$$

Definition 4 ($\delta-$LLL reduction basis). *Given a lattice basis $\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_t})$, the corresponding Gram-Schmidt basis $\mathbf{B}^* = (\mathbf{b_1^*}, \cdots, \mathbf{b_t^*})$, $\mathbf{B}$ 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 $1 \leq j < i \leq t$;*

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

Definition 5 (Geometric Series Assumption[8],GSA). *Given Gram-Schmidt basis $(\mathbf{b_1^*}, \cdots, \mathbf{b_t^*})$,*

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

for $i = 1, 2, \cdots, t$, where $0 < \theta < 1$ is called GSA constant.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

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 $\|\mathbf{b_i^*}\| \leq \|\mathbf{b_1}\|$ for $i = 1, 2, \cdots, t$.

In the subsections 3.1 and 3.2, the analysis based on the following assumptions:

Assumption 1 ([37]). Let $\mathcal{L}$ be a "random" lattice of rank t and $\mathbf{b_1}, \cdots, \mathbf{b_t}$ be an LLL-reduced basis for $\mathcal{L}$, then

$$\|\mathbf{b_1}\| \leq (1.02)^t \det(\mathcal{L})^{1/t} \tag{2}$$

and

$$\|\mathbf{b_1}\| \leq (1.04)^t \lambda_1(\mathcal{L}). \tag{3}$$

Nguyen and Stehlé [9] have studied the performance of LLL on "random" lattices and have hypothesised that an LLL-reduced basis satisfies the improved bound (2). By analogy with the relationship between the worst-case bounds $\|b_1\| < 2^{t/4} \det(\mathcal{L})/^{1/t}$ and $\|b_1\| < 2^{t/2}\lambda_1(\mathcal{L})$, it is natural to suppose that (3) is hold.

Assumption 2([37]). Let $\mathcal{L}$ be a "random" lattice of rank t and let $\mathbf{b_1}, \cdots, \mathbf{b_t}$ be an LLL-reduced basis for $\mathcal{L}$, then

$$\|\mathbf{b_i}\| < \sqrt{i}(1.02)^i \det(\mathcal{L})^{1/t}. \tag{4}$$

Nguyen and Stehlé [9] show that $\|\mathbf{b_{i+1}^*}\| \leq \|\mathbf{b_i^*}\|$ almost always in Figure 4, and certainly $\|\mathbf{b_{i+1}^*}\| \leq 1.2\|\mathbf{b_i^*}\|$ with overwhelming probability. Galbraith et al. make the heuristic assumption that $\|\mathbf{b_i^*}\| \leq \|\mathbf{b_1^*}\|$ for all $2 \leq i \leq n$, so it is easy to show that, for all $2 \leq i \leq n$, $\|\mathbf{b_i}\| \leq \sqrt{1 + (i-1)/4}\|\mathbf{b_1}\|$. So it leads to Assumption 2 for "random" lattices.

In the subsection 3.2.2, the conclusion of the following theorem will be used.

Theorem 1. *[29] Given a LLL reduction lattice basis $\mathbf{B} = (\mathbf{b_1}, \cdots, \mathbf{b_t})$, $(\mathbf{b_1^*}, \cdots, \mathbf{b_t^*})$ is the corresponding Gram-schmidt basis. The following results were hold:*

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

(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.*

After the LLL algorithm, a number of lattice reduction algorithms emerged. In practice, the Block-Korkine-Zolotarev (BKZ) algorithm proposed by Schnorr and Euchner [3] has a good performance. For the BKZ algorithm, according to [17], the block size β determines how short the output vector is. With the increase of β, the output basis becomes much reduced but the cost significantly increases. Gama and Nguyen [12] identified the Hermite factor of the reduced basis as the dominant parameter in the runtime of the lattice reduction and the quality of the reduced basis. For an t-dimensional lattice $\mathcal{L}$, the Hermite factor

$$\delta_0^t = \frac{\|b_1\|}{(\det(\mathcal{L})^{\frac{1}{t}}}, \tag{5}$$

where b_1 is the first reduced basis vector of $\mathcal{L}$ and δ_0 is called as the root-Hermite factor. Chen [27] gave an expression between the root-Hermite factor δ_0 and the block size β:

$$\delta_0 = \left(\frac{\beta}{2\pi e}(\pi e)^{\frac{1}{\beta}}\right)^{\frac{1}{2(\beta-1)}}.$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Under the GSA and based on the Hermit factor (5), Xu et al. [42] gave an upper bound on the i-th reduced basis vector

$$\|\mathbf{b_i}\| \leq \frac{\sqrt{i+3}}{2} \cdot {\delta_0}^t \cdot (\det \mathcal{L})^{\frac{1}{t}}. \tag{6}$$

3. Algorithms to solve ACD problem

In this section, the first three subsections describe and analysis the lattice-based algorithms (SDA, OL, MP) to solve the ACD problem, and the last subsection analyzes the prospects of the pre-processing technology for ACD samples, when sufficiently many samples are available.

3.1. Simultaneous Diophantine approximation (SDA)

Van Dijk et al. [16] showed that the ACD problem can be solved using the SDA method. The basic idea of this attack is to note that if $x_i = pq_i + r_i$ for $1 \leq i \leq t$, where r_i is small, then

$$\frac{x_i}{x_0} \approx \frac{q_i}{q_0}$$

for $1 \leq i \leq t$, where $x_0 = pq_0 + r_0$. That's what it means the fractions q_i/q_0 are an instance of simultaneous diophantine approximation to x_i/x_0. Once q_0 is determined, r_0 can be computed from

$$r_0 \equiv x_0 \pmod{q_0}.$$

Hence,

$$p = \frac{x_0 - r_0}{q_0}.$$

In fact, this attack does not benefit significantly from having an exact sample $x_0 = pq_0, r_0 = 0$, so such a sample can be unknown. As in [16], construct the lattice $\mathcal{L}$ of rank $t+1$, which is generated by the basis matrix $\mathbf{B}$, where

$$\mathbf{B} = \begin{pmatrix} 2^{\rho+1} & x_1 & \dots & x_t \\ & -x_0 & & \\ & & \ddots & \\ & & & -x_0 \end{pmatrix}.$$

Let $\mathbf{v} \in \mathcal{L}$, then

$$\begin{aligned} \mathbf{v} =& (q_0, q_1, \cdots, q_t)\mathbf{B} \\ =& (2^{\rho+1}q_0, q_0x_1 - q_1x_0, \cdots, q_0x_t - q_tx_0) \\ =& (2^{\rho+1}q_0, q_0r_1 - q_1r_0, \cdots, q_0r_t - q_tr_0). \end{aligned}$$

Since $q_i \approx 2^{\gamma-\eta}$, the length of the first entry of $\mathbf{v}$ is approximately $2^{\gamma-\eta+\rho+1}$. The length of the rest of entries of $\mathbf{v}$, which are of the form $q_0r_i - q_ir_0$ for $1 \leq i \leq t$, is estimated to be $|q_0r_i - q_ir_0| \leq 2|q_0r_i| \approx 2^{\gamma-\eta+\rho+1}$. Therefore, $\|\mathbf{v}\|$ is approximately

$$2^{\gamma-\eta+\rho+1}\sqrt{t+1}.$$

The vector $\|\mathbf{v}\|$ satisfying the above conditions is called the target vector.

Hence, if

$$2^{\gamma-\eta+\rho}\sqrt{t+1} < \sqrt{(t+1)/2\pi e}\,\det(\mathcal{L})^{1/(t+1)},$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

then the target vector $\mathbf{v}$ is expected to be the shortest non-zero vector in the lattice. The attack is to run a lattice basis reduction algorithm to get a candidate $\mathbf{w}$ for the shortest non-zero vector. The first entry of $\mathbf{w}$ divided by $2^{\rho+1}$ will give a candidate for q_0, then computes $r_0 \equiv x_0 \pmod{q_0}$ and $p = (x_0 - r_0)/q_0$. Finally test this value for p by checking if x_i mod p are small for all $1 \le i \le t$. This is the SDA algorithm.

A better approximation $2^{\gamma-\eta+\rho}\sqrt{t+1}$ of tager vector $\|\mathbf{v}\|$ is given by Van Dijk et al. [16], and an exact approximation

$$0.47\frac{\sqrt{(t+1)}}{p} \cdot 2^{\rho+\gamma} \tag{7}$$

is given by Galbraith et al. [37]. That is to say about twice the above approximation (taking $p \approx 2^{\eta}$).

Galbraith et al. [37] applies the Gaussian heuristic and $\lambda_2(\mathcal{L}) > 2^{t/2}\lambda_1(\mathcal{L})$ to estimate

$$\lambda_2(\mathcal{L}) \approx \sqrt{(t+1)/(2\pi e)}(\det \mathcal{L})^{1/(t+1)} \approx \sqrt{(t+1)/(2\pi e)}2^{(\rho+1+\gamma t)/(t+1)}.$$

Hence, the target vector $\mathbf{v}$ is the shortest vector in the lattice and is found by LLL if the expression (1) of Gaussian Heuristic is less than $\sqrt{(t+1)/2\pi e}\det(\mathcal{L})^{t+1}$ when multiplied by the factor 1.04^{t+1} according to Equation (3). Namely,

$$0.47\sqrt{(t+1)}(1.04)^{t+1}2^{\gamma+\rho-\eta} < \sqrt{(t+1)/(2\pi e)}2^{(\rho+1+\gamma t)/(t+1)}. \tag{8}$$

Ignoring constants and the $(1.04)^{t+1}$ term (the term $(1.04)^{t+1}$ does not have any significant effect on the performance of the algorithm[37]) in the above equation (8), a necessary (not sufficient) condition for the algorithm to succeed is

$$t+1 > \frac{\gamma-\rho}{\eta-\rho} \approx \frac{\gamma}{\eta}. \tag{9}$$

So t should be greater than $\frac{\gamma}{\eta}$ to ensure that the target vector $\mathbf{v}$ will likely be the shortest vector in the lattice $\mathcal{L}$. This lower bound on t is more advantageous for the analysis of CS-ACD [34]. For CS-ACD, if ρ is close to η, which means that for smaller γ, the dimension of the Lattice $\mathcal{L}$ grows rapidly. Concretely speaking, the parameters in CS-ACD are set to

$$(\rho, \eta, \gamma) = (\lambda, \lambda + d\log\lambda, \Omega(d^2\lambda\log\lambda)),$$

where d is the circuit depth and λ is the security paremeter. Let's say $\lambda = 200, d = 20$, set $\Omega(x) = x$, and then $(\rho, \eta, \gamma) = (200, 353, 611508)$. For these values, $t \ge 3995$. Therefore, in order to prevent lattice attacks, this ratio $(\gamma-\rho)/(\eta-\rho)$ needs to be large enough. These arguments reconfirm that the method in [34] can provide more efficient parameters for homomorphic encryption.

3.2. Orthogonal Lattice (OL) based approach

Nguyen and Stern ([6]) 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 $t-1$ 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 $t-1$ linearly independent vectors $\mathbf{u}$ orthogonal to $\mathbf{q}$, that is,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$\sum_{i=1}^{t} u_i \cdot q_i = 0 \tag{10}$$

then establish and solve indefinite equations

$$\sum_{j=1}^{t} u_{ij} \cdot x_j = \sum_{i=1}^{t} u_{ij} \cdot r_j, \tag{11}$$

where $\mathbf{u_i} = (u_{i1}, u_{i1}, \cdots, u_{it}), (i = 1, 2, \cdots, t-1)$.

Let the general solution of (11) be

$$\mathbf{d} = \mathbf{d_0} + t_1 \mathbf{d_1} \tag{12}$$

Where $\mathbf{d_0}$ is a particular solution of (11), $\mathbf{d_1}$ is the basis vector of the solution space of the corresponding homogeneous system of linear equations, and t_1 is the integer parameter.

Second, find small positive integer solutions to (11). At present, the common way to find the small solutions is to construct the lattice $\mathcal{L}$ with basis matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{d_0} \\ \mathbf{d_1} \end{pmatrix}. \tag{13}$$

Next, LLL algorithm is employed to reduce the basis matrix $\mathbf{B}$, and the first output is expected to be the vector $\mathbf{r}$. However, at present, what can meet this expectation are experimental conditions, and there is still a lack of theory. Now, the existing OL methods orthogonal to $\mathbf{q}$ are classified. According to the constructed lattice, they are divided into two categories. The first OL algorithm constructs lattice $\mathcal{L}_1(\alpha)$ with basis matrix $\mathbf{B_1}$:

$$\mathbf{B_1} = \begin{pmatrix} x_1 & \alpha & & & \\ x_2 & & \alpha & & \\ \vdots & & & \ddots & \\ x_t & & & & \alpha \end{pmatrix} \tag{14}$$

and its shape likes $\wedge$, so it is called OL-$\wedge$ algorithm. This kind of algorithm can be referred to [GGM16,XSH18].

The second OL algorithm constructs lattice $\mathcal{L}_2(\alpha)$ with basis matrix $\mathbf{B}_2$:

$$\mathbf{B}_2 = \begin{pmatrix} \alpha & & & & x_1 \\ & \alpha & & & x_2 \\ & & \ddots & & \vdots \\ & & & \alpha & x_{t-1} \\ & & & & N \end{pmatrix}, \tag{15}$$

where N is a big integer with $\gamma + \eta$ bits . Or construct a lattice $\mathcal{L}'_2(\alpha)$ with basis matrix is $\mathbf{B}'_2$:

$$\mathbf{B}'_2 = \begin{pmatrix} \alpha & & & & x_1 \\ & \alpha & & & x_2 \\ & & \ddots & & \vdots \\ & & & \alpha & x_{t-1} \\ & & & & x_t \end{pmatrix}. \tag{16}$$

Both (15) and (16) are all shaped $\vee$, so they are called OL-$\vee$ algorithm. This kind of algorithm can be referred to [[30],[31],[41],[42]].

3.2.1 OL-$\wedge$ algorithm

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

This algorithm uses the lattice $\mathcal{L}_1(\alpha)$ mentioned above, and the basis matrix is $\mathbf{B_1}$, then

$$\mathbf{B_1 B_1}^{\mathbf{T}} = \mathbf{A} + \mathbf{x}^{\mathbf{T}}\mathbf{x}, \mathbf{A} = \alpha^2 \mathbf{I_t}, \mathbf{x} = (x_1, x_2, \cdots, x_t),$$

where $\mathbf{I_t}$ is the identity matrix of order t, and

$$\det(\mathbf{B_1 B_1}^{\mathbf{T}}) = \det(1 + \mathbf{x}\mathbf{A}^{-1}\mathbf{x}) = \alpha^{2t-2}(\alpha^2 + x_1^2 + x_2^2 + \cdots + x_t^2).$$

Therefore,

$$\det(\mathcal{L}_1(\alpha)) = \alpha^{t-1}\sqrt{(\alpha^2 + x_1^2 + x_2^2 + \cdots + x_t^2)} < \sqrt{t+1} \cdot \alpha^{t-1} \cdot 2^{\gamma}.$$

(I) When $\alpha = 2^{\rho}$

Let $\mathbf{v} = (\sum_{i=1}^{t} u_i x_i, u_1 2^{\rho}, \cdots, u_t 2^{\rho}) \in \mathcal{L}_1(\alpha)$, in order to find the condition that $\mathbf{u}$ is orthogonal to $\mathbf{q}$ (t dimension), it needs to be satisfied

$$\left|\sum_{i=1}^{t} u_i x_i - \sum_{i=1}^{t} u_i r_i\right| \leq p/2, \tag{17}$$

so it forces the equation

$$\sum_{i=1}^{t} u_i x_i = \sum_{i=1}^{t} u_i r_i \tag{18}$$

to be true. In order to make the equation (18) be established, Galbraith et al. ([37]) gives the bounds of the short vectors in the lattice $\mathcal{L}_1$:

$$\|\mathbf{v}\| \leq 2^{\eta - 2 - \log(t+1)}. \tag{19}$$

Next, to show that under condition (19), formulas (17) and (10) hold, the following proof is given. Let $\|\mathbf{v}\| = N$, then $|\sum_{i=1}^{t} u_i x_i| \leq N$ and $|u_i r_i| \leq |u_i 2^{\rho}| \leq N$ for $1 \leq i \leq n$. Thus

$$\left|\sum_{i=1}^{t} u_i x_i - \sum_{i=1}^{t} u_i r_i\right| \leq \left|\sum_{i=1}^{t} u_i x_i\right| + \left|\sum_{i=1}^{t} u_i r_i\right| \leq (t+1)N.$$

Because (19) is true and $p > 2^{\eta-1}$, $(t+1)N < 2^{\eta-2} < p/2$. Hence

$$\left|\sum_{i=1}^{t} u_i x_i - \sum_{i=1}^{t} u_i r_i\right| < p/2.$$

To prove that (10) holds, suppose $|\sum_{i=1}^{t} u_i q_i| \neq 0$, so

$$2^{\eta-1} < p\left|\sum_{i=1}^{t} u_i q_i\right| = \left|\sum_{i=1}^{t} u_i (x_i - r_i)\right| \leq \left|\sum_{i=1}^{t} u_i x_i\right| + \left|\sum_{i=1}^{t} u_i r_i\right| \leq (t+1)N < 2^{\eta-2},$$

this is a contradiction.

To analyse the method, Galbraith et al. [37] use Assumption 2. This shows that LLL algorithm can be used to find $t-1$ linearly independent vectors as long as

$$\sqrt{t}(1.02)^t \det(\mathcal{L}_1(\alpha))^{1/t} \leq 2^{\eta - 2 - \log(t+1)},$$

and as well

$$\det(\mathcal{L}_1(\alpha)) = \alpha^{t-1}\sqrt{(\alpha^2 + x_1^2 + x_2^2 + \cdots + x_t^2)} \approx 2^{\rho(t-1)+\gamma}.$$

Hence, the condition for success is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$4\sqrt{t(t+1)}(1.02)^t 2^{\rho+(\gamma-\rho)/t} \leq 2^\eta.$$

Ignoring constants and the exponential approximation factor $(1.02)^t$ from the lattice reduction algorithm, Galbraith et al. [37] gives a lower bound on sample t:

$$t \geq \frac{\gamma-\rho}{\eta-\rho}. \tag{20}$$

Find $t-1$ vectors $\mathbf{u}$ that satisfies the bound of the short vector by LLL algorithm, then the system of equations can be set up to solve, and then find $\mathbf{r}$.

(II) α in the general case

Using the bound of (6), the condition that $\mathbf{u}$ is orthogonal to $\mathbf{q}$ is constructed. Specific measures are as follows([42]):

① Let $\mathbf{v} = (\sum_{i=1}^t u_i x_i, \alpha u_1, \cdots, \alpha u_t) \in \mathcal{L}_1(\alpha)$, then

$$|\sum_{i=1}^t u_i q_i| \leq \frac{\alpha+\sqrt{t}\cdot 2^\rho}{\alpha} \cdot \frac{\|\mathbf{v}\|}{2^{\eta-1}};$$

② Using BKZ-β alogrithm, reduce the lattice matrix $\mathbf{B}_1$. Let $\mathbf{v_i} = (\sum_{j=1}^t u_{ij} x_j, \alpha u_{i1}, \cdots, \alpha u_{it})$ be the i-th reduce basis of $\mathcal{L}_1(\alpha)$, Then

$$\begin{aligned}
\|\mathbf{v_i}\| &\leq \frac{\sqrt{i+3}}{2} \cdot \delta_0^t \cdot \det(\mathcal{L}_1)^{\frac{1}{t}} \\
&\leq \frac{\sqrt{i+3}}{2} \cdot \delta_0^t \cdot (\sqrt{t+1} \cdot \alpha^{t-1} \cdot 2^\gamma)^{1/t} \\
&\leq \frac{\sqrt{i+3}}{2} \cdot (t+1)^{\frac{1}{2t}} \cdot {\delta_0}^t \cdot 2^{\frac{\gamma}{t}} \cdot \alpha^{\frac{t-1}{t}}.
\end{aligned}$$

Thus

$$|\sum_{j=1}^t u_{ij} q_j| < \sqrt{i+3} \cdot (t+1)^{\frac{1}{2t}} \cdot {\delta_0}^t \cdot 2^{\frac{\gamma}{t}-\eta} \cdot \frac{\alpha+\sqrt{t}\cdot 2^\rho}{\alpha^{\frac{1}{t}}}.$$

Let

$$f(\alpha) = \frac{\alpha+\sqrt{t}\cdot 2^\rho}{\alpha^{\frac{1}{t}}},$$

then minimize $f(\alpha)$. Xu et al. [42] offer the following conclusion: When

$$\alpha_0 = \frac{\sqrt{t}}{t-1} \cdot 2^\rho,$$

$$\min_{\alpha>0} f(\alpha) = f(\alpha_0) = t(\frac{\sqrt{t}}{t-1})^{\frac{t-1}{t}} \cdot 2^{\frac{t-1}{t}\cdot\rho}.$$

Using the minimum of $f(\alpha)$, the tighter bound of $|\sum_{j=1}^t u_{ij} q_j|$ was found:

$$|\sum_{j=1}^t u_{ij} q_j| < \sqrt{i+3} \cdot g(t) \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho},$$

where

$$g(t) = t(\frac{\sqrt{t}}{t-1})^{\frac{t-1}{t}} \cdot (t+1)^{\frac{1}{2t}},$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

As analyzed by Xu et al. [42],

$$\lim_{t\to\infty} \frac{g(t)}{\sqrt{t}} = 1.$$

therefore

$$|\sum_{j=1}^{t} u_{ij}q_j| < \sqrt{(i+3)t} \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho}.$$

In order to make $\sum_{j=1}^{t} u_{ij}q_j = 0$, the right side of the upper bound has to be less than 1:

$$\sqrt{(i+3)t} \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho} < 1,$$

thus the condition

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log\sqrt{t(i+3)} < 0 \tag{21}$$

holds. The dominant calculation of OL attacks is the lattice reduction for finding $t-1$ linearly independent homogeneous equations on $r_1, \cdots, r_t$ or $q_1, \cdots, q_t$. Based on the condition (21), it is expected that

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log\sqrt{t^2+2t} < 0. \tag{22}$$

Since $\delta_0 > 1$, the attack can work when

$$t \geq \frac{\gamma-\rho}{\eta-\rho}.$$

According to (22), it is obtained that

$$\log\delta_0 < \frac{\eta-\rho}{t} - \frac{\gamma-\rho}{t^2} - \frac{\log\sqrt{t^2+2t}}{t},$$

which is equivalent

$$\log\delta_0 < -(\gamma-\rho)\left(\frac{1}{t} - \frac{\eta-\rho}{2(\gamma-\rho)}\right)^2 + \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{\log\sqrt{t^2+2t}}{t}.$$

When $t = \frac{2(\gamma-\rho)}{\eta-\rho}$, the above expression is optimized as

$$\log\delta_0 < \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{\eta-\rho}{2(\gamma-\rho)} - \frac{\eta-\rho}{4(\gamma-\rho)}\log\left(\left(\frac{\gamma-\rho}{\eta-\rho}\right)^2 + \frac{\gamma-\rho}{\eta-\rho}\right). \tag{23}$$

Also notice that when $t \geq 2$, $\log\sqrt{t^2+2t} \geq \log\sqrt{4t}$, then the logarithm term on t of condition (22) can reduce to get the following condition:

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log\sqrt{4t} < 0.$$

Similarly, taking $t = \frac{2(\gamma-\rho)}{\eta-\rho}$, the above condition is optimimized as

$$\log\delta_0 < \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{3(\eta-\rho)}{4(\gamma-\rho)} - \frac{\eta-\rho}{4(\gamma-\rho)}\log\frac{\gamma-\rho}{\eta-\rho}. \tag{24}$$

(III) Using the rounding technique, construct a deformed lattice

Construct a lattice $\mathcal{L}_1\hat{(}\alpha)$ whose lattice basis matrix is $\hat{B}_1$ [42]:

$$\hat{B_1} = \begin{pmatrix} \lfloor \frac{x_1}{\alpha} \rfloor & 1 & & \\ \lfloor \frac{x_2}{\alpha} \rfloor & & 1 & \\ \vdots & & & \ddots & \\ \lfloor \frac{x_t}{\alpha} \rfloor & & & & 1 \end{pmatrix} \tag{25}$$

where $\alpha > 0$. Similar to the idea of the case (II), the condition for $\sum_{j=1}^{t} u_{ij}q_j = 0$ was found by Xu et al. [XSH18-3.3]. The optimal value of α is:

$$\alpha_0 = \frac{\sqrt{t}}{(t-1)(\sqrt{t}+1)} \cdot 2^{\rho},$$

and the condition (21) holds as well. As discussed at the end of case (II), this attack can also be performed at

$$t \geq \frac{\gamma - \rho}{\eta - \rho}.$$

3.2.2 OL-∨ algorithm

This algorithm uses the lattice $\mathcal{L}_2(\alpha)$ mentioned above with basis matrix $\mathbf{B}_2$ or the lattice $\mathcal{L}_2'(\alpha)$ with basis matrix $\mathbf{B}_2'$.

(I) When $\alpha = 1, \mathbf{B}_2(t,t) = N$

Let

$$\mathbf{v} = (u_1, \cdots, u_{t-1}, \sum_{i=1}^{t-1} u_i x_i + u_t N) \in \mathcal{L}_2,$$

to find the condition that $\mathbf{u}$ is orthogonal to $\mathbf{q}$ ($t-1$ dimension), it needs to be satisfied:

$$\left| \sum_{i=1}^{t-1} u_i x_i \right| \leq N/2, \tag{26}$$

then it is to force the equation

$$u_t = 0. \tag{27}$$

To make the equation $u_t = 0$ true, Ding et al. [30] and Yang et al. [41] gave the bounds of the short vectors of the lattice $\mathcal{L}_2(\alpha)$, they are shown in (28) and (29) respectively,

$$\|\mathbf{v}\| \leq 2^{\eta-\rho-1-\log\sqrt{t-1}} \tag{28}$$

$$\|\mathbf{v}\| \leq 2^{\eta-\rho-2-\log\sqrt{t-1}} \tag{29}$$

In order to show that under the condition (28) or (29), formulas (26) and (27) hold, the following proof is given. Here, only condition (29) is used to prove.
Let $M = 2^{\eta-\rho-2-\log\sqrt{t-1}}$, and $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{t-1} u_i^2 + (\sum_{i=1}^{t-1} u_i x_i + u_t N)^2} < M$.
Thus

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

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

Therefore, there is no modular N operation and $u_t = 0$.

Next, it is easy to obtain that $\sum_{i=1}^{t-1} u_i q_i = 0$ and $\sum_{i=1}^{t-1} u_i x_i = \sum_{i=1}^{t-1} u_i r_i$ hold.

(II) When $\alpha = 1, \mathbf{B}_2'(t,t) = x_t$

Let

$$\mathbf{v} = (u_1, \cdots, u_{t-1}, \sum_{i=1}^{t} u_i x_i) \in \mathcal{L}_2',$$

in order to find the condition that $\mathbf{u}$ is orthogonal to $\mathbf{q}$ (t dimension), it needs to be satisfied:

$$\left|\sum_{i=1}^{t} u_i x_i\right| \leq p/2. \tag{30}$$

In order to make (30) true, Yu et al. [41] and Gebregiyorgis et al. [36] gave the bounds of the short vectors in the lattice $\mathcal{L}_2'$ and they were given by the following formulas respectively,

$$\|\mathbf{v}\| \leq 2^{\eta-\rho-2-\log\sqrt{t}}, \tag{31}$$

$$\|\mathbf{v}\| \leq 2^{\eta-\rho-3-\log t}. \tag{32}$$

After analysis, (32) is tighter than (31). Similarly, it can be proved that (30) holds under the condition (31) or(32). So the equations $\sum_{i=1}^{t-1} u_i q_i = 0$ and $\sum_{i=1}^{t} u_i x_i = \sum_{i=1}^{t} u_i r_i$ can also be obtained.

(III) $\alpha = 1$, lower bound estimating for the number of samples t

Under the GSA, Yu et al. use LLL algorithm to get the upper bound of $(t-1)$-th short vector $\mathbf{v_{t-1}}$, by Theorem 1:

$$\begin{aligned}\|\mathbf{v_{t-1}}\|^2 &\leq \alpha^{\frac{(t-2)}{2}}\|\mathbf{v}^*_{\mathbf{t-1}}\|^2 \\ &= \left(\frac{4}{3}\right)^{\frac{(t-2)}{2}}\|\mathbf{v}^*_{\mathbf{t-1}}\|^2, (\alpha = \frac{4}{3}) \\ &\leq \left(\frac{4}{3}\right)^{\frac{(t-2)}{2}}\|\mathbf{v}^*_{\mathbf{1}}\|^2 \\ &\leq \left(\frac{4}{3}\right)^{\frac{(t-2)}{2}}\left(\frac{4}{3}\right)^{\frac{(t-1)}{4}}\cdot\det(\mathcal{L}_2') \\ &= \left(\frac{4}{3}\right)^{\frac{(3t-5)}{4}}\cdot 2^{\frac{\gamma}{t}}.\end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Due to

$$\left(\frac{4}{3}\right)^{\frac{(3t-5)}{4}} \cdot 2^{\frac{\gamma}{t}} \leq 2^{\eta-\rho-2-\log\sqrt{t}},$$

the bound of t in [30] is optimized, and the optimization result is given as follows [41]:

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

Yu et al. also indicates the hypothesis in [36]

$$\lambda_{t-1}(\mathcal{L}) = \det(\mathcal{L})^{1/t}\sqrt{t/2\pi e},$$

is too strong and unreasonable, and $t > \gamma/(\eta - \rho)$ is too small, it should be increased by 10 or 20.

For OL algorithm in [31], where $-N$ or N works equally, the idea of this algorithm can be classified as OL-$\vee$, and it is equivalent to the case

$$\alpha = 1, \mathbf{B}_2(t,t) = N,$$

just N is the same as length as x x_i, so the algorithm is a little bit more conservative.

In [31], the lattices $\mathcal{L}_3$ and $\mathcal{L}_3'$ were defined, their basis matrices were $\mathbf{B}_3$ and $\mathbf{B}_3'$:

$$\mathbf{B}_3 = \begin{pmatrix} 1 & & & & x_1 \\ & 1 & & & x_2 \\ & & \ddots & & \\ & & & 1 & x_t \\ & & & & -N \end{pmatrix},$$

$$\mathbf{B}_3' = \begin{pmatrix} 1 & & & & r_1 \\ & 1 & & & r_2 \\ & & \ddots & & \\ & & & 1 & r_t \end{pmatrix}.$$

Let $\mathbf{v} \in \mathcal{L}_3 \cap \mathcal{L}_3'$, then

$$\sum_{i=1}^{t} u_i \cdot x_i - u_{t+1}N = \sum_{i=1}^{t} u_i \cdot r_i.$$

For the sake of $\sum_{i=1}^{t} u_i \cdot q_i = 0$, it is to force the equation $u_{t+1} = 0$ is true. Find a vector $\mathbf{v} = (u_1, u_2, \cdots, u_t, \sum_{i=1}^{t} u_i \cdot x_i) \in \mathcal{L}_3$, such that the corresponding vector $\mathbf{u} = (u_1, u_2, \cdots, u_t)$ is orthogonal to $\mathbf{q}$. The experiment in [31] gives the following conditions that LLL algorithm can generate $t - z$ $(z = 1, 2)$ vectors $\mathbf{u}$ (theoretically not proved):
condition 1: N is a large random integer with γ bits;
condition 2: $z \leq 2$;
condition 3: $\rho < \eta/2$;
condition 4: $t \geq (4\gamma)^{1/3}$;
condition 5: $\|\mathbf{v}\| \leq 2^{\gamma/(t+1)}$.
Under the above conditions, the equation $\sum_{i=1}^{t} u_i x_i = \sum_{i=1}^{t} u_i r_i$ is true. So $\mathbf{r}$ can be solved and p is recovered.

(IV) α in general

Similar to the idea of OL-$\wedge$ when α is in general, the condition for $\sum_{j=1}^{t} u_{ij} q_j = 0$ is found by [XSH18-3.2]. The conclusions are as follows: the optimal value of α is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$\alpha_0 = \frac{\sqrt{t^2+t}}{t-1}\cdot 2^{\rho},$$

and the condition

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log(t\sqrt{i+3}) < 0.$$

holds. The specific steps are below ([42]):
① Let $\mathbf{v} = (\alpha u_1, \cdots, \alpha u_{t-1}, \sum_{i=1}^{t} u_i x_i,) \in \mathcal{L}_2(\alpha)$, Then

$$|\sum_{i=1}^{t} u_i q_i| \leq \frac{\alpha + \sqrt{t^2+t}\cdot 2^{\rho}}{\alpha}\cdot\frac{\|\mathbf{v}\|}{2^{\eta-1}};$$

② Using BKZ-β alogrithm, reduce the basic matrix $\mathbf{B}_2$. Let $\mathbf{v_i} = (\alpha u_{i1}, \cdots, \alpha u_{i,t-1}, \sum_{j=1}^{t} u_{ij} x_j)$ be the i-th reduce basis of $\mathcal{L}_2(\alpha)$, Then

$$\|\mathbf{v_i}\| \leq \frac{\sqrt{i+3}}{2}\cdot\delta_0^t\cdot\det(\mathcal{L}_2)^{\frac{1}{t}} \leq \frac{\sqrt{i+3}}{2}\cdot\delta_0^t\cdot(\alpha^{t-1}N)^{1/t} \leq \frac{\sqrt{i+3}}{2}\cdot{\delta_0}^t\cdot 2^{\frac{\gamma}{t}}\cdot\alpha^{1-\frac{1}{t}}.$$

Thus

$$|\sum_{j=1}^{t} u_{ij} q_j| \leq \sqrt{i+3}\cdot\frac{\alpha + 2^{\rho}\sqrt{t^2+t}}{\alpha^{\frac{1}{t}}}\cdot{\delta_0}^t\cdot 2^{\frac{\gamma}{t}-\eta}.$$

A little bit of clarification here. When $\alpha = 1$, then the formula

$$|\sum_{j=1}^{t} u_{ij} q_j| \leq \sqrt{i+3}\cdot(1 + 2^{\rho}\sqrt{t^2+t})\cdot{\delta_0}^t\cdot 2^{\frac{\gamma}{t}-\eta}$$

is true. In order to make $\sum_{j=1}^{t} u_{ij} q_j = 0$, the formula

$$\sqrt{i+3}\cdot(1 + 2^{\rho}\sqrt{t^2+t})\cdot{\delta_0}^t\cdot 2^{\frac{\gamma}{t}-\eta} < 1$$

holds, then

$$\frac{\gamma}{t} - (\eta-\rho) + t\log\delta_0 + \log\sqrt{(i+3)(t^2+t)} < 0.$$

For finding $t-1$ linearly independent vectors orthogonal to $(q_1, \cdots, q_t)$, the following condition

$$\frac{\gamma}{n} - (\eta-\rho) + n\log\delta_0 + \log\sqrt{((t^3+3t^2+2t)} < 0 \tag{33}$$

is established. Next, let

$$f(\alpha) = \frac{\alpha + 2^{\rho}\sqrt{t^2+t}}{\alpha^{\frac{1}{t}}},$$

then minimize $f(\alpha)$. When

$$\alpha_0 = \frac{\sqrt{t^2+t}}{t-1}\cdot 2^{\rho},$$

$$\min_{\alpha>0} f(\alpha) = f(\alpha_0) = t\left(\frac{\sqrt{t^2+t}}{t-1}\right)^{\frac{t-1}{t}}\cdot 2^{\frac{t-1}{t}\cdot\rho}.$$

Using the minimum of $f(\alpha)$, the tighter bound of $|\sum_{j=1}^{t} u_{ij} q_j|$ was found:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$|\sum_{j=1}^{t} u_{ij}q_j| < \sqrt{i+3} \cdot g(t) \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho},$$

where

$$g(t) = t\left(\frac{\sqrt{t^2+t}}{t-1}\right)^{\frac{t-1}{t}}.$$

As analyzed by Xu et al. [42],

$$\lim_{t\to\infty} \frac{g(t)}{t} = 1,$$

therefore

$$|\sum_{j=1}^{t} u_{ij}q_j| < \sqrt{(i+3)} \cdot t \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho}.$$

In order to make $\sum_{j=1}^{t} u_{ij}q_j = 0$, the right side of the upper bound has to be less than 1:

$$\sqrt{(i+3)} \cdot t \cdot \delta_0^t \cdot 2^{\frac{\gamma-\rho}{t}-\eta+\rho} < 1,$$

thus the condition

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log(t\sqrt{i+3}) < 0 \tag{34}$$

holds. The dominant calculation of OL attacks is the lattice reduction for finding $t-1$ linearly independent homogeneous equations on $r_1, \cdots, r_t$ or $q_1, \cdots, q_t$. Based on the condition (34), it is expected that

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log\sqrt{t^3+2t^2} < 0 \tag{35}$$

Since $\delta_0 > 1$, the attack can work when

$$t \geq \frac{\gamma-\rho}{\eta-\rho}.$$

According to (35), It is obtained that

$$\log\delta_0 < \frac{\eta-\rho}{t} - \frac{\gamma-\rho}{t^2} - \frac{\log\sqrt{t^3+2t^2}}{t},$$

which is equivalent

$$\log\delta_0 < -(\gamma-\rho)\left(\frac{1}{t} - \frac{\eta-\rho}{2(\gamma-\rho)}\right)^2 + \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{\log\sqrt{t^3+2t^2}}{t}.$$

When $t = \frac{2(\gamma-\rho)}{\eta-\rho}$, the above expression is optimized as

$$\log\delta_0 < \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{\eta-\rho}{2(\gamma-\rho)}\log\frac{2(\gamma-\rho)}{\eta-\rho} - \frac{\eta-\rho}{4(\gamma-\rho)}\log\left(\frac{2(\gamma-\rho)}{\eta-\rho}+2\right) \tag{36}$$

Also notice that when $t \geq 2$, $\log\sqrt{t^3+2t^2} \geq \log(2t)$, then the logarithm term on t of condition (36) can reduce to get the following condition:

$$\frac{\gamma-\rho}{t} - (\eta-\rho) + t\log\delta_0 + \log(2t) < 0.$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Similarly, taking $t = \frac{2(\gamma-\rho)}{(\eta-\rho)}$, this condition is optimized as

$$\log \delta_0 < \frac{(\eta-\rho)^2}{4(\gamma-\rho)} - \frac{(\eta-\rho)}{2(\gamma-\rho)} - \frac{\eta-\rho}{2(\gamma-\rho)} \log \frac{2(\gamma-\rho)}{\eta-\rho}. \tag{37}$$

3.2.3 Recover r or q

(1) Recover r by LLL algorithm

Let the general solution formula of (11) be

$$\mathbf{d} = \mathbf{d_0} + t_1\mathbf{d_1} + \cdots + \mathbf{d_z}, \tag{38}$$

where $\mathbf{d_0}$ is a particular solution of (11), $\mathbf{d_1}, \cdots, \mathbf{d_z}$ are the basis vectors for the solution space of the corresponding homogeneous system of linear equations, and $t_1, \cdots, t_z$ are the integer parameters.

Next, find small positive integer solutions to (11) to get $\mathbf{r}$. Constract the lattice $\mathcal{L}$ with basis matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{d_0} \\ \mathbf{d_1} \\ \cdots \\ \mathbf{d_z} \end{pmatrix}. \tag{39}$$

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

$$\mathbf{d}' = k_0\mathbf{d_0} + k_1\mathbf{d_1} + \cdots + k_z\mathbf{d_z} \tag{40}$$

where $k_0, k_1, \cdots, k_z$ are integers. Obviously, when $k_0 = 1$, (40) = (38). Reduce the lattice $\mathbf{B}$ to $\mathbf{B}'$:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{d'_0} \\ \mathbf{d'_1} \\ \vdots \\ \mathbf{d'_z} \end{pmatrix}. \tag{41}$$

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 (11).

Let $\mathbf{d'_i}$ is the solution to (11), 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 [31] found the conditions that the algorithm can work well (see 3.2.2). In addition, if $\mathbf{d'_i} \neq \mathbf{d'_0}$, we find an interesting thing that the recovery value p' is only 1 or 2 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

$$p' - p = d_{ru}, \tag{42}$$

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.

(2) Recover q

Let $\mathbf{U} = (u_{ij})_{t\times t}$, which satisfies $\mathrm{LLL}(\mathbf{B}) = \mathbf{UB}$, where $\mathbf{B}$, $\mathrm{LLL}(\mathbf{B})$ are the lattice basis matrix and LLL reduced lattice basis matrix respectively, then $\mathbf{U}$ is a unimodular matrix with $|\mathbf{U}| = \pm 1$. Constract the system

$$\mathbf{U}\mathbf{q}^{\mathbf{T}} = (0, 0, \cdots, 0, d)^{\mathbf{T}}.$$

Because $(q_1, q_2, \cdots, q_t) = 1$ with probability $1/\zeta(t)$, where $\zeta(t) = \sum_{i=1}^{\infty} 1/k^t$ is the function of Euler-Rieamann zeta, the probability that $d = \pm 1$ is very high. Therefore $(q_1, q_2, \cdots, q_t)$ is the absolute value of the last column of $\mathbf{U}^{-1}$ and $p = \lfloor x_i/q_i \rfloor$ [42]. It can be known from OL algorithm, the first $t-1$ row vector of matrix $\mathbf{U}$ can be generated by LLL algorithm, but the t-th row vector $\mathbf{u}_t = (u_{t1}, u_{t2}, \cdots, u_{tt})$ has to satisfy the following equation:

$$u_{t1}q_1 + u_{t2}q_2 + \cdots + u_{tt}q_t = \pm 1. \tag{43}$$

If (43) is considered in isolation, it is very possible for (43) to be established. But from the above analysis, it can be seen that the matrix $\mathbf{U}$ is a transition matrix from a basis of a lattice to its reduced basis, and $\mathbf{U}$ is a unimodular matrix. Furthermore, through our experiments, it is difficult to guarantee that the last row vector $\mathbf{u_t}$ of $\mathbf{U}$ satisfies the equation (43). So finding such a matrix $\mathbf{U}$ is still an open question.

3.2.3 An improved algorithm of OL-$\vee$

In this part, an improved algorithm of OL-$\vee$ is proposed. Constract a lattice $\mathcal{L}$ with the basis matrix

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

where N is an integer with $\gamma + \eta$ bits. Using the lower bound of t in [41] and the upper bound of the short vector in [36], the following improved Algorithm 1 is given.

Algorithm 1 An improved OL algorithm for GACD

Input: An appropriate positive integer $t = \lfloor \frac{5}{3}(\eta - \rho - \sqrt{(\eta-\rho)^2 - 1.2\gamma}) \rfloor$, and ACD samples $x_1, \cdots, x_t$.

Output: Integer p.

1. Randomly choose $N \in (2^{\gamma+\eta-1}, 2^{\gamma+\eta})$.

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

3. If $\|\mathbf{v_i}\| < 2^{\eta-\rho-3-\log t}, (i = 1, \cdots, t-z)$, where $z = 1, 2$, then solve the integer linear system with t unknowns $r_1, \cdots, r_t$ as follows

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

Therefore, the integer solutions can be expressed as follow:

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

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

4. $\mathbf{d_0} = \mathbf{r}$.

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

Return p.

When

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$\eta \geq \rho + 1.1\sqrt{\gamma}, \tag{44}$$

this algorithm can successfully recover p. It is an improvement of Ding and tao's OL algorithm [31]. Firstly, the lower bound of t and the upper bound of the short vector $\mathbf{v}$ are modified. Secondly, the later step using the LLL algorithm has been cancelled in the recovery of $\mathbf{r}$. This is because when the algorithm is implemented with isolve command of Maple, the special solution of the equations (11) is exactly the small positive integer solution under the condition (44). Thirdly, unlike Ding and tao's OL algorithm [31] which is not proved theoretically, our algorithm is correct theoretically. And it can be seen that the attack range is extended greatly and the efficiency increases quickly.

3.3. Multivariate polynomial (MP) equations method

Howgrave-Graham ([5]) is the first to consider reducing the PACD problem by giving two ACD sample inputs, $N = pq_0$ and $a = pq_1 + r_1$. The idea is based on finding small roots of modular univariate linear equations of the form $a + x \equiv 0 \pmod{p}$ for unknown p. It is generalized to a multivariate version in [16] which is called MP method. In fact, MP method is an extension of Coppersmith's method ([Cop96b]). A rigorous analysis of this algorithm was provided by Cohn and Heninger [25] and a variant for the case when the "errors" are not all the same size was given by Takayasu and Kunihiro [28]. It is well-known that MP approach has some advantages if the number of ACD samples is very small, but the application with a large number of samples in actual cryptanalysis needs a great deal of attention. In addition, the MP approach can be applied to both PACD and GACD problems, but it is simpler to explain and analyse the PACD problem. Hence, in the following discussion, only this case will be told.

Notice that some notations change here. let $N = pq_0$ and let $a_i = pq_i + r_i$ for $1 \leq i \leq m$ be our ACD samples, where $|r_i| \leq R$ for some given bound R. The idea is to construct a polynomial $Q(X_1, X_2, \cdots, X_m)$ in m variables such that $Q(r_1, \cdots, r_m) \equiv 0 \pmod{p^k}$ for some k. The parameters m and k are to be optimized. In [25], such a multivariate polynomial is constructed as integer linear combinations of the products $(X_1 - a_1)^{i_1}(X_1 - a_m)^{i_m}N^l$ where l is chosen such that $i_1 + \cdots + i_m + l \geq k$.

Let

$$f_{[i_1,i_2,\cdots,i_m](X_1,X_2,\cdots,X_m)} = (RX_1 - a_1)^{i_1}(RX_2 - a_2)^{i_2}\cdots(RX_m - a_m)^{i_m}N^l. \tag{45}$$

Here, the bound of the polynomial degree t has to be chosen. It doesn't do any good to talk about $k > t$, because it leads to the entire matrix being multiplied by the scalar N^{k-t} for the case $t = k$. Accordingly, Cohn and Heninge [25] consider the lattice $\mathcal{L}$ generated by the coefficient row vectors of (45) such that $i_1 + \cdots + i_m \leq t$ and $l = max(k - \sum_j i_j, 0)$. If the occurrence of monomial in $f[i_1, i_2, \cdots, i_j]$ is sorted in inverse lexicographical order, then the basis matrix for the lattice $\mathcal{L}$ is lower triangular. For example, when $(t, m, k) = (3, 2, 1)$, the corresponding basis matrix is $\mathbf{B}$:

$$\mathbf{B} = \begin{array}{c|ccccccccc} f[i_1,i_2] & 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & \cdots & x_2^3 \\ \hline f[0,0] & N & & & & & & & \\ f[1,0] & -a_1 & R & & & & & & \\ f[0,1] & -a_2 & & R & & & & & \\ f[2,0] & a_1^2 & -2a_1R & & R^2 & & & & \\ f[1,1] & a_1a_2 & -a_2R & -a_1R & & R^2 & & & \\ f[0,2] & a_2^2 & & -2a_2R & & & R^2 & & \\ \vdots & \vdots & \vdots & \vdots & & & & \ddots & \\ f[0,3] & -a_2^3 & & 3a_2^2R & & & -3a_2^2R^2 & & R^3 \end{array}.$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

It is shown that the dimension of the lattice $\mathcal{L}$ is $\dim(\mathcal{L}) = d = \binom{t+m}{m}$, and its determinant is

$$\det(\mathcal{L}) = R^{d\frac{mt}{m+1}} N^{\binom{k+m}{m}\frac{k}{m+1}} = 2^{d\frac{\rho mt}{m+1} + \binom{k+m}{m}\frac{\gamma k}{m+1}},$$

where $R = 2^{\rho}$, $N = 2^{\gamma}$. The following is a brief proof of $\dim(\mathcal{L})$ and $\det(\mathcal{L})$. Clearly, $\dim(\mathcal{L})$ is the number of possible polynomials of the form

$$(RX_1 - a_1)^{i_1}(RX_2 - a_2)^{i_2} \cdots (RX_m - a_m)^{i_m} N^l$$

in the variables $(X_1, X_2, \cdots, X_m)$. So, count the possible number of combinations of the exponents i_j, where $i_j \geq 0$, so that $0 \leq i_1 + \cdots + i_m \leq t$. Assigning $t+1$ values $(0, 1, 2, \cdots, t)$ to m exponents i_j can be denoted by $\binom{t+m}{m}$. Equivalently, count the number of non-negative integer solutions to the equation $i_1 + \cdots + i_m = t'$ in the variables i_j. It has $\binom{t'+m-1}{t'}$ possible number of solutions. Note that since

$$\binom{t+m}{m} = \binom{t+m-1}{t} + \binom{t+m-1}{t-1},$$

and

$$\binom{m-1}{0} + \binom{m}{1} + \cdots + \binom{t+m-1}{t} = \binom{t+m}{m}.$$

Adding all possible number of solutions for $0 \leq t' \leq t$ gives the result

$$\dim(\mathcal{L}) = \binom{t+m}{m}.$$

Next, $\det(\mathcal{L}) = N^{S_N} R^{mS_R}$, where S_N is the sum of exponents of N and S_R is the sum of exponents of R. Because there are in total $\binom{t+m}{m}$ monomials with $\binom{m+i-1}{i}$ of them having exponent i. This implies that R has exponent $i\binom{m+i-1}{i}/m = \binom{m+i-1}{i-1}$. Summing up for $1 \leq i \leq t$ gives the total exponent of R. So

$$S_R = \binom{m}{0} + \binom{m+1}{1} + \cdots + \binom{t+m-1}{t-1} = \binom{t+m}{m}\frac{\binom{t+m}{t-1}}{\binom{t+m}{m}} = \binom{t+m}{m}\frac{t}{m+1}.$$

The exponent of N in each monomial expression is l, where $l = max(k - \sum_j i_j, 0)$. A similar analysis gives the exponent of N to be

$$S_N = \binom{k+m}{m}\frac{k}{m+1}.$$

Substituting N and R by their size estimates 2^{γ} and 2^{ρ} respectively gives the result

$$\det(\mathcal{L}) = R^{d\frac{mt}{m+1}} N^{\binom{k+m}{m}\frac{k}{m+1}} = 2^{d\frac{\rho mt}{m+1} + \binom{k+m}{m}\frac{\gamma k}{m+1}}.$$

Let $\mathbf{v}$ be a vecor in $\mathcal{L}$ and

$$Q(X_1, \cdots, X_m) = \sum_{i_1, i_2, \cdots, i_m} (Q_{i_{1m}} x_{1\ m}^{i_1 i_m}),$$

then

$$\mathbf{v} = \sum_{i_1, i_2, \cdots, i_m} (Q_{i_{1m}} R^{i_1 + \cdots + i_m}).$$

If $|Q(r_1, \cdots, r_m)| < p^k$, clearly the equation $Q(r_1, \cdots, r_m) = 0$ holds over the integers. So the norm of $|Q(r_1, \cdots, r_m)|$ needs to be bounded. Notice that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

$$\begin{aligned}|Q(r_1,\cdots,r_m)| &\le \sum_{i_1,i_2,\cdots,i_m} |r_1|^{i_1}+\cdots+|r_m|^{i_m}\\ &\le \sum_{i_1,i_2,\cdots,i_m} R^{i_1}+\cdots+R^{i_m}\\ &= \|\mathbf{v}\|_1.\end{aligned}$$

where the norm $\|\mathbf{v}\|_1$ represents the sum of the absolute values of the components for the vector $\mathbf{v}$. Hence, if $\|\mathbf{v}\|_1 < p^k$ for some k, then $\mathbf{v} \in \mathcal{L}$ is the target vector found in the MP algorithm. In order to save time and memory, more than m algebraically independent target vectors are usually selected for elimination. By using Gröbner basis method or the existing corresponding results to reduce the system to a univariate polynomial equation and hence solve for $(r_1,\cdots,r_m)$. Then $p = \gcd(N, a_1 - r_1)$ can be determined.

When $(t,k) = (1,1)$, the MP algorithm is the same as the orthogonal lattice attack [DGHV10, CH13, GGM16]. Such parameters $(t,k) = (1,1)$ are called "unoptimised". The question is whether the algorithm is better at $t > 1$.

3.3.1 The heuristic analysis results of the MP algorithm in [25]

Cohn and Heninger [25] give a heuristic theoretical analysis of the MP algorithm and suggest optimal parameter choices (t,m,k). Later, Galbraith et al. sketched CH approach in [37]. The main results of the analysis are presented here.

Result 1. Let $\beta = \eta/\gamma \ll 1$, so that $p \approx N^\beta$, then the parameters (t,m,k) satisfy the relational expression

$$\frac{mt\rho}{(m+1)k} + \frac{\gamma k^m}{(m+1)t^m} < \beta\gamma = \eta.$$

Proof. Because the MP algorithm is executed successfully, m vectors satisfying $\|\mathbf{v}\|_1 < p^k$ need to be generated. Using $\|\mathbf{v}\|_1 < \sqrt{d}\|\mathbf{v}\|$ and the bounds from Assumption 2, the LLL-reduced basis satisfying the condition $\|\mathbf{b_i}\|_1 \le d(1.02)^d(\det\mathcal{L})^{1/d}$ can be found, where d is the dimension of the lattice. If this bound is less than $p^k \approx 2^{\eta k}$, then enough vectors are needed to be obtained. Hence,

$$d^d(1.02)^d\det(\mathcal{L}) < 2^{\eta k d},$$

and so

$$d\log d + d^d\log 1.02 + d\rho\frac{mt}{m+1} + \gamma\binom{k+m}{m}\frac{k}{m+1} < k\eta d. \tag{46}$$

Let $\beta = \eta/\gamma \ll 1$, so that $p \approx N^\beta$. With the first two terms of (46) deleted and approximating $\binom{k+m}{m} \approx k^m$, $d = \binom{t+m}{m} \approx t^m$, the formula (46) can be reduced to

$$\frac{mt\rho}{(m+1)k} + \frac{\gamma k^m}{(m+1)t^m} < \beta\gamma = \eta. \tag{47}$$

Result 2 (Heuristic). For fixed m, if $\eta^2 \gg \gamma$ and $\rho = \log R < \eta(1+o(1))\beta^{1/m}$, then the ACD problem can be solved in polynomial time.

Remark 1. *Result 2 does not imply that the MP approch is better than the SDA or OL approches. When ρ is small, all algorithms based on lattices of dimension approximately γ/η, and the lattice input size is proportional to γ, so they are all polynomial time if they return a correct solution to the problem.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

3.3.2 The heuristic analysis results of the MP algorithm in [36]

Gebregiyorgis [36] solved the corresponding polynomial equations in m variables. The following conclusion was obtained.

Result 3. Under the hypothesis

$$\lambda_m(\mathcal{L}) = (\det \mathcal{L})^{\frac{1}{d}} \sqrt{\frac{d}{2\pi e}},$$

where $d = \binom{t+m}{m}$, $\lambda_m(\mathcal{L})$ is short if $d > \dfrac{\binom{k+m}{m}\gamma}{(m+1)(\eta-\rho)}$.

Proof. If $\lambda_m(\mathcal{L}) = (\det \mathcal{L})^{\frac{1}{d}} \sqrt{\frac{d}{2\pi e}}$, then $\lambda_m(\mathcal{L}) < p^k$ which implies

$$\log(\det \mathcal{L}) < dk\eta.$$

So

$$d\rho \frac{mt}{m+1} + \gamma \binom{k+m}{m} \frac{k}{m+1} < kd\eta,$$

which is implied by

$$\gamma \binom{k+m}{m} \frac{1}{m+1} < d(\eta - \rho(t/k)).$$

This is equivalent to

$$d = \binom{t+m}{m} > \frac{\binom{k+m}{m}\gamma}{(m+1)(\eta - \rho(t/k))} \approx \frac{\binom{k+m}{m}\gamma}{(m+1)(\eta-\rho)}.$$

For $d > \frac{\binom{k+m}{m}\gamma}{(m+1)(\eta-\rho)}$, the first m output vectors $\mathbf{v_i}$ of the LLL algorithm satisfy $\|\mathbf{v_i}\|_1 < p^k$ giving us polynomial relations between $r_1, \cdots, r_m$. Let $\mathbf{v} = (u_1, \cdots, u_d)$ and consider the d monomials $(1, X_1, X_2, X_1^2, X_1X_2, X_2^2, \cdots, X_m^t)$ in degree reverse ordering. Then the corresponding polynomial to lattice vector $\mathbf{v}$ is

$$Q(X_1, X_2, \cdots, X_m) = \sum_{i=1}^{d} \frac{u_i}{R^{j_1+j_2+\cdots+j_m}} X^{j_1\, j_m}_{1\;\, m}.$$

Next, collect m such independent polynomial equations. The system of equations can be solved using the Gröbner basis algorithms to find $r_1, \cdots, r_m \in \mathbb{Z}$. Note that the first m output of the LLL algorithm do not necessarily give an algebraic independent vectors. In this case, subsequent vectors generated by the LLL algorithm with l_1 norm less than p^k (if there are any) need to be added. Alternatively, to get algebraic independent polynomial equations, the polynomial equations needs to be factored. Finally, p is recovered with a high probability by computing $\gcd(N, a_1 - r_1, a_2 - r_2, \cdots, a_m - r_m)$.

The drawback of the CH-approach is that enough independent polynomial equations cannot be discovered. Gebregiyorgis's experiment shows that this is the case. Moreover, the running of the Gröbner basis part is stuck even for small parameters.

3.3.3 The heuristic analysis results of the MP algorithm in [37]

Galbraith et al. [37] analyzed the conclusions of [25] and considered the parameters more generally, where it was assumed that the optimal solution would be to take $t, k > 1$. Here are the main results.

Result 4. The condition $\eta^2 \gg \gamma$ in Result 2 means the MP attack can be avoided in practice relatively easily.

Remark 2. *The OL method does not have any such hard limit on its theoretical feasibility. However, in practice the restriction $\eta^2 \gg \gamma$ is not so different from the usual condition that the dimension must be at least γ/η. If $\gamma > \eta^2$, then the required dimension would be at least η, which is infeasible for lattice reduction algorithms for the parameters used in practice.*

Result 5. For CS-ACD parameters, Galbraith et al. [37] suppose $\rho \approx \eta$ (e.g., $\rho/\eta = 0.9$) and $\gamma = \eta^{1+\delta}$ for some $\delta > 0$, their experimental condition $t\rho < k\eta$ implies that $t \approx k$ in which case $\binom{k+m}{m} \approx d = \binom{t+m}{m} > \frac{\gamma}{\eta}\binom{k+m}{m}\frac{1}{m+1}$. So this bound suggested that MP approach has no advantage over other methods for parameters of CS-ACD. Their experimental results also confirm this.
Result 6. When m is large, the best choices for the MP algorithm are $(t,k) = (1,1)$, and so MP method was not better than the SDA or OL methods by practical experiments.

3.4. Pre-processing of the ACD samples

The most important factor in the hardness of the ACD problem is the ratio γ/η, which is the size of the integers x_i relative to the size of p. The main idea of pre-processing method is: for the same p, without changing the size of the errors r_i, reduce γ and find an easier set of ACD instances.

The method of preserving the sample size was analysed briefly in [36] and further discussion about preserving and aggressive shortening the sample size was given in [37]. Here is a brief overview and the statistic analysis of Galbraith's results [37] on the pre-processing method.

The main idea of the pre-processing is the step by taking differences $x_k - x_i$ for $x_k > x_i$ and $x_k \approx x_i (1 \le i \le \tau)$. The essence is that if $x_k \approx x_i$ then $q_k \approx q_i$ but r_k and r_i are not affected at not. Hence, $x_k - x_i = p(q_k - q_i) + (r_k - r_i)$ is an ACD sample for the same p but with a smaller q and a similar sized error r. It is natural to want to be able to iterate this process until the sample size is suitable for the OL attack.

3.4.1 Preserving the sample size

Let the original samples $x_i = pq_i + r_i, |r_i| \le 2^\rho$, and the samples at iteration k are of the form

$$x = \sum_{i=1}^{2^k} c_i x_i, c_i = \pm 1,$$

so the error terms is a "random" sum of 2^k ρ-bit integers:

$$r = \sum_{i=1}^{2^k} c_i r_i, c_i = \pm 1.$$

Since the $r_i \in [-2^\rho, 2^\rho]$ are uniformly distributed, for large k,

$$\mathbf{E}(r) = 0, \mathbf{Var}(r) = \frac{1}{3}2^{2\rho+k}.$$

So the condition of $|r| \le 2^{2^{\rho+k/2}}$ is expected. The analysis results in [37] are as follows:
Result 7. An absolute upper limit on the number of iterations is $2(\eta - \rho)$, and after the final iteration, the samples are reduced to bitlength no fewer than $\gamma - 2b(\eta - \rho)$ bits.

Remark 3. *Let $x_1, x_2, \cdots, x_\tau$ be the intial γ-bit ACD samples. Suppose $B = 2^b$, b is typically 8 or 16. After I iterations, approximately $\tau - IB$ samples are generated, each of $\gamma - Ib$ bits. If $\rho + k/2 > \eta$, then the errors have grown so large that all information about p is lost essentially. Hence, an absolute upper limit on the number of iterations is $2(\eta - \rho)$). This means that after the final iteration the samples are reduced to bitlength no fewer than $\gamma - 2b(\eta - \rho)$ bits.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Result 8. The pre-processing approach can make very little effect on the ACD problem.

Remark 4. *An attack on the original ACD problem requires a lattice of dimension roughly γ/η (assuming $\rho \ll \eta$). After k iterations of pre-processing, a lattice of dimension $\frac{\gamma-bk}{\eta-(\rho+k/2)}$ was needed. Even in the best case when taking $k = 2(\eta-\rho)$ and keep the denominator constant at $(\eta-\rho)$, the lattice dimension is lowered from γ/η to $(\gamma/\eta) - 2b$. Since $b = 8$ or 16, the lattice dimension decreased very little.*

3.4.2 Aggressive shortening

The idea of the aggressive shortening method in [37] is to generate new samples (that are still about the same bitlength) by taking sums/differences of the initial list of samples. The steps consist of the following four steps:
Step 1. Let $\mathbb{S} = (x_1, \cdots, x_\tau)$ be a set of ACD samples, with $x_k = pq_k + r_k$ having mean and standard deviation given respectively by

$$\mu = \mathbf{E}(x_k) = 2^{\gamma-1} \quad \text{and} \quad \sigma_0 = \sqrt{\frac{1}{3}2^{2(\gamma-1)}(1+2^{-2(\gamma-\rho-1)})} \approx \frac{1}{\sqrt{3}}2^{\gamma-1}.$$

Let

$$S_k = \sum_{i=1}^{l} x_{k_i}, [k = 1, \cdots, m],$$

that is to say, the m random sums $S_1, \cdots, S_m$ of l elements of $\mathbb{S}$ were generated. So

$$\mathbf{E}(S_k) = l \cdot 2^{\gamma-1}, \mathbf{Var}(S_k) = \frac{l}{3}2^{2(\gamma-1)}(1+2^{-2(\gamma-\rho-1)}).$$

Step 2. Sort the new samples $S_1, \cdots, S_m$ to obtain the list $S(1) \leq \cdots \leq S(m)$. These are called order statistics and are represented by $S(k)$.
Step 3. Consider the neighbouring differences or spacings $T_k = S(k+1) - S(k)$ for $k = 1, \cdots, m-1$, and derive the statistical distribution of the spacings.
Step 4. Store the $\tau = m/2$ spacings as input to the next iteration of the algorithm. After I iterations, OL attacks can be applied.

The following analysis results were presented in [37]:
Result 9. The total number of iterations performed satisfies $I < \eta$.

The complexity is proportional to $Imlog(m)$, since each iteration computes a sorted list of size m. The mean and the standard deviation of the spacings is inversely proportional to m, so m is expected to be very large. Suppose, at the j-th iteration, a list of τ_{j-1} values are $Y_1^{(j-1)}, Y_2^{(j-1)}, \cdots, Y_{\tau_{j-1}}^{(j-1)}$ (so $\tau_0 = \tau$) with standard deviation σ_{j-1}. The statistical distribution of such generic spacings have Exponential distributions. Suppose $Z_1, \cdots, Z_m$ are independent and identically distributed random variables on $\mathbb{R}$ with common distribution function F, inverse distribution function F^{-1} and density function $f = F'$. If $Z(1) \leq \cdots \leq Z(m)$ denote the order statistics of $Z_1, \cdots, Z_m$, then the k-th spacing $Z(k+1) - Z(k)$ is well-approximated for large m as an Exponential random variable with (rate) parameter $mf(F^{-1}(\frac{k}{m})$ [37]. In particular, Suppose $Z_1, \cdots, Z_m \sim N(\mu, \sigma^2)$ are normally distributed with mean μ and variance σ^2, then $f(F^{-1}(u)) = \frac{g(G^{-1}(u))}{\sigma}$, where G^{-1} and g are respectively the inverse distribution function and density function of a standard Normal $N(0,1)$ random variable. Let $H(u)$ denote the function $g(G^{-1}(u))^{-1}$, Galbraith et al. [37] have been graphed and analyzed that H is a moderately small value away from the extreme order statistics, for example $H(u) \approx 4$ for $0.2 < u < 0.8$. Thus the spacings have an Exponential distribution (with parameter depending on k) given by $Z_{k+1} - Z_k \sim Exp\left(\frac{m}{\sigma H(\frac{k}{m})}\right)$ with mean $\frac{\sigma H(\frac{k}{m})}{m}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Remark 5. *As noted in [37] that a random sum S_k is well-approximated as a Normal random variable with variance $l\sigma_{j-1}^2$ for $l > 1$. The k-th spacing in this Normal approximation case essentially has a distribution given by $S(k+1) - S(k) \sim Exp\left(\frac{m}{\sqrt{l}\sigma_{j-1}H(\frac{k}{m})}\right)$ with mean $\frac{\sqrt{l}H(\frac{k}{m})}{m}\sigma_{j-1}$. $H(\frac{k}{m}) \approx 4$ when $0.2m \leq k \leq 0.8m$, so by considering the "middle" spacings of $T_1, \cdots, T_{m-1}$, $\tau_j = m/2$ random variables can be obtained with approximately the same distribution that are in general independent. Thus at the end of the j-th iteration, random variables $Y_1^{(j)}, Y_2^{(j)}, \cdots, Y_{\tau_j}^{(j)}$ is obtained, with mean and standrad deviation $\sigma_j = \frac{4\sqrt{l}}{m}\sigma_{j-1}$. After j iterations, the random variables γ are sums of $(2l)^j$ of the original ACD samples, so the standard deviation of an error term in the output of the j-th has increased by a multiple of $(2l)^{\frac{j}{2}}$. Hence, the total number of iterations performed satisfies $I < \eta$.*

Result 10. To have samples of size close to η-bits thus required $\eta \approx i \cdot \log(4\sqrt{l}/m) + \gamma - 1$. Optimistically taking $i = \eta$, the number of new samples m should satisfy: $\log m \approx \frac{\gamma-1}{\eta} + \log\sqrt{l} + 1$. In other words, m is close to $2^{\gamma/\eta}$.

Remark 6. *After i iterations, the average size of samples is $(4\sqrt{l}/m)^i 2^{\gamma-1}$.*

Result 11. In practice, m was prohibitively large. For the parameter sizes required for a cryptographic system, the resulting errors grew too rapidly to be useful for the neighbouring difference.

4. Comparisons of OL with SDA and MP algorithms

The ACD problem is currently a hard problem for appropriate parameter settings. Some cryptographic applications exploited the hardness problem of ACD. The homomorphic encryption schemes over the integers are particular examples, such as [16,19,34,35,38]. The security analysis of these schemes was based on the complexity of different algorithms to analyze and solve the ACD calculation problem. These algorithms were in turn based on the worst-case performance of the lattice reduction algorithm. It is important to analyze the current most effective algorithm to solve the ACD problem from practical point of view.

4.1 Comparision with the SDA algorithm

The SDA-approach (see Section 3.1) solves the ACD problem using simultaneous diophantine approximation method. The dimension of the lattice required is greater than $(\gamma - \rho)/(\eta - \rho)$. As if the proportion of these parameters is too large, the LLL algorithm cannot produce the desired output.

Van Dijk et al. [16] and Galbraith etal. [36] point out that the SDA algorithm is comparable to the performance when $\alpha = 2^\rho$ in the OL-$\wedge$ approach. Hence, the OL-$\wedge$ attack using the rounding technique is the fastest since it employs the input basis matrix with smaller entries, especially when $(\gamma - \rho)$ is small. This fact is confirmed by experiments of Xu et al. [42].

4.2 Comparison of the two types of OL attacks

At first, the following asymptotic complexity estimatioans are given. Then according to operating conditions that depend on δ_0, the corrsponding comparion is presented.

Theorem 2. *([42]) The time complexity for solving (γ, η, ρ)-ACD instances is*

$$2^{\Omega\left(\frac{\gamma-\rho}{(\eta-\rho)^2}\log\frac{\gamma-\rho}{(\eta-\rho)^2}\right)}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

by running BKZ-β to achieve a root-Hermite factor δ_0 such that (24) holds if one SVP oracle costs $2^{O(\beta)}$.

Theorem 3. *([42]) For given (γ, η, ρ)-ACD instances and some sufficiently large security parameter λ, if the condition*

$$\gamma \geq \Omega\left(\frac{\lambda}{\log \lambda}(\eta - \rho)^2\right) + \rho$$

holds, then the time complexity for solving (γ, η, ρ)-ACD instances is 2^{λ} by running BKZ-β if one SVP oracle costs $2^{O(\beta)}$.

Comparision of the OL-$\wedge$ attacks According to the analysis in Section 4.1 of [42], OL-$\wedge$ attacks in Section 3.2.1, cases (II) and (III) have the same asymptotic time complexity. Notice that in OL-$\wedge$ attack, the entries of input basis matrix in case (III) are approximately reduced by ρ bits compared to that in case (II). Hence, the OL-$\wedge$ attack in case (III) will be faster in practical cryptanalysis. In typical scenarios, the OL-$\wedge$ attack in case (III) only achieves a constant improvement of the overall attack complexity. Based on the time complexity in the paper [20], the acceleration caused by reducing the number of bits ρ is $1 - (\rho/\gamma)$. This improvement may be quite significant in practice.
Comparision with the two types of OL attacks When $\gamma \gg \rho$, all these OL attacks for solving the (γ, η, ρ)-ACD problem have the same asymptotic time complexities; the OL-$\wedge$ attack is more advantageous than the OL-$\vee$ attack when $\gamma - \rho$ is relatively small; the case (II) is almost close to the case (III) of OL-$\wedge$ attack.

In order to hinder the attack of OL-$\wedge$, for the security parameter λ of [16], the following are the asymptotics conditions given by various literature.
condition 1: $\gamma \geq \Omega(\lambda\eta^2)$ ([16]);
condition 2: $\gamma \geq \Omega\left(\frac{\lambda}{\log \lambda}(\eta - \rho)^2\right)$ ([34]);
condition 3: $\gamma \geq \Omega\left(\frac{\lambda}{\log \lambda}(\eta - \rho)^2\right) + \rho$ ([42]).

Obviously, condition 2 is better than condition 1. Compared to the condition 2 in [34], condition 3 in [42] is better in the case that $(\gamma - \rho)$ is relatively small.

Galbraith et al. [36] showed the success condition of the case (I) of OL-$\wedge$ attack based on the LLL algorithm. In [42], when α is in general case, the two OL attacks based on the BKZ-β algorithm, the expression on ACD parameters γ, η, ρ, the number t of ACD samples and the root-Hermite factor δ_0 were given by (24). This expression can be used to evaluate the specific security of ACD-based schemes.

4.3 Comparision with the MP Algorithm

The common drawback of MP Algorithm is that the dimensions and entries of the involved lattices are quite large, which affects the speed of the algorithm. Galbraith et al.([36]) pointed out that the MP approach is not better than the OL-$\wedge$ attack for practical cryptanalysis. Hence, the OL-$\wedge$ attacks in case (II) and (III) have more advantageous than the MP approach.

4.4 Brief summary

From (22) and (33), the following theorem can be obtained.

Theorem 4. *([42])The time complexities to solve (γ, η, ρ)-ACD instances by running BKZ-β if one SVP oracle costs $2^{O(\beta)}$ is given:*

① The OL-$\wedge$ with $\alpha = 2^{\rho}$: $2^{\Omega\left(\frac{\gamma-\rho}{(\eta-\rho)^2}\log\frac{\gamma-\rho}{(\eta-\rho)^2}\right)}$;

② The OL-$\vee$ with $\alpha = 1$: $2^{\Omega\left(\frac{\gamma}{(\eta-\rho)^2}\log\frac{\gamma}{(\eta-\rho)^2}\right)}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Based on the above analysis and comparison in the first three sections, all the results are presented in Tables1 and 2.

Table 1: Compare the effects of SDA, MP, CN and OL algorithms on ACD problem

OL Attack	**Comparative Results**	**SDA** [2] [37] [36]	**MP** [5] [25] [28] [33]	**Exhaustive search(CN)**

OL-$\wedge$
OL-$\vee$
	case(I): $\alpha = 2^{\rho}$,[OL-$\wedge$];
	case(I): $\alpha = 1$,[OL-$\vee$].
		The performance of SDA is comparable to that of OL.	OL is better than MP;
			OL has many advantages over MP;
				[23] : $O\left(2^{\frac{3\rho}{2}\gamma}\right)$ $(\rho < 40, \gamma < 2^{20})$;

	case(II): α is in general of OL-$\wedge$; case(IV): α is in general of OL-$\vee$.	When $(\gamma - \rho)$ is very small, the case(III) of OL-$\wedge$ is the fast.	The cases with α in general and rounding technique of the two types of OL attacks are more suitable for the situation that ρ is no longer extremely small than η.	[CMNT12]: CN is faster than MP;
	case(III): OL-$\wedge$ with rounding technique.			CN: Not valid for [34].

Table 2: Comparison of Algorithms OL-$\wedge$ and OL-$\vee$

OL Attack	**Comparative Results**	$\gamma > \eta > \rho$	$\gamma - \rho$ **is very small**	$\gamma \gg \rho$	**Bounds on the complexity of [16] with parameter λ**
OL-$\wedge$	case(I): $\alpha = 2^{\rho}$;	Cases (II) and (III) have the same asymptotic time complexity, but case(III) is faster than case(II).	OL-$\wedge$ with $\alpha = 2^{\rho}$ outperforms OL-$\vee$ with $\alpha = 1$. The two performances of OL-$\wedge$ with cases (I) and (II) are equivalent.	The asymptotic time complexity is the same.	Attack conditions against OL-$\wedge$ with case(I) ([33]): $\gamma \geq \Omega(\lambda\eta^2)$;
	case(II): α is in general.				Conditions for improvement([34]): $\gamma \geq \Omega(\frac{\lambda}{\log\lambda}(\eta - \rho)^2)$.
	Case(III): Rounding techniques are used in case(II).				Attack conditions against OL-$\wedge$ with all cases([42]): $\gamma \geq \Omega(\frac{\lambda}{\log\lambda}(\eta - \rho)^2) + \rho$.
OL-$\vee$	case(I): $\alpha = 1$;	The running speed of the improved algorithm in section 3.2.3 is increased compared to that of OL-$\vee$ with $\alpha = 1$.			when $\gamma \gg \rho$, the conditions for resisting OL-$\vee$ attacks are the same as those for resisting OL-$\wedge$.
	case(IV): α is in general;				
	The improved algorithm of OL-$\vee$.				

5. Cryptanalysis of OL attacks in ACD-based FHE Schemes

In this section, by using OL attack, the time complexity of the ACD problem in FHE scheme [DGHV10, KN15, CS15, KT16] is analysed and summarized. In particular, Cheon and Stehle [34] proposed a homomorphic encryption scheme whose parameters are

$$(\rho, \eta, \gamma) = (\lambda, \lambda + log\lambda, \Omega(d^2\lambda log\lambda)),$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Table 3: Comparison of Algorithms OL-$\wedge$ and OL-$\vee$

Comparative Results **OL Attack**		$\gamma > \eta > \rho$	$\gamma - \rho$ **is very small**	$\gamma \gg \rho$	**Bounds on the complexity of [16] with parameter λ**
OL-$\wedge$	case(I): $\alpha = 2^\rho$;	Cases (II) and (III) have the same asymptotic time complexity, but case(III) is faster than case(II).	OL-$\wedge$ with $\alpha = 2^\rho$ outperforms OL-$\vee$ with $\alpha = 1$. The two performances of OL-$\wedge$ with cases (I) and (II) are equivalent.	The asymptotic time complexity is the same.	Attack conditions against OL-$\wedge$ with case(I) ([33]): $\gamma \geq \Omega(\lambda\eta^2)$; Conditions for improvement([34]): $\gamma \geq \Omega(\frac{\lambda}{\log\lambda}(\eta-\rho)^2)$. Attack conditions against OL-$\wedge$ with all cases([42]): $\gamma \geq \Omega(\frac{\lambda}{\log\lambda}(\eta-\rho)^2) + \rho$.
	case(II): α is in general.				
	Case(III): Rounding techniques are used in case(II).				
OL-$\vee$	case(I): $\alpha = 1$;	The running speed of the improved algorithm in section 3.2.3 is increased compared to that of OL-$\vee$ with $\alpha = 1$.			when $\gamma \gg \rho$, the conditions for resisting OL-$\vee$ attacks are the same as those for resisting OL-$\wedge$.
	case(IV): α is in general;				
	The improved algorithm of OL-$\vee$.				

where d is the depth of the circuit for homomorphic calculation, here $\rho \ll \eta$ is no longer satisfied. They also indicated that if the (decision) ACD problem can be solved, then it can be solved LWE. And this set of parameters is obviously difficult.

According to Theorem 2 and Theorem 4, the log of the asymptotic time complexities for solving (γ, η, ρ)-ACD instances are summarized as Table 3.

Table 4: Cryptanalysis of FHE schemes based on ACD

FHE **OL Attack**	**OL-$\wedge$**			**OL-$\vee$**
	$\alpha = 2^\rho$	α in the general case	Rounding technique	$\alpha = 1$
[34] $\gamma = \Omega(L^2\lambda\log\lambda)$ $\eta = \gamma - \lambda$ $\rho = \eta - L\log\lambda$	$\Omega\left(\dfrac{\lambda}{\log\lambda}\right)$			$\Omega(\lambda)$
[16] $\rho = \lambda$ $\eta = \Theta(\lambda^2 - \log^2\lambda)$ $\gamma = \Omega(\lambda^5 - \log^4\lambda)$	$\Omega(\lambda\log\lambda)$			
[35] $\rho = \Theta(\lambda\log\log\log\lambda)$ $\eta = \Theta(\lambda^2\log\log\lambda)$ $\gamma = \Theta(\lambda^4\log^2\lambda)$	$\Omega\left(\dfrac{\log^2\lambda}{\log\log\lambda}\right)$			
[38] $\rho = \Theta(\lambda\log\log\log\lambda)$ $\eta = \Theta(Q^3\lambda^2 - \log\log\lambda)$ $\gamma = \Theta(Q^6\lambda^4\log^2\lambda)$	$\Omega\left(\dfrac{\log^2\lambda}{\log\log\lambda}\right)$			

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Through the analysis of the Table 3, the following conclusions can be drawn:
① For the scheme of [16], those parameters are conservative to get $\Omega(\lambda)$-bit security for OL attacks. Further, according to Theorem 3, using $\gamma = \Omega((\lambda/\log\lambda)\eta^2)$ instead of $\gamma = \Omega(\lambda\eta^2)$ in order to achieve λ-bit security;
② For the scheme of [35], those parameters are optimistic to achieve $\Omega(\lambda)$-bit security for the OL attacks. Furthermore, based on 3, taking $\gamma = \Theta\left(\frac{\lambda^5(\log\log\lambda)^2}{\log\lambda}\right)$ instead of $\gamma = \Theta(\lambda^4\log^2\lambda)$ for obtaining λ-bit security;
③ For the scheme of [38], Q does not effect the asymptotic time complexities of OL attacks compared to the case of the [35] Scheme. The corresponding parameters are also optimistic to achieve $\Omega(\lambda)$-bit security for the OL attacks;
④ For the scheme of [34], the asymptotic time complexities of obtainning the $(\gamma - \rho)$ most significant bits of p for OL attacks is presented.

6. Prospects

Based on the above survey of ACD problem attacks , the ACD problem can be solved under certain conditions using the SDA, OL, MP algorithms. These results show that the applicable range of the three algorithms can be expanded even further. To date, several FHE schemes have been designed based on ACD and variant problems. Existing schemes conservatively set the parameters to be secure against these algorithms. Therefore, it is still worth further exploration to improve the existing algorithms for solving the ACD problem.

Although the present survey offers an initial contribution to the literature concerning the algorithms for ACD problem, the following open problems for further research are left. The application of the improved algorithm in section 3.2.3 needs to be considered for achieving cryptanalysis. Whether the algorithm can be improved to reduce parameter constraints.

For future work, our work points to some directions for addressing the ACD problem, which has great potential in term of further improvement in both theory and practice. This improvement is very much related to the Hermit factor of the reduction algorithm.

7. Conclusions

In this paper, known attacks on the ACD problem are investigated. The performance and application range of each algorithm are analyzed. OL algorithms are divided into two categories (OL-$\wedge$ and OL-$\vee$) for the first time. This work is very helpful for OL attacks to achieve better results. An improved algorithm of OL-$\vee$ is presented to solve the GACD problem. This algorithm works well in polynomial time if the parameter satisfies certain conditions. Compared with [31], the lattice reduction algorithm is used only once, and when the error term $\mathbf{r}$ is recovered in [31], the possible difference between the restored and the true value of p is given. It is helpful to expand the scope of OL attacks.

Author Contributions: Conceptualization, Ran Y. and Wang L.; Methodology, Ran Y. and Wang L.; Validation, Pan Y. and Wang L.; Writing—original draft preparation, Ran Y.; Writing—review and editing, all; 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. A. K. Lenstra; H. W. Lenstra; and L. Lovász. Factoring polynomials with rational coeffcients. *Math. Ann.* **1982**, 261(4): 515–534.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

2. J. C. Lagarias. The computational complexity of simultaneous Diophantine approximation problems. *SIAM J. Comput.* **1985**, 14(1): 196–209.
3. Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.* **1994**, 66: 181–199.
4. D. E. Knuth. The art of computer programming, *seminumerical algorithm*[J]. Software: Practice and Experience, **1982**, 12(9): 883–884
5. N. Howgrave-Graham. Approximate integer common divisors. Cryptography and Lattices. *Springer Berlin Heidelberg*, **2001**: 51–66.
6. 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.
7. Avrim Blum, Adam Kalai and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of ACM*, **2003**, 50(4): 506–519.
8. 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.
9. Phong Q. Nguyen and Damien Stehlé. LLL on the Average. In Florian Hess, Sebastian Pauli and Michael E. Pohst (eds.), ANTS-VII, *Springer LNCS 4076*, **2006**: 238-256.
10. V. Lyubashevsky. The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. APPROX-RANDOM 2005, *Springer LNCS 3624*, **2005**: 378–389.
11. C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions[C], *Proceedings of the fortieth annual ACM symposium on Theory of computing. ACM*, **2008**: 197–206.
12. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology-EUROCRYPT 2008*, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13–17, 2008. Proceedings, **2008**: 31-51.
13. C. Gentry. A Fully Homomorphic Encryption Scheme. PhD thesis, The department of computer science, *Stanford University*, Stanford, CA, USA, **2009**.
14. P. Q. Nguyen, Valle B. The LLL algorithm: survey and applications. *Springer Publishing Company*, Incorporated, **2009**.
15. C. Gentry, Toward basing fully homomorphic encryption on worst-case hardness, in: T. Rabin(ed.), *Advances in Cryptology-CRYPTO 2010*, Lecture Notes in Comput. Sci. Springer, Berlin, Heidelberg, **2010**, 6223: 116–137.
16. M. Van Dijk, C.Gentry, S. Halevi, V. Vaikuntanathan, Fully homomorphic encryption over the integers, in: H. Gilbert (ed.), *Advances in CryptologyEUROCRYPT 2010*, Lecture Notes in Comput. Sci. Springer, Berlin, Heidelberg, **2010**, 6110: 24–43.
17. G. Hanrot, X. Pujol, and D. Stehlé. Terminating bkz. *IACR Cryptology ePrint Archive*, **2011**: 198.
18. H. Cohn, N. Heninger. Approximate common divisors via lattices. *CoRR*, abs/1108. 2714, **2011**.
19. 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.
20. A. Novocin, D. Stehlé, and G. Villard. An LLL-reduction algorithm with quasi-linear time complexity: extended abstract. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, **2011**: 403–412.
21. S. D. Galbraith, Mathematics of Public Key Cryptography. *Cambridge University Press*, **2012**.
22. Y. Ramaiah, G. Kumari, Efficient public key generation for homomorphic encryption over the integers[C]. *Third International conference on advances in communication, network and computing*. **2012**.
23. 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.
24. 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.
25. H. Cohn, N. Heninger. Approximate common divisors via lattices. In *proceedings of ANTS X, vol. 1 of The Open Book Series*, **2013**: 271–293.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

26. J. H. Cheon, J. S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *Proc. of EUROCRYPT*, Springer LNCS, **2013**, 7881: 315-335.
27. Y. Chen. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. *Ph.d theses*, Paris 7, June **2013**.
28. Atsushi Takayasu and Noboru Kunihiro, Better Lattice Constructions for Solving Multivariate Linear Equations Modulo Unknown Divisors, *IEICE Transactions* 97-A, **2014**, 6: 1259-1272.
29. J. Hoffstein, J. Pipher, and J. H. Silverman. An Introduction to Mathematical Cryptography. *Springer Publishing Company*, 2nd edition, **2014**.
30. 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**.
31. J. Ding, C. Tao. A New Algorithm for Solving the Approximate Common Divisor Problem and Cryptanalysis of the FHE based on GACD. *IACR Cryptol. ePrint Arch*, **2014**: 42.
32. J. S. Coron, T. Lepoint, M. Tibouchi, Scale-Invariant Fully Homomorphic Encryption Over the Integers, in: H. Krawczyk (ed.), Public-Key CryptographyPKC 2014, Lecture Notes in *Comput. Sci. Springer*, Berlin, Heidelberg, **2014**, 8383: 311-328.
33. T. Lepoint. Design and Implementation of Lattice-Based Cryptography. *Cryptography and Security* [cs.CR]. Ecole Normale Supérieure de Paris-ENS Paris, **2014**.
34. 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.
35. K. Nuida, K. Kurosawa. (Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces. *Springer*, Berlin, Heidelberg, **2015**.
36. 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**.
37. S. Galbraith, S. Gebregiyorgis, S. Murphy. Algorithms for the approximate common divisor problem. *LMS Journal of Computation and Mathematics*. 19(A), **2016**.: 58-72.
38. Eunkyung Kim and Mehdi Tibouchi. FHE over the integers and modular arithmetic circuits. In *Cryptology and Network Security-15th International Conference*, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings, **2016**: 435–450.
39. D. Benarroch, Z. Brakerski, T. Lepoint. FHE over the Integers: Decomposed and Batched in the Post-Quantum Regime. *Springer*, Berlin, Heidelberg, **2017**.
40. J. Dyer, M. Dyer, J. Xu. Order-preserving encryption using approximate integer common divisors,Data Privacy Management, Cryptocurrencies and Blockchain Technology: *ESORICS 2017 International Workshops*, DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings. Springer International Publishing, **2017**: 257–274.
41. 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.
42. J. Xu, S. Sarkar, L. Hu, Revisiting orthogonal lattice attacks on approximate common divisor problems and their applications. *Cryptology ePrint Archive*, **2018**.
43. 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.
44. 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 June 2023