Approximate Polynomial Common Divisor Problem Relates to Noisy Multipolynomial Reconstruction

Jun Xu[1,2], Santanu Sarkar[3], and Lei Hu[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] Data Assurance and Communications Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
[3] Indian Institute of Technology, Sardar Patel Road, Chennai 600036, India
{xujun,hulei}@iie.ac.cn,sarkar.santanu.bir@gmail.com

Abstract. In this paper, we investigate the hardness of the approximate polynomial common divisor problem, which is regarded as a polynomial analogy of the approximate integer common divisor problem. In order to solve this problem, we present a simple method by using the polynomial lattice reduction algorithm and contain complete theoretical analyses. Further, we propose an improved lattice attack to reduce both space and time costs. Moreover, these two attacking methods can be directly applied to solving the noisy multipolynomial reconstruction problem in the field of error-correcting codes. On the basis of the above situations, our improved lattice attack performs fastest.

Keywords: Approximate polynomial common divisor problem, noisy multipolynomial reconstruction, polynomial lattice

1 Introduction

1.1 Background

It is well known that the common divisor of given integers can be easily solved by using the extended Euclidean algorithm. However, this problem becomes hard when given integers are the sums of some unknown noises and multiples of the desired common divisor. Such a problem firstly introduced by Howgrave-Graham [21] is called the approximate integer common divisor (Integer-ACD) problem, which is the integer version of approximate common divisor (ACD) problem and has been plenty of applications in the fully homomorphic encryption (FHE) schemes [35, 10, 11, 2, 9, 3]. In fact, the strategy that transforming an easy problem into a hard one by adding the noises has been widely used in cryptography, e.g., the celebrated learning with errors (LWE) problem [33].

There is an analogue between the ring of integers and the ring of polynomials over a field. Naturally, the approximate common divisor problem exists a polynomial version, which is called the approximate polynomial common divisor (Polynomial-ACD) problem. It contains the general approximate polynomial common divisor (Polynomial-GACD) problem and the partial approximate polynomial common divisor (Polynomial-PACD) problem. To be specific, for given nonnegative integers γ, η, ρ satisfying $\gamma > \eta > \rho$, a (γ, η, ρ)-Polynomial-GACD problem is stated as follows:

Let $\mathbb{F}[x]$ be the polynomial ring over the field $\mathbb{F}$. For a random η-degree monic polynomial $p(x) \in \mathbb{F}[x]$, given n samples $a_1(x), \cdots, a_n(x)$ that are polynomials in $\mathbb{F}[x]$ with at most γ-degree satisfy

$$a_i(x) = p(x)q_i(x) + r_i(x) \text{ for } 1 \leq i \leq n,$$

where the $q_i(x)$ and $r_i(x)$ are random polynomials and the degree of $r_i(x)$ is no more than ρ. The goal is to output the approximate common divisor $p(x)$.

The definition of a (γ, η, ρ)-Polynomial-PACD problem is the same as that of a (γ, η, ρ)-Polynomial-GACD problem except that an exact multiple (a γ-degree polynomial) of $p(x)$ is given.

There are efficient algorithms for computing a common divisor of given polynomials. However, the presence of noises leads to that given polynomials may be inexact and changes the nature of such a question, which is the so-called Polynomial-ACD problem. Its various variants have been investigated by many researchers such as [20, 34, 30, 22, 7, 14, 31, 25, 8, 24, 13, 19, 36, 15]. The Polynomial-ACD problem is a key research topic in the symbolic-numeric computing area.

In the coding field, codewords are often affected by noises during transmission. Therefore, one needs to design the efficient decoding algorithm in order to recover the corrupted codewords. The Reed-Solomn code is a classical group of error-correcting codes, which is based on univariate polynomials over finite fields and has been many prominent applications. At STOC 1999, Naor and Pinkas [28] first proposed the noisy polynomial reconstruction problem, which is closely connected to the list decoding of Reed-Solomon codes. At EUROCRYPT 2000, Bleichenbacher and Nguyen [1] distinguished the noisy polynomial reconstruction problem from the noisy polynomial interpolation problem. At ANTS 2012, Cohn and Heninger [4] further considered the multivariate version of this problem, which is called noisy multipolynomial reconstruction problem and defined as follows:

Let $r_1(x), \cdots, r_m(x)$ be m univariate polynomials with at most ρ-degree in $\mathbb{F}[x]$. For given γ distinct points $x_1, \cdots, x_\gamma$ in $\mathbb{F}$, there exist the following γ codewords:

$$\left(r_1(x_1), \cdots, r_m(x_1)\right), \cdots, (r_1(x_\gamma), \cdots, r_m(x_\gamma)).$$

Suppose that η codewords are not corrupted and correct in the received γ codewords, the goal is to efficiently reconstruct each polynomial $r_i(x)$.

In fact, this problem for $m = 1$ corresponds to a list decoding algorithm of Reed-Solomon codes. In order to increase the feasible decoding radius of these codes, Guruswami and Sudan [18] gave a list-decoding algorithm that outputs a list of polynomially many solutions. In [5], Cohn and Heninger put forward a faster variant of the Guruswami-Sudan algorithm, which was inspired by Howgrave-Graham's approach [21] for solving the Integer-PACD problem.

Parvaresh-Vardy codes [32] are based on the noisy multipolynomial reconstruction problem. Lately, Guruswami-Rudra codes [17] achieved the improved rates by transmitting less symbols. Recently, Devet, Goldberg and Heninger [12] pointed out that the connections between the noisy multipolynomial reconstruction problem and some kind of private information retrieval (PIR) and further designed an optimally robust PIR based on this problem.

Based on the Lagrange interpolation technique, a polynomial with degree at most ρ can be reconstructed when at least $\rho + 1$ points and the corresponding evaluations are given. It implies that the number η of correct codewords should be greater than or equal to $\rho + 1$ for solving the noisy multipolynomial reconstruction problem in the polynomial time. By utilizing clever polynomial constructions to decode the codewords, Parvaresh and Vardy [32] and Guruswami and Rudra [17] approach such an asymptotic limit of $\eta \geq \rho + 1$. Lately, Cohn and Heninger [4] heuristically analyzed the noisy multipolynomial reconstruction problem based on the algebraic independence hypothesis and obtained the bound $\eta > \rho^{\frac{m}{m+1}}\gamma^{\frac{1}{m+1}}$ by using the idea of Coppersmith's method [6] for finding small solutions of multivariate polynomial equations. However, the drawback of the Cohn-Heninger work is that the dimensions and degrees of the input lattice basis matrices are quite large. Moreover, it is also very time consuming to solve the obtained multivariate polynomial equations by using the Gröbner basis technique or the resultant method. In [12], Devet, Goldberg and Heninger proposed a heuristic lattice method and presented the bound $\eta \geq \gamma - \frac{m}{m+1}(\gamma - \rho - 1)$ to solve the noisy polynomial reconstruction problem. Compared with previous works, this approach is extremely fast in practice. Unfortunately, the corresponding theoretical analysis of this algorithm was not given.

1.2 Our work

In this paper, the reduction from the noisy multipolynomial reconstruction problem to the Polynomial-ACD problem is presented. There are the following three contributions:

- A simple method to solve the Polynomial-ACD problem is proposed, which is based on the polynomial lattice reduction algorithm.
- In order to further reduce both space and time attack complexities, an improved lattice method is put forward.
- The fastest attack on the noisy multipolynomial reconstruction problem is obtained by utilizing our improved lattice algorithm for solving the Polynomial-PACD problem.

1.3 Organization

In Section 2, we present some notations and preliminary knowledge. In Section 3, we give a new attack for solving Polynomial-ACD problems and present the corresponding explanation. We further propose an improved attack in Section 4. We analyze the noisy multipolynomial reconstruction problem in Section 5. We present the experiment result in Section 6. Section 7 concludes the paper.

2 Preliminaries

2.1 Notations

Let $\mathbb{F}[x]$ be the polynomial ring over the field $\mathbb{F}$. The components or entries of the involved row vectors and matrices in this paper are all polynomials in $\mathbb{F}[x]$. Row vectors are denoted by lowercase bold letters and matrices by uppercase bold letters. Let $\mathbf{a}$ be the vector $(a_1(x), \cdots, a_n(x))$ then the i-th component of $\mathbf{a}$ is the polynomial $a_i(x)$. We write $\deg a_i(x)$ for the degree of the polynomial $a_i(x)$ and denote $\deg \mathbf{a} = \max\limits_i \deg a_i(x)$ by the degree of the vector $\mathbf{a}$. Moreover, for polynomials $a(x), b(x) \in \mathbb{F}[x]$, we denote $\lfloor \frac{a(x)}{b(x)} \rfloor \in \mathbb{F}[x]$ by the quotient after $a(x)$ is divided by $b(x)$. Moreover, the transpose of the vector or matrix is denoted by the symbol T.

2.2 Polynomial lattices

Let $\mathbf{b}_1, \cdots, \mathbf{b}_n$ in $\mathbb{F}[x]^n$ be n linearly independent row vectors. A polynomial lattice L is $\mathbb{F}[x]$-spanned by $\mathbf{b}_1, \cdots, \mathbf{b}_n$ as follows,

$$L = \left\{ \sum_{i=1}^{n} k_i(x) \cdot \mathbf{b}_i \mid k_i(x) \in \mathbb{F}[x] \right\},$$

where $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$ is a basis for L and $\mathbf{B} = [\mathbf{b}_1^T, \cdots, \mathbf{b}_n^T]^T$ is the corresponding basis matrix. The rank or dimension of L is denoted as $\dim L = n$. The determinant of L is computed as $\det L = \det \mathbf{B}$, which is a polynomial in $\mathbb{F}[x]$.

Polynomial lattices have been well studied in [23]. There are several polynomial lattice basis reduction algorithms such as [27, 16] in the polynomial time that outputs a reduced basis $\mathbf{v}_1, \cdots, \mathbf{v}_n$ for L satisfying

$$\deg \mathbf{v}_1 + \cdots + \deg \mathbf{v}_n = \deg \det L. \tag{1}$$

If the reduced basis has been ordered such that $\deg \mathbf{v}_1 \leq \cdots \leq \deg \mathbf{v}_n$, then there is

$$\deg \mathbf{v}_1 \leq \frac{\deg \det L}{n},$$

Further, from $\deg \det(L) - (n - (i-1)) \deg \mathbf{v}_i = \sum\limits_{j=1}^{n} \deg \mathbf{v}_j - \sum\limits_{j=i}^{n} \deg \mathbf{v}_i \geq 0$, one can obtain the following properties:

$$\deg \mathbf{v}_i \leq \frac{\deg \det L}{n - (i-1)} \text{ for } 1 \leq i \leq n.$$

Here $\mathbf{v}_1$ is a minimal vector in polynomial lattice L. However, in the case of integer lattices [29], finding a shortest vector is an NP-hard problem and efficient lattice reduction algorithms such as LLL [26] only get an exponential approximation.

3 An Algorithm on Solving Polynomial-ACD Problem

In this section, we present a new method to solve the polynomial-ACD problem. For n samples of a (γ, η, ρ)-polynomial-ACD problem, $a_1(x), \cdots, a_n(x)$, we first define a polynomial lattice $L(\alpha)$ parameterized by polynomial $\alpha(x)$, which is spanned by the row vectors of the following $n \times n$ matrix

$$\mathbf{M}(\alpha) = \begin{pmatrix} \alpha(x) & & & a_1(x) \\ & \ddots & & \vdots \\ & & \alpha(x) & a_{n-1}(x) \\ & & & a_n(x) \end{pmatrix}$$

where $\deg a_n(x) = \gamma$ and $0 \le \deg \alpha(x) < \gamma$.

3.1 Finding polynomial equations on $\mathbf{q}_1(x), \cdots, \mathbf{q}_n(x)$

First, we present the following lemma on the vector in lattice $L(\alpha)$.

Theorem 1. *Given a vector* $\mathbf{v} = \left(\alpha(x)u_1(x), \cdots, \alpha(x)u_{n-1}(x), \sum\limits_{i=1}^{n} u_i(x)a_i(x)\right)$ *in* $L(\alpha)$*, then*

$$\deg \sum_{i=1}^{n} u_i(x)q_i(x) \le \max\{\deg \mathbf{v}, \deg \mathbf{v} - \deg \alpha(x) + \rho\} - \eta.$$

Proof. From $a_i(x) = p(x)q_i(x) + r_i(x)$ for $i = 1, \cdots, n$, we get the following equation

$$p(x)\sum_{i=1}^{n} u_i(x)q_i(x) = \sum_{i=1}^{n} u_i(x)a_i(x) - \sum_{i=1}^{n} u_i(x)r_i(x).$$

According to $\deg p(x) = \eta$, we have

$$\eta + \deg \sum_{i=1}^{n} u_i(x)q_i(x) = \deg\left(\sum_{i=1}^{n} u_i(x)a_i(x) - \sum_{i=1}^{n} u_i(x)r_i(x)\right). \tag{2}$$

Next, we analyze the upper bound of $\deg \sum\limits_{i=1}^{n} u_i(x)q_i(x)$. According to $\mathbf{v} = \left(\alpha(x)u_1(x), \cdots, \alpha(x)u_{n-1}(x), \sum\limits_{i=1}^{n} u_i(x)a_i(x)\right)$ and the definition of the degree of polynomial vector, we obtain

$$\deg \sum_{i=1}^{n} u_i(x)a_i(x) \le \deg \mathbf{v}, \tag{3}$$

Algorithm 1 Solving Polynomial-ACD problem

```
Input: $(\gamma, \eta, \rho)$-Polynomial-ACD samples $a_1(x), \cdots, a_n(x)$ where $\gamma > \eta > \rho + 1$
Output: $p(x)$ or the $(\gamma - \rho)$ most significant coefficients of $p(x)$
Construct the $n \times n$ polynomial matrix
    $$\mathbf{M}(x^\rho) = \begin{pmatrix} x^\rho & & & a_1(x) \\ & \ddots & & \vdots \\ & & x^\rho & a_{n-1}(x) \\ & & & a_n(x) \end{pmatrix},$$
  where $\deg a_n(x) = \gamma$. For the case of the Polynomial-PACD problem, take $a_n(x) = p(x)q_n(x)$, i.e. $r_n(x) = 0$
Run a polynomial lattice basis row reduction algorithm on $\mathbf{M}(x^\rho)$
Rearrange rows of the reduced matrix according to the degrees from small to large and write it as matrix $\mathbf{M}'(x^\rho)$
If the degrees of at least two rows in $\mathbf{M}'(x^\rho)$ are larger than or equal to $\eta$, abort
Write $\mathbf{U}$ such that $\mathbf{U} \cdot \mathbf{M}(x^\rho) = \mathbf{M}'(x^\rho)$, where $\mathbf{U}$ is a unimodular $n \times n$ matrix. Write the last column of matrix $\mathbf{U}^{-1}$ as $(w_{1n}(x), \cdots, w_{nn}(x))^T$
if it is a case of Polynomial-PACD problem then
    Calculate $d^{-1}\frac{a_n(x)}{w_{nn}(x)}$, where $d$ is some constant such that $d^{-1}\frac{a_n(x)}{w_{nn}(x)}$ is monic.
    Set $p(x) = d^{-1}\frac{a_n(x)}{w_{nn}(x)}$
    return $p(x)$
else
    Compute $d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$, where $d$ is some constant satisfying $d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$ is monic.
    if $\gamma > \eta + \rho$ then
        Set $p(x) = d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$
        return $p(x)$
    else
        return the $(\gamma - \rho)$ most significant coefficients of $d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$
    end if
end if
```

and $\deg u_i(x) \leq \deg \mathbf{v} - \deg \alpha(x)$ for $1 \leq i \leq n - 1$. Note that $\deg r_i(x) \leq \rho$, we further have

$$\deg \sum_{i=1}^{n-1} u_i(x) r_i(x) \leq \deg \mathbf{v} - \deg \alpha(x) + \rho. \tag{4}$$

Due to

$$u_n(x)a_n(x) = \left(\sum_{i=1}^{n} u_i(x)a_i(x)\right) - \left(\sum_{i=1}^{n-1} u_i(x)a_i(x)\right),$$

from (3) and $\deg \sum_{i=1}^{n-1} u_i(x)a_i(x) \leq \deg \mathbf{v} - \deg \alpha(x) + \gamma$, we deduce

$$\deg u_n(x) + \deg a_n(x) \leq \max\{\deg \mathbf{v}, \deg \mathbf{v} - \deg \alpha(x) + \gamma\}.$$

According to $0 \leq \deg \alpha(x) \leq \gamma$ and $\deg a_n(x) = \gamma$, we obtain $\deg u_n(x) \leq \deg \mathbf{v} - \deg \alpha(x)$. Further,

$$\deg u_n(x) r_n(x) \leq \deg \mathbf{v} - \deg \alpha(x) + \rho. \tag{5}$$

Based on (3), (4) and (5), there are

$$\deg \left(\sum_{i=1}^{n} u_i(x) a_i(x) - \sum_{i=1}^{n} u_i(x) r_i(x) \right) \leq \max \left\{ \deg \mathbf{v}, \deg \mathbf{v} - \deg \alpha(x) + \rho \right\}.$$

Plugging this relation into (2), we have

$$\deg \sum_{i=1}^{n} u_i(x) q_i(x) \leq \max \left\{ \deg \mathbf{v}, \deg \mathbf{v} - \deg \alpha(x) + \rho \right\} - \eta.$$

□

In order to find polynomial equations on $q_1(x), \cdots, q_n(x)$, we run the lattice basis row reduction algorithm on $\mathbf{M}(\alpha)$. For the sake of discussion, we rearrange the reduced matrix according to the degrees of row vectors from small to large and let the obtained matrix be $\mathbf{M}'(\alpha)$. Then, we can directly get the following corollary based on Lemma 1.

Corollary 1. *Let* $\mathbf{v}_i = \left(\alpha(x) u_{i1}(x), \cdots, \alpha(x) u_{i,n-1}(x), \sum_{j=1}^{n} u_{ij}(x) a_j(x) \right)$ *be the i-th row vector of* $\mathbf{M}'(\alpha)$ *for* $1 \leq i \leq n$. *Then*

$$\deg \sum_{j=1}^{n} u_{ij}(x) q_j(x) \leq \max \left\{ \deg \mathbf{v}_i, \deg \mathbf{v}_i - \deg \alpha(x) + \rho \right\} - \eta.$$

Further, under the condition

$$\max \left\{ \deg \mathbf{v}_i, \deg \mathbf{v}_i - \deg \alpha(x) + \rho \right\} \leq \eta - 1, \tag{6}$$

we have $\sum_{j=1}^{n} u_{ij}(x) q_j(x) = 0$.

3.2 Recovering $q_1(x), \cdots, q_n(x)$

Suppose that the condition (6) holds for the first $n-1$ row vectors of $\mathbf{M}'(\alpha)$, we can obtain $n-1$ linearly independent homogeneous equations

$$\sum_{j=1}^{n} u_{ij}(x) q_j(x) = 0 \text{ for } i = 1, \cdots, n-1.$$

Let $d_n(x) = \sum_{j=1}^{n} u_{nj}(x)q_j(x)$ and denote $\mathbf{U}$ as matrix $(u_{ij}(x))_{n\times n}$. We have $\mathbf{U}\cdot\mathbf{M}(\alpha) = \mathbf{M}'(\alpha)$ and

$$\mathbf{U}\left(q_1(x), \cdots, q_n(x)\right)^T = \left(0, \cdots, 0, d_n(x)\right)^T.$$

Note that $\mathbf{M}(\alpha)$ and $\mathbf{M}'(\alpha)$ are lattice basis matrices of $L(\alpha)$, hence $\mathbf{U}$ and $\mathbf{U}^{-1}$ are both unimodular matrices. Left multiply $\mathbf{U}^{-1}$ by both sides of the above equation and get

$$(q_1(x), \cdots, q_n(x))^T = \mathbf{U}^{-1}(0, \cdots, 0, d_n(x))^T.$$

Let $(w_{1n}(x), \cdots, w_{nn}(x))^T$ be the last column of matrix $\mathbf{U}^{-1}$, which can be publicly computed. According to the above equation, we get

$$(q_1(x), \cdots, q_n(x)) = d_n(x)\cdot(w_{1n}(x), \cdots, w_{nn}(x)). \tag{7}$$

It implies that $d_n(x)$ is a common divisor of $q_1(x), \cdots, q_n(x)$. With an overwhelming probability, polynomials $q_1(x), \cdots, q_n(x)$ are coprime, that is, $d_n(x)$ is a unit in field $\mathbb{F}$. We denote the nonzero constant d as $d_n(x)$ for the sake of discussion.

Next, we obtain such a d. From $a_i(x) = p(x)q_i(x)+r_i(x)$, $\deg r_i(x) < \deg p(x)$ and $p(x)$ is monic, we get that the leading coefficient of $q_i(x)$ is equal to that of the corresponding $a_i(x)$. Therefore, we can determine d by comparing the leading coefficients of both sides in (7). Furthermore, $q_1(x), \cdots, q_n(x)$ are acquired.

3.3 Recovering $p(x)$

The case of Polynomial-PACD. Without loss of generality, let $r_n(x) = 0$, i.e., $a_n(x) = p(x)q_n(x)$. From (7), we deduce $p(x) = d^{-1}\frac{a_n(x)}{w_{nn}(x)}$. Moreover, we recover $r_i(x)$ $(1 \le i \le n)$ according to $r_i(x) = a_i(x) \bmod p(x)$.

The case of Polynomial-GACD. According to $a_n(x) = p(x)q_n(x) + r_n(x)$ and (7), we obtain

$$d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor = p(x) + d^{-1}\lfloor\frac{r_n(x)}{w_{nn}(x)}\rfloor. \tag{8}$$

Note that $\deg a_n(x) = \gamma$ and $\deg p(x) = \eta$, we have $\deg q_n(x) = \deg w_{nn}(x) = \gamma - \eta$. From $\deg r_n(x) \le \rho$, we derive

$$\deg\lfloor\frac{r_n(x)}{w_{nn}(x)}\rfloor \le \rho - (\gamma - \eta) = \eta - (\gamma - \rho).$$

If $\gamma > \eta + \rho$, we have $\deg\lfloor\frac{r_n(x)}{w_{nn}(x)}\rfloor < 0$, i.e., $\lfloor\frac{r_n(x)}{w_{nn}(x)}\rfloor = 0$. Plugging it into (8), we get $p(x) = d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$. Furthermore, the $r_i(x)$ $(1 \le i \le n)$ are obtained due to $r_i(x) = a_i(x) \bmod p(x)$.

If $\gamma \le \eta + \rho$, according to (8), $\deg\lfloor\frac{r_n(x)}{w_{nn}(x)}\rfloor \le \eta - (\gamma - \rho)$ and $\deg p(x) = \eta$, we obtain that the $(\gamma - \rho)$ most significant coefficients of $p(x)$ are respectively equal to those of $d^{-1}\lfloor\frac{a_n(x)}{w_{nn}(x)}\rfloor$.

3.4 Optimizing $\alpha(x)$

Note that the key point that our strategy can work is to get the following polynomial equations

$$\sum_{j=1}^{n} u_{ij}(x)q_j(x) = 0 \text{ for } i = 1, \cdots, n-1 \text{ and } \sum_{j=1}^{n} u_{nj}(x)q_j(x) = d$$

where d is some nonzero constant. According to Corollary 1, we obtain that the above equations hold under the condition

$$\begin{cases} \max\{\deg \mathbf{v}_1, \deg \mathbf{v}_1 - \deg \alpha(x) + \rho\} \leq \eta - 1 \\ \qquad\qquad\qquad \vdots \\ \max\{\deg \mathbf{v}_{n-1}, \deg \mathbf{v}_{n-1} - \deg \alpha(x) + \rho\} \leq \eta - 1 \\ \max\{\deg \mathbf{v}_n, \deg \mathbf{v}_n - \deg \alpha(x) + \rho\} = \eta \end{cases} \tag{9}$$

When $0 \leq \deg \alpha(x) \leq \rho$, the condition (9) becomes

$$\deg \mathbf{v}_i \leq \eta - \rho + \deg \alpha(x) - 1 \text{ for } 1 \leq i \leq n-1 \text{ and } \deg \mathbf{v}_n = \eta - \rho + \deg \alpha(x). \tag{10}$$

When $\deg \alpha(x) \geq \rho$, (9) becomes

$$\deg \mathbf{v}_i \leq \eta - 1 \text{ for } 1 \leq i \leq n-1 \text{ and } \deg \mathbf{v}_n = \eta. \tag{11}$$

Since $\deg \det L(\alpha) = (n-1)\deg \alpha(x) + \gamma$, from (1) we get

$$\deg \mathbf{v}_1 + \cdots + \deg \mathbf{v}_n = (n-1)\deg \alpha(x) + \gamma. \tag{12}$$

Plugging (10) and (11) into (12) respectively, we deduce that

$$n \geq \begin{cases} \frac{\gamma + \rho - \eta - \deg \alpha(x)}{\eta - \rho - 1} + 1, & 0 \leq \deg \alpha(x) \leq \rho, \\ \frac{\gamma - \eta}{\eta - \deg \alpha(x) - 1} + 1, & \deg \alpha(x) \geq \rho. \end{cases}$$

It is easy to see that the above condition is optimal when $\deg \alpha(x) = \rho$. For the sake of simplicity, we take $\alpha(x) = x^{\rho}$. In this situation, the above condition is

$$n \geq \frac{\gamma - \eta}{\eta - \rho - 1} + 1. \tag{13}$$

4 Improved Lattice for Polynomial-ACD Problem

In this section, we propose an improved lattice to obtain more optimal space and time complexities. Let $\hat{L}(\beta)$ be the polynomial lattice spanned by the row vectors of the following $n \times n$ matrix

$$\hat{\mathbf{M}}(\beta) = \begin{pmatrix} 1 & & & \lfloor \frac{a_1(x)}{\beta(x)} \rfloor \\ & \ddots & & \vdots \\ & & 1 & \lfloor \frac{a_{n-1}(x)}{\beta(x)} \rfloor \\ & & & \lfloor \frac{a_n(x)}{\beta(x)} \rfloor \end{pmatrix}$$

Algorithm 2 Further solving Polynomial-ACD problem

```
Input: $(\gamma, \eta, \rho)$-Polynomial-ACD samples $a_1(x), \cdots, a_n(x)$ where $\gamma > \eta > \rho + 1$
Output: $p(x)$ or the $(\gamma - \rho)$ most significant coefficients of $p(x)$
Construct the $n \times n$ polynomial matrix
    $$\hat{\mathbf{M}}(x^\rho) = \begin{pmatrix} 1 & & & \lfloor \frac{a_1(x)}{x^\rho} \rceil \\ & \ddots & & \vdots \\ & & 1 & \lfloor \frac{a_{n-1}(x)}{x^\rho} \rceil \\ & & & \lfloor \frac{a_n(x)}{x^\rho} \rceil \end{pmatrix}$$
    where $\deg a_n(x) = \gamma$. For the case of the Polynomial-PACD problem, take $a_n(x) = p(x)q_n(x)$, i.e. $r_n(x) = 0$
Run a polynomial lattice basis row reduction algorithm on $\hat{\mathbf{M}}(x^\rho)$
Rearrange rows of the reduced matrix according to the degrees from small to large and write it as matrix $\hat{\mathbf{M}}'(x^\rho)$
If the degrees of at least two rows in $\hat{\mathbf{M}}'(x^\rho)$ are larger than or equal to $\eta - \rho$, abort
Write $\mathbf{U}$ such that $\mathbf{U} \cdot \hat{\mathbf{M}}(x^\rho) = \hat{\mathbf{M}}'(x^\rho)$, where $\mathbf{U}$ is a unimodular $n \times n$ matrix. Write the last column of matrix $\mathbf{U}^{-1}$ as $(w_{1n}(x), \cdots, w_{nn}(x))^T$
if it is a case of Polynomial-PACD problem then
    Calculate $d^{-1}\frac{a_n(x)}{w_{nn}(x)}$, where $d$ is some constant such that $d^{-1}\frac{a_n(x)}{w_{nn}(x)}$ is monic.
    Set $p(x) = d^{-1}\frac{a_n(x)}{w_{nn}(x)}$
    return $p(x)$
else
    Compute $d^{-1}\lfloor \frac{a_n(x)}{w_{nn}(x)} \rceil$, where $d$ is some constant satisfying $d^{-1}\lfloor \frac{a_n(x)}{w_{nn}(x)} \rceil$ is monic.
    if $\gamma > \eta + \rho$ then
        Set $p(x) = d^{-1}\lfloor \frac{a_n(x)}{w_{nn}(x)} \rceil$
        return $p(x)$
    else
        return the $(\gamma - \rho)$ most significant coefficients of $d^{-1}\lfloor \frac{a_n(x)}{w_{nn}(x)} \rceil$
    end if
end if
```

where $\deg a_n(x) = \gamma$ and $0 \le \deg \beta(x) < \gamma$. Then, we present the Algorithm 2 for further solving Polynomial-ACD problem.

4.1 Main results

In this subsection, we provide the corresponding explanations on Algorithm 2. First, we give the following theorem, the analysis of which is similar to that in Section 3. The difference is that the rounding operation is involved in $\hat{L}(\beta)$. We give the detailed analysis in Appendix A.

Theorem 2. *Given a vector* $\hat{\mathbf{v}} = \left(u_1(x), \cdots, u_{n-1}(x), \sum_{i=1}^{n} u_i(x) \lfloor \frac{a_i(x)}{\beta(x)} \rceil \right)$ *in* $\hat{L}(\beta)$, *then*

$$\deg \sum_{i=1}^{n} u_i(x)q_i(x) \le \max\{\deg \hat{\mathbf{v}} + \deg \beta(x), \deg \hat{\mathbf{v}} + \rho\} - \eta.$$

For finding polynomial equations on $q_1(x), \cdots, q_n(x)$, we implement the lattice basis row reduction algorithm on $\hat{\mathbf{M}}(\beta)$. Then we rearrange the row vectors of reduced matrix according to the degrees from small to large and let the corresponding matrix be $\hat{\mathbf{M}}'(\beta)$. Based on Theorem 2, we directly have the following result.

Corollary 2. *Let* $\hat{\mathbf{v}}_i = \left(u_{i1}(x), \cdots, u_{i,n-1}(x), \sum_{j=1}^{n} u_{ij}(x) \lfloor \frac{a_j(x)}{\beta(x)} \rfloor \right)$ *be the i-th row vector of* $\hat{\mathbf{M}}'(\beta)$*. Then*

$$\deg \sum_{j=1}^{n} u_{ij}(x) q_j(x) \leq \max \{\deg \hat{\mathbf{v}}_i + \deg \beta(x), \deg \hat{\mathbf{v}}_i + \rho\} - \eta.$$

Furthermore, under the condition

$$\max \{\deg \hat{\mathbf{v}}_i + \deg \beta(x), \deg \hat{\mathbf{v}}_i + \rho\} \leq \eta - 1, \tag{14}$$

we get $\sum_{j=1}^{n} u_{ij}(x) q_j(x) = 0$.

Note that $\dim \hat{L}(\beta) = n$ and $\deg \det \hat{L}(\beta) = \gamma - \deg \beta(x)$. Similar to the analysis in Section 3.4, we can get the optimal $\deg \beta(x) = \rho$. For the sake of simplicity, we choose $\beta(x) = x^\rho$. Correspondingly, the condition (14) becomes

$$\deg \hat{\mathbf{v}}_i \leq \eta - \rho - 1. \tag{15}$$

Therefore, we can deduce that the optimal n satisfies (13), that is, $n \geq \frac{\gamma-\eta}{\eta-\rho-1} + 1$. After the desired $n-1$ polynomial equations $\sum_{j=1}^{n} u_{ij}(x) q_j(x) = 0$ are acquired, we can recover $p(x)$ and $r_1(x), \cdots, r_n(x)$ by using the similar methods in Sections 3.2 and 3.3.

4.2 Analysis of the attack complexity

The dominant calculation of our algorithms is the polynomial lattice reduction for finding equations on $q_1(x), \cdots, q_n(x)$. Mulders and Storjohann [27] presented a simple algorithm in time $O(n^3\delta^2)$. Lately, Giorgi et al. [16] proposed another algorithm which runs in time $O(n^{\omega+o(1)}\delta)$, where δ is the maximum degree of the input basis matrix, n is the dimension, and ω is a valid exponent for matrix multiplication.

Corresponding to lattice $\hat{L}(x^\rho)$ in Algorithm 2, the smallest number of samples $\lceil \frac{\gamma-\eta}{\eta-\rho-1} \rceil + 1$ satisfying (13) is taken as the dimension n, the maximum degree of the input basis matrix $\hat{\mathbf{M}}(\beta)$ is $\gamma - \rho$. Therefore, the involved running time of the lattice reduction algorithm is $O\left((\lceil \frac{\gamma-\eta}{\eta-\rho-1} \rceil + 1)^{\omega+o(1)} (\gamma - \rho) \right)$ for Giorgi et al.'s algorithm.

Remark 1. Compared to lattice $L(x^\rho)$ in Algorithm 1, lattice $\hat{L}(x^\rho)$ adopts the smaller degree, especially when $(\gamma-\rho)$ is small. Note that the involved dimensions are same in these two lattices, hence Algorithm 2 only amounts to a constant improvement of the overall attack complexity. However, such an improvement can be quite significant in practice.

5 Application to Noisy Multipolynomial Reconstruction

In this section, we reduce the noisy multipolynomial reconstruction to Polynomial-PACD problem and then solve the noisy polynomial reconstruction by using our methods.

In the noisy multipolynomial reconstruction problem, there exist η codewords incorrupted and correct in the received γ codewords. First, we let the received γ codewords be

$$(y_{11}, \cdots, y_{m1}), \cdots, (y_{1\gamma}, \cdots, y_{m\gamma})$$

which are respectively corresponding to γ points $x_1, \cdots, x_\gamma$. Without loss of generality, suppose that the first η codewords $(y_{11}, \cdots, y_{m1}), \cdots, (y_{1\eta}, \cdots, y_{m\eta})$ are correct, i.e., the function values $r_i(x_j)$ of η points $x_1, \cdots, x_\eta$ are not corrupted where $i = 1, \cdots, m$. Hence, there are the following relations

$$y_{ij} = r_i(x_j) \text{ for } i = 1, \cdots, m \text{ and } j = 1, \cdots, \eta.$$

Then, we use Lagrange interpolation to construct m polynomial $a_i(x)$ with degree $\gamma - 1$ such that

$$a_i(x_k) = y_{ik} \text{ where } i = 1, \cdots, m \text{ and } j = 1, \cdots, \gamma.$$

Moreover, let $a_{m+1}(x) = (x - x_1) \cdots (x - x_\gamma)$ and $p(x) = (x - x_1) \cdots (x - x_\eta)$. Note that

$$a_i(x_j) = y_{ij} = r_i(x_j) \text{ with } i = 1, \cdots, m \text{ and } j = 1, \cdots, \eta,$$

we have

$$\begin{cases} a_1(x) \equiv r_1(x) \bmod p(x), \\ \qquad \cdots \\ a_m(x) \equiv r_m(x) \bmod p(x), \\ a_{m+1}(x) \equiv 0 \bmod p(x). \end{cases}$$

Then, we let $q_1(x), \cdots, q_m(x), q_{m+1}(x)$ be polynomials in $\mathbb{F}[x]$ such that

$$a_i(x) = p(x)q_i(x) + r_i(x) \text{ for } i = 1, \cdots, m \text{ and } a_{m+1}(x) = p(x)q_{m+1}(x). \quad (16)$$

Obviously, finding $p(x)$ from (16) can be regarded as solving a (γ, η, ρ)-Polynomial-PACD problem. Once the approximate common divisor $p(x)$ is found out, the desired $r_1(x), \cdots, r_m(x)$ are easily obtained. In other words, the noisy multipolynomial reconstruction problem is settled.

Finally, we utilize Algorithms 1 and 2 to solve (16). According to the required condition (13), we get that the noisy multipolynomial reconstruction problem can be solved under the condition $m+1 \geq \frac{\gamma-\eta}{\eta-\rho-1}+1$, i.e.,

$$\eta \geq \frac{\gamma+m(\rho+1)}{m+1}. \tag{17}$$

Remark 2. The condition (17) is close to the theoretical limit $\eta \geq \rho+1$ when m is sufficiently large.

Remark 3. In [12], the authors proposed an algorithm for solving the noisy multipolynomial reconstruction and presented the experimental results. Compared with previous attack methods, this algorithm is the fastest in practice. Meanwhile, the authors conjectured the fail probability is $\left(\frac{1}{|\mathbb{F}|}\right)^{m(\eta-\rho-1)-(\gamma-\eta-1)}$ but did not give a proof, where $|\mathbb{F}|$ is the size of the underlying field $\mathbb{F}$. Hence, in order to make this algorithm work with a high possibility, the authors need the condition $\eta \geq \gamma - \frac{m}{m+1}(\gamma-\rho-1)$, which is the same as (17) in fact. However, we firstly find out that this algorithm exists a slight error (please see [12, Algorithm 1]). Concretely speaking, one should replace "larger than" in Step 5 with "larger than or equal to" and "**b**" in Step 7 with "$-$**b**". Furthermore, we can give the condition (17) theoretically rather than heuristically. Moreover, Algorithm 2 in this paper is faster than Algorithm 1 in [12], which is because that the degrees of polynomials of the involved input matrix are reduced.

6 Experimental Verification

In this section, we respectively utilize Algorithms 1 and 2 to analyze concrete securities of the Polynomial-ACD problem and present the corresponding experimental results in Tables 1 and 2. The experiments are done in Sage 7.4 on Linux Ubuntu 16.04 on a laptop with Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz, 3 GB RAM and 3 MB Cache. We respectively take GF(2), GF(p) and the rational field $\mathbb{Q}$ as the underlying field $\mathbb{F}$ respectively, where p is a random 128-bit prime. The runtime in Tables 1 and 2 refers to the consuming time in second on the polynomial lattice reduction algorithm. The obtained reduced basis matrix by using the polynomial lattice reduction algorithm is the weak Popov form [27]. Moreover, the degrees of the reduced row vectors are almost average for the lattices $L(x^\rho)$ and $\hat{L}(x^\rho)$. In the experiments, random instances of Polynomial-ACD problem are always solved when n is equal to $\lceil \frac{\gamma-\eta}{\eta-\rho-1} \rceil + 2$, which is slightly larger than the theoretical lower bound $\lceil \frac{\gamma-\eta}{\eta-\rho-1} \rceil + 1$. From the experiment results, we can see that Algorithm 2 is significantly faster than Algorithm 1 for larger parameters γ, η, ρ over the 128-bit prime field GF(p) and the rational field $\mathbb{Q}$.

Table 1. Analysis of the Polynomial-ACD problem instances over finite fields GF(2) and GF(p) by utilizing Algorithms 1 and 2.

n (exp.)	η	γ	ρ	Algorithm 1 GF(2) time	Algorithm 2 GF(2) time	Algorithm 1 GF(p) time	Algorithm 2 GF(p) time
4	12	20	7	0.004	0.001	0.016	0.016
5	11	20	7	0.004	0.001	0.028	0.028
7	10	20	7	0.004	0.004	0.080	0.068
13	9	20	7	0.016	0.016	0.336	0.328
25	86	200	80	0.544	0.532	63.752	23.400
31	85	200	80	0.998	1.040	117.360	40.436
41	84	200	80	2.104	2.092	264.072	82.580
61	83	200	80	5.584	5.576	813.452	239.360
120	82	200	80	32.488	32.472	5630.788	1504.964

Table 2. Analysis of the Polynomial-ACD problem instances over the rational field $\mathbb{Q}$ by utilizing Algorithms 1 and 2.

n (exp.)	η	γ	ρ	Algorithm 1 $\mathbb{Q}$ time	Algorithm 2 $\mathbb{Q}$ time
4	12	20	7	0.012	0.004
5	11	20	7	0.016	0.012
7	10	20	7	0.052	0.040
13	9	20	7	0.236	0.168
25	86	200	80	63498.372	23393.656
31	85	200	80	110532.316	30974.632

7 Conclusion

In this paper, the Polynomial-ACD problem was analyzed and two novel lattice attacks were proposed. Further, these two attacks were used for solving the noisy multipolynomial reconstruction problem. In our improved attack, the polynomials for the involved lattice are reduced so that it is the fastest till now.

References

1. Bleichenbacher, D., Nguyen, P.Q.: Noisy polynomial interpolation and noisy chinese remaindering. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. (2000) 53–69
2. Cheon, J., Coron, J.S., Kim, J., Lee, M., Lepoint, T., Tibouchi, M., Yun, A.: Batch fully homomorphic encryption over the integers. In Johansson, T., Nguyen, P., eds.: Advances in Cryptology – EUROCRYPT 2013. Volume 7881 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 315–335
3. Cheon, J., Stehlé, D.: Fully homomophic encryption over the integers revisited. In Oswald, E., Fischlin, M., eds.: Advances in Cryptology – EUROCRYPT 2015. Volume 9056 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2015) 513–536
4. Cohn, H., Heninger, N.: Approximate common divisors via lattices. The Open Book Series **1**(1) (2013) 271–293
5. Cohn, H., Heninger, N.: Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. Adv. in Math. of Comm. **9**(3) (2015) 311–339
6. Coppersmith, D.: Small solutions to polynomial equations, and low exponent rsa vulnerabilities. Journal of Cryptology **10**(4) (1997) 233–260

7. Corless, R.M., Gianni, P.M., Trager, B.M., Watt, S.M.: The singular value decomposition for polynomial systems. In: Proceedings of the 1995 international symposium on Symbolic and algebraic computation, ACM (1995) 195–207
8. Corless, R.M., Watt, S.M., Zhi, L.: Qr factoring to compute the gcd of univariate approximate polynomials. IEEE Transactions on Signal Processing **52**(12) (2004) 3394–3402
9. Coron, J.S., Lepoint, T., Tibouchi, M.: Scale-invariant fully homomorphic encryption over the integers. In Krawczyk, H., ed.: Public-Key Cryptography – PKC 2014. Volume 8383 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 311–328
10. Coron, J.S., Mandal, A., Naccache, D., Tibouchi, M.: Fully homomorphic encryption over the integers with shorter public keys. In Rogaway, P., ed.: Advances in Cryptology – CRYPTO 2011. Volume 6841 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 487–504
11. Coron, J.S., Naccache, D., Tibouchi, M.: Public key compression and modulus switching for fully homomorphic encryption over the integers. In Pointcheval, D., Johansson, T., eds.: Advances in Cryptology – EUROCRYPT 2012. Volume 7237 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 446–464
12. Devet, C., Goldberg, I., Heninger, N.: Optimally robust private information retrieval. In: Proceedings of the 21st USENIX Conference on Security Symposium. Security'12, Berkeley, CA, USA, USENIX Association (2012) 13–13
13. Eliaš, J.: Approximate Polynomial Greatest Common Divisor. PhD thesis, Master Thesis, Charles University in Prague (2012)
14. Emiris, I.Z., Galligo, A., Lombardi, H.: Numerical univariate polynomial gcd. LECTURES IN APPLIED MATHEMATICS-AMERICAN MATHEMATICAL SOCIETY **32** (1996) 323–344
15. Giesbrecht, M., Haraldson, J., Kaltofen, E.: Computing approximate greatest common right divisors of differential polynomials. CoRR **abs/1701.01994** (2017)
16. Giorgi, P., Jeannerod, C., Villard, G.: On the complexity of polynomial matrix computations. In: Symbolic and Algebraic Computation, International Symposium ISSAC 2003, Drexel University, Philadelphia, Pennsylvania, USA, August 3-6, 2003, Proceedings. (2003) 135–142
17. Guruswami, V., Rudra, A.: Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. IEEE Trans. Information Theory **54**(1) (2008) 135–150
18. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometry codes. IEEE Trans. Information Theory **45**(6) (1999) 1757–1767
19. Halikias, G., Galanis, G., Karcanias, N., Milonidis, E.: Nearest common root of polynomials, approximate greatest common divisor and the structured singular value. IMA J. Math. Control & Information **30**(4) (2013) 423–442
20. Hough, D.G.: Explaining and Ameliorating the Ill Condition of Zeros of Polynomials. PhD thesis (1977) AAI7731401.
21. Howgrave-Graham, N.: Approximate integer common divisors. In Silverman, J., ed.: Cryptography and Lattices. Volume 2146 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 51–66
22. Hribernig, V., Stetter, H.J.: Detection and validation of clusters of polynomial zeros. Journal of Symbolic Computation **24**(6) (1997) 667–681
23. Kailath, T.: Linear systems. Volume 156. Prentice-Hall Englewood Cliffs, NJ (1980)

24. Kaltofen, E., Yang, Z., Zhi, L.: Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In: Proceedings of the 2006 international symposium on Symbolic and algebraic computation, ACM (2006) 169–176
25. Karmarkar, N., Lakshman, Y.N.: On approximate gcds of univariate polynomials. Journal of Symbolic Computation **26**(6) (1998) 653–666
26. Lenstra, A.K., Lenstra, H.W., Lovasz, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4) (1982) 515–534
27. Mulders, T., Storjohann, A.: On lattice reduction for polynomial matrices. J. Symb. Comput. **35**(4) (2003) 377–401
28. Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing. STOC '99, New York, NY, USA, ACM (1999) 245–254
29. Nguyen, P.Q., Stern, J.: The two faces of lattices in cryptology. In: Cryptography and Lattices, International Conference, CaLC 2001, Providence, RI, USA, March 29-30, 2001, Revised Papers. (2001) 146–180
30. Noda, M.T., Sasaki, T.: Approximate gcd and its application to ill-conditioned equations. Journal of Computational and Applied Mathematics **38**(1-3) (1991) 335–351
31. Pan, V.Y.: Numerical computation of a polynomial GCD and extensions. PhD thesis, INRIA (1996)
32. Parvaresh, F., Vardy, A.: Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings. (2005) 285–294
33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing. STOC '05, New York, NY, USA, ACM (2005) 84–93
34. Schönhage, A.: Quasi-gcd computations. Journal of Complexity **1**(1) (1985) 118–137
35. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In Gilbert, H., ed.: Advances in Cryptology – EUROCRYPT 2010. Volume 6110 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 24–43
36. Winkler, J.R., Yang, N.: Resultant matrices and the computation of the degree of an approximate greatest common divisor of two inexact bernstein basis polynomials. Computer Aided Geometric Design **30**(4) (2013) 410–429

A Proof on Theorem 2

Proof. According to $a_i(x) = p(x)q_i(x) + r_i(x)$ for $i = 1, \cdots, n$, we get the following equation

$$p(x)\sum_{i=1}^{n} u_i(x)q_i(x) = \sum_{i=1}^{n} u_i(x)a_i(x) - \sum_{i=1}^{n} u_i(x)r_i(x).$$

Note that $\deg p(x) = \eta$, we have

$$\eta + \deg \sum_{i=1}^{n} u_i(x)q_i(x) = \deg \left(\sum_{i=1}^{n} u_i(x)a_i(x) - \sum_{i=1}^{n} u_i(x)r_i(x) \right). \tag{18}$$

Let us analyze the upper bound of $\deg \sum_{i=1}^{n} u_i(x)q_i(x)$ as follows. First, since $\hat{\mathbf{v}} = \left(u_1(x), \cdots, u_{n-1}(x), \sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor\right)$, we easily acquire

$$\deg \sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor \leq \deg \hat{\mathbf{v}}. \tag{19}$$

Note that $\deg a_i(x) = \deg \beta(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor$, we have

$$\deg \sum_{i=1}^{n} u_i(x)a_i(x) = \deg \left(\beta(x) \sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor\right).$$

Furthermore,

$$\deg \sum_{i=1}^{n} u_i(x)a_i(x) \leq \hat{\mathbf{v}} + \deg \beta(x). \tag{20}$$

Second, due to that $\deg u_i(x) \leq \deg \hat{\mathbf{v}}$ and $\deg r_i(x) \leq \rho$ for $1 \leq i \leq n-1$, we easily get

$$\deg \sum_{i=1}^{n-1} u_i(x)r_i(x) \leq \deg \hat{\mathbf{v}} + \rho. \tag{21}$$

Third, according to

$$u_n(x)\lfloor \frac{a_n(x)}{\beta(x)} \rfloor = \left(\sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor\right) - \left(\sum_{i=1}^{n-1} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor\right),$$

we deduce

$$\deg u_n(x)\lfloor \frac{a_n(x)}{\beta(x)} \rfloor \leq \max \left\{\deg \sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor, \deg \sum_{i=1}^{n-1} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor\right\}.$$

Since $\deg\lfloor \frac{a_i(x)}{\beta(x)} \rfloor \leq \gamma - \deg \beta(x)$ and $\deg u_i(x) \leq \deg \hat{\mathbf{v}}$ for $1 \leq i \leq n-1$, we obtain

$$\deg \sum_{i=1}^{n-1} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor < \deg \hat{\mathbf{v}} - \deg \beta(x) + \gamma.$$

We have obtained (19), i.e., $\deg \sum_{i=1}^{n} u_i(x)\lfloor \frac{a_i(x)}{\beta(x)} \rfloor \leq \deg \hat{\mathbf{v}}$, therefore we have

$$\deg u_n(x)\lfloor \frac{a_n(x)}{\beta(x)} \rfloor \leq \max \left\{\deg \hat{\mathbf{v}}, \deg \hat{\mathbf{v}} - \deg \beta(x) + \gamma\right\}.$$

According to $0 \leq \deg \beta(x) \leq \gamma$, we get $\deg u_n(x)\lfloor \frac{a_n(x)}{\beta(x)} \rfloor \leq \deg \hat{\mathbf{v}} - \deg \beta(x) + \gamma$. Note that $\deg\lfloor \frac{a_n(x)}{\beta(x)} \rfloor = \gamma - \deg \beta(x)$, we get $\deg u_n(x) \leq \deg \hat{\mathbf{v}}$. From $\deg r_n(x) \leq \rho$, we further have

$$\deg u_n(x)r_n(x) \leq \deg \hat{\mathbf{v}} + \rho. \tag{22}$$

According to (20), (21) and (22), we get

$$\deg\left(\sum_{i=1}^{n} u_i(x)a_i(x) - \sum_{i=1}^{n} u_i(x)r_i(x)\right) \le \max\left\{\deg \hat{\mathbf{v}} + \deg \beta(x), \deg \hat{\mathbf{v}} + \rho\right\}.$$

Plugging this inequality into (18), we obtain

$$\deg \sum_{i=1}^{n} u_i(x)q_i(x) \le \max\left\{\deg \hat{\mathbf{v}} + \deg \beta(x), \deg \hat{\mathbf{v}} + \rho\right\} - \eta.$$

□