Research Article

Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, and Changmin Lee*

Algorithms for CRT-variant of Approximate Greatest Common Divisor Problem

https://doi.org/10.1515/jmc-2019-0031
Received Jul 15, 2019; accepted Aug 25, 2020

Abstract: The approximate greatest common divisor problem (ACD) and its variants have been used to construct many cryptographic primitives. In particular, the variants of the ACD problem based on Chinese remainder theorem (CRT) are being used in the constructions of a batch fully homomorphic encryption to encrypt multiple messages in one ciphertext. Despite the utility of the CRT-variant scheme, the algorithms that secures its security foundation have not been probed well enough.
In this paper, we propose two algorithms and the results of experiments in which the proposed algorithms were used to solve the variant problem. Both algorithms take the same time complexity $2^{\tilde{O}(\frac{\gamma}{(\eta-\rho)^2})}$ up to a polynomial factor to solve the variant problem for the bit size of samples γ, secret primes η, and error bound ρ. Our algorithm gives the first parameter condition related to η and γ size. From the results of the experiments, it has been proved that the proposed algorithms work well both in theoretical and experimental terms.

Keywords: CCK-ACD; Lattice; orthogonal lattice attack; SDA

2020 Mathematics Subject Classification: 11Y16

1 Introduction

Howgrave-Graham had defined and studied the approximate greatest common divisor (ACD) problem in [16]. The ACD problem and its variant problems have been used to construct cryptographic schemes such as fully homomorphic encryption (FHE) and cryptographic multilinear map [4, 6, 9, 19].

As the first variant problem, the partial approximate common divisor (PACD) problem was suggested. This variant problem has allowed increasing efficiency of ACD-based homomorphic encryption scheme [7]. As the series of work, in the paper [4], another variant of the ACD problem was introduced to suggest a new FHE scheme, which is called CCK-FHE scheme, over the integers. This scheme utilizes Chinese remainder theorem to encrypt multiple messages in one ciphertext. Informally, for integers γ, n, η, and ρ such that $\gamma \gg n \cdot \eta$ and $\eta \gg \rho$, the γ-bit ciphertext integer b of this scheme is characterized by satisfying modulo equations $b \equiv r_i \bmod p_i$ for $1 \le i \le n$, where r_i's are ρ-bit integers and p_i's are η-bit fixed secret primes. The problem that distinguishes between ciphertexts of CCK-FHE scheme and uniform samples of γ-bit integer, in which the γ-bit integer $N = \prod_{i=0}^{n} p_i$ is given as the product of secret primes, is called the CCK-ACD. [1] In case $n = 1$, the problem is called PACD problem.

On the other hand, algorithms to directly solve the CCK-ACD problem have garnered less attention. Galbraith, Gebregiyorgis and Murphy said that an algorithm to solve the CCK-ACD problem exploiting CRT struc-

***Corresponding Author: Changmin Lee:** ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), 46 Allée d'Italie, 69007 Lyon, France; Email: changmin.lee@ens-lyon.fr
Jung Hee Cheon: Seoul National University, 1 Gwanak-ro, 08826 Seoul, South Korea; Email: jhcheon@snu.ac.kr
Wonhee Cho: Seoul National University, 1 Gwanak-ro, 08826 Seoul, South Korea; Email: wony0404@snu.ac.kr
Minki Hhan: Seoul National University, 1 Gwanak-ro, 08826 Seoul, South Korea; Email: hhan_@snu.ac.kr
Jiseung Kim: Seoul National University, 1 Gwanak-ro, 08826 Seoul, South Korea; Email: kaiser351@snu.ac.kr

1 We give a formal definition of the CCK-ACD problem in Section 2.

Open Access. © 2020 J. H. Cheon *et al.*, published by De Gruyter. CC BY This work is licensed under the Creative Commons Attribution 4.0 License

ture is an open problem [13]. In fact, there has been no algorithms for solving the CCK-ACD problem so far except for the method of Chen and Nguyen [3], which depends only on ρ. Instead, in order to provide the evidence of CCK-FHE's security, authors in [4] suggested a reduction from PACD to CCK-ACD.

However, while the current CCK-FHE parameters are set to be secure for the Chen and Nguyen's attack, the authors in [4] did not use the parameter settings obtained from the reduction for known PACD parameters. Therefore, it is necessary to determine whether the CCK-FHE parameters satisfies the desired security even under the current conditions of η and γ. In sum, one can naturally pose the following question:

Is it possible to present the time complexity for solving CCK-ACD
by using a mathematical algorithm that depends on η *and* γ*?*

Previous works

In order to solve the CCK-ACD problem, several naive methods are suggested. Their main idea was to exploit the feature of the problem that the error terms are relatively small and the product of the secret primes is given. In other words, one can try a brute-force attack to recover a secret prime p_i from a multiple $N = \prod_{i=0}^{n} p_i$ and an sample of CCK-ACD represented by $b = p_i \cdot q_i + r_i$ for some fixed i, where an integer $r_i \in (-2^\rho, 2^\rho)$ except $i = 0$. The method is to compute the greatest common divisor between (GCD) $b-a$ and N for all integers $a \in (-2^\rho, 2^\rho)$. It would have time complexity $\tilde{O}(2^\rho)$, so ρ should be set to $\Omega(\lambda)$ for the security parameter λ. Furthermore, [3] and [7] that were proposed as the variants of exhaustive search to solve (P)ACD in $\tilde{O}(2^{\rho/2})$ time complexity, can be applied to solve the CCK-ACD problem for the feature mentioned previously. In addition, one can also use the factorization with the elliptic curve method to find a factor of N in $2^{\tilde{O}(\sqrt{\eta})}$ time complexity, where η is the log-size of p_i. Thus, η should be set to $\Omega(\lambda^2)$ for the security parameter λ.

As another trial to solve CCK-ACD, authors in [14] considered well-known algorithms for solving PACD such as orthogonal lattice attack method (OLA) and simultaneous Diophantine approximation (SDA) [6, 12, 16, 19] in the context of CCK-ACD. The SDA and OLA make use of a lattice reduction algorithm for a specific lattice whose entries consist of the given PACD samples and a multiple $N = \prod_{i=0}^{n} p_i$. If one can obtain a short vector from the lattice by the lattice reduction algorithm, it leads to a solution of the PACD problem which utilizes the coordinates of the vector. Since these algorithms for (P)ACD have time complexity depending on η and γ, one can expect that the expansion of the algorithms to the CCK-ACD problem will provide answers to the main question.

However, if a lattice as similar to SDA and OLA is being constructed to solve CCK-ACD, there exist several short vectors of similar length in the lattice due to the symmetry of p_i. Thus if short vector from the lattice by a lattice reduction algorithm is a short linear combination of some of these vectors, one cannot extract information on a certain prime p_i from the vector.

Independent work

Recently, Coron and Pereira [10] proposed an algorithm to solve the multi-prime ACD problem, which is the same as the 'search' CCK-ACD problem in this paper. The main idea of the attack is also the same as our SDA-style algorithm that combines the SDA with algebraic steps from the Cheon *et al.* [5]. In this paper, we also propose another OLA-style algorithm to solve 'decisional' CCK-ACD problem using OLA with a new distinguisher determinant.

1.1 Our Work

In this paper, we propose two mathematical algorithms to solve the CCK-ACD problem by extending the OLA and SDA methods that are well-known for solving the ACD problem using lattice technique. Both algorithms take the same time complexity $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ up to polynomial factors for the bit-size of samples γ, secret primes η and error ρ. Our algorithms are the first algorithms related to η and γ for solving the CCK-ACD problem.

Let b_j be a CCK-ACD sample of $b_j \equiv r_{ij} \bmod p_i$ for $1 \le j \le k$ and $0 \le i \le n$. Let $\mathbf{b}$ and $\mathbf{r}_i$ be a vector (b_j) and (r_{ij}), respectively. Technically, the first step of the classical OLA algorithm on input b_j is to compute a lattice $\Lambda_N^{\perp}(\mathbf{b})$, which is a set of orthogonal vectors to $\mathbf{b}$ over $\mathbb{Z}_N$. Similarly, one can define a lattice $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$, which is a set of orthogonal vectors to $\mathbf{r}_i$ for all i over the integers. Then we have

$$\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\}) \subset \Lambda_N^{\perp}(\mathbf{b}).$$

It implies that the size of $k-n-1$ shortest vectors of $\Lambda_N^{\perp}(\mathbf{b})$ is less than that of $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$. The classical OLA algorithm assumes that the $k-n-1$ shortest vectors is a generator of $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$. Even more, the algorithm expects that $k-n-1$ short vectors become a generator. So finding $k-n-1$ short vectors is likely to lead us to recover the lattice $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$.

However, one problem might arise after finding those short vectors. In the case of PACD, (i.e $n = 1$), the recovered lattice has rank two and $\|\mathbf{r_1}\| \ll \|\mathbf{r_0}\|$. So we can obtain the vector $\mathbf{r}_1$ easily. Then, the next step is to recover the secret integer p_1 by computing the GCD between $b_j - r_{j1}$ and $N = p_0 \cdot p_1$. If the last step reveals a non-trivial factor of N, we can conclude that the b_j's are PACD samples. Unfortunately, in the case of CCK-ACD, the classical OLA algorithm faces a hard task to recover the exact vector $\mathbf{r}_i$ except for small n since a short vector from the lattice would be a short linear combination of several $\mathbf{r}_i$'s. Instead, we employ a determinant of the lattice as a new distinguisher to solve the decision CCK-ACD problem. We show that a sub-lattice of the output lattice of the classic OLA has determinant of a different-sized depending on the type of inputs. Then, computation of a determinant enables us to avoid the obstacle to find the exact vector $\mathbf{r}_i$. The overall time complexity heavily depends on the cost of a lattice reduction to find a short vector. Therefore, the time complexity shall be asymptotically same to the classical one. For more details, please refer to Section 3.

We also propose a SDA-style algorithm to find all secret parameters in the CCK-ACD problem beyond the decision problem. The algorithm consists of two steps; find a short vector of certain lattice using a lattice reduction algorithm and then recover the factors $p_1, \cdots, p_n$ by employing the Cheon *et al.*'s technique [5]. More precisely, we consider a column lattice generated by the following matrix:

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ \mathbf{b}^T & N \cdot \mathbf{I}_k \end{pmatrix}.$$

According to the original SDA approach, this lattice includes a short vector of the form $(N/p_i, r_{1i} \cdot N/p_i, \ldots, r_{ki} \cdot N/p_i)$ for all i. In the case of $n = 1$, (i.e. PACD problem), the lattice has only one short vector and the first entry is a multiple of N/p_1. So it allows us to factorize N. When it comes to the CCK-ACD problem, any short vector is a linear combination of the vectors and it would not be a multiple of nontrivial factor of N. It means that the first entry of a short vector that we obtain is an integer of the form $\sum_{i=1}^{n} c_i \cdot N/p_i$ for some small integers c_i. In order to use the integer, we should factor in another well-known algorithm. Namely, we would like to cite a technique introduced in [5]. The reference we cite from [5] allows the linear summation of N/p_i to be called a *dual instance*. This instance allows to convert modulo equations into integer equations by exploiting the CRT properties of the CCK-ACD samples and its relation to dual instance. Therefore, it leads to recover N/p_i for all i. The complexity of the new algorithm primarily depends on the first step, so it takes time complexity as stated above. For more details, please refer to Section 4.

We provide experimental results to guarantee that our algorithms work well both in theoretical and experimental terms under the various parameters of CCK-ACD. We observe the OLA is more practical than SDA while the asymptotic complexities are the same.

Organization

In Section 2, we introduce preliminary information related to the lattice. Next, we revisit the OLA to solve the CCK-ACD problem in Section 3. Also, we extend the SDA algorithm in the context of CCK-ACD and propose the first algorithm which recovers all secret primes p_i's of the CCK-ACD problem in Section 4. In addition, we present some experimental results for our algorithms in Section 5.

2 Preliminaries

Notation

Throughout this paper, we use $a \leftarrow A$ to denote the operation by uniformly choosing an element a from a finite set A or generating a sample according to a distribution A. We let $\mathbb{Z}_q$ denote the set $\mathbb{Z} \cap (-q/2, q/2]$ for the positive integer q. We use the notation $[t]_p$ to denote the integer in $\mathbb{Z}_p$ congruent to t mod p. We define $\mathsf{CRT}_{(p_1,p_2,\ldots,p_n)}(r_1, r_2, \ldots, r_n)$ (or abbreviated as $\mathsf{CRT}_{(p_i)}(r_i)$) for pairwise co-prime integers $p_1, p_2, \ldots, p_n$ as the integer in $\left(-\frac{1}{2}\prod_{i=1}^{n} p_i, \frac{1}{2}\prod_{i=1}^{n} p_i\right]$ congruent to r_i in the modulus p_i for each $i \in \{1, 2, \ldots, n\}$.

We use bold letters to denote vectors or matrices and denote the set of all $m \times n$ matrices over $\mathbb{Z}$ by $\mathbb{Z}^{m\times n}$. For matrix $\mathbf{A}$, we denote the transpose of $\mathbf{A}$ by $\mathbf{A}^T$ and denote the i-th row vector of $\mathbf{A}$ by $[\mathbf{A}]_i$. When $\mathbf{A} = (a_{i,j}) \in \mathbb{Z}^{m\times n}$ is given, we define the infinite norm $\|\mathbf{A}\|_\infty$ as $\max_{1\le j\le n} \sum_{i=1}^{n} |a_{i,j}|$ and use the notation $\mathbf{A}$ mod N to denote the matrix $([a_{i,j}]_N) \in \mathbb{Z}^{m\times n}$. We denote by $\mathsf{diag}(a_1, \ldots, a_n)$ the diagonal matrix with diagonal coefficients $a_1, \ldots, a_n$. When $\mathbf{b}$ is an integral matrix, we define $size(\mathbf{b})$ as the logarithm of the largest entries of $\mathbf{b}$.

For a vector $\mathbf{v} = (v_1, \ldots, v_n)$, we define the ℓ_2-norm $\|\mathbf{v}\|_2$ (or abbreviated as $\|\mathbf{v}\|$) and ℓ_1-norm $\|\mathbf{v}\|_1$ as $\sqrt{\sum_{i=1}^{n} {v_i}^2}$ and $\sum_{i=1}^{n} |v_i|$, respectively.

2.1 Lattices

A lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. We call a set of linearly independent vectors $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m\} \subset \mathbb{R}^n$ a basis of a lattice Λ if Λ is the set of all $\mathbb{Z}$-linear combinations of the vectors $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m$. We denote such lattice Λ generated by the basis $\mathbf{b}$ by $\Lambda(\mathbf{b})$. We sometimes use the notation Λ as abbreviated, instead of $\Lambda(\mathbf{b})$. In particular, when a lattice Λ is a subset of $\mathbb{Z}^n$, it is called an integral lattice. In this work, we only take into account the integral lattice and regard a lattice as an integral lattice without special mention. If we regard a basis $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m\}$ of lattice Λ as a matrix whose column vectors consist of vectors $\mathbf{b}_i$ for $1 \le i \le m$, $\mathbf{b}$ is called a basis matrix of Λ. The rank and determinant of lattice Λ is defined as m and $\det(\Lambda) = \sqrt{\det(\mathbf{b}^T\mathbf{b})}$ for any basis matrix $\mathbf{b}$, respectively. When $n = m$, this lattice is called a full-rank lattice and $\det(\Lambda) = \det(\mathbf{b})$ holds. Throughout this paper, we denote lattice Λ whose basis vectors are $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m$ as $\Lambda = \langle \mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m \rangle$.

It is known that for a lattice $\Lambda = \Lambda(\mathbf{b}) \in \mathbb{R}^n$ with basis $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m\}$, the following premise holds:

$$\det(\Lambda) \le \prod_{i=1}^{m} \|\mathbf{b}_i\|$$

In addition, when a set of column vectors $\mathbf{u} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} \subset \mathbb{Z}^n$ is given, we define the orthogonal lattices

$$\Lambda^{\perp}(\mathbf{u}) := \{\mathbf{v} \in \mathbb{Z}^n \mid \langle \mathbf{v}, \mathbf{u}_j \rangle = 0 \textit{ for all } 1 \le j \le k\}.$$
$$\Lambda_q^{\perp}(\mathbf{u}) := \{\mathbf{v} \in \mathbb{Z}^n \mid \langle \mathbf{v}, \mathbf{u}_j \rangle \equiv 0 \bmod q \textit{ for all } 1 \le j \le k\}.$$

Successive Minima

Let Λ be a lattice of rank n. The successive minima of Λ are $\lambda_1, \cdots, \lambda_n \in \mathbb{R}$ and λ_i is minimal for any $1 \le i \le n$ such that there exist i linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_i \in \Lambda$ with $\|\mathbf{v}_j\| \le \lambda_i$ for $1 \le j \le i$.

In order to reduce the size of successive minima, the Gaussian Heuristic [1] is deemed effective.

Gaussian Heuristic

Let Λ be a rank-n lattice. The Gaussian Heuristic states that the size of successive minima of Λ is approximately as follows.

$$\lambda_i(\Lambda) \approx \sqrt{\frac{n}{2\pi e}} \det(\Lambda)^{1/n} \quad \text{for all} \quad i \in \{1, 2, \cdots, n\}.$$

Ajtai showed that the above equation holds for a random lattice with overwhelming probability [1].

Finding a short vector of a lattice is essential in our attack. There are some algorithms to find a short vector of a lattice, which is called lattice reduction algorithms.

Lattice Reduction Algorithm

The LLL algorithm [17] and the BKZ algorithm [15] are well-known lattice reduction algorithms. We mainly use BKZ algorithms to find an approximately short vector of a lattice. According to [15], the block size β of the BKZ algorithm determines how short should the output vector of the BKZ algorithm be. With the BKZ algorithm to the rank-n lattice Λ with basis matrix $\mathbf{B}$, we can achieve a short vector $\mathbf{v}$ in poly$(n, size(\mathbf{B})) \cdot \mathcal{C}_{HKZ}(\beta)$ times which satisfies the following

$$\|\mathbf{v}\| \leq \min\{2(\gamma_\beta)^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \cdot (\det \Lambda)^{1/n}, 4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3} \cdot \lambda_1(\Lambda)\},$$

where $\gamma_\beta \leq \beta$ is the Hermite constant of a rank-β lattice and $\mathcal{C}_{HKZ}(\beta)$ denotes the time spent to get the shortest vector of a rank-β lattice and can be regarded as $2^{O(\beta)}$.

In the case of LLL algorithm, according to [17], the LLL algorithm upon the rank-n lattice Λ with basis matrix $\mathbf{B}$ gives an LLL-reduced basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$ in poly$(n, size(\mathbf{B}))$ times which satisfies the following

$$\|\mathbf{b}_1\| \leq 2^{\frac{n-1}{4}} \cdot (\det \Lambda)^{1/n}, \ \|\mathbf{b}_i\| \leq 2^{\frac{n-1}{2}} \cdot \lambda_i(\Lambda) \text{ for } 1 \leq i \leq n.$$

In particular, it is known that a LLL-reduced basis $\{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m\} \subset \mathbb{R}^n$ with $\delta = 1/4 + 1/\sqrt{2} \approx 0.957$ for a lattice Λ, the following holds

$$\|\mathbf{b}_j\| \leq 2^{i/4} \cdot \|\mathbf{b}_i^*\| \text{ for } 1 \leq j \leq i \leq m. \tag{1}$$

when we let $\{\mathbf{b}_1^*, \cdots \mathbf{b}_m^*\}$ be the Gram-Schmidt orthogonalization.

For the convenience of calculation, throughout this paper, we use $\mathcal{A}_\delta$ to denote a lattice reduction whose output contains a short vector $\mathbf{v}$ with Euclidean norm less than $\delta^n \cdot \det(\Lambda)^{1/n}$ or $\delta^{2n} \cdot \lambda_1(\Lambda)$ for an n-dimensional lattice Λ instead of $2(\gamma_\beta)^{\frac{n-1}{2(\beta-1)}+\frac{3}{2}} \cdot (\det \Lambda)^{1/n}$ or $4(\gamma_\beta)^{\frac{n-1}{\beta-1}+3} \cdot \lambda_1(\Lambda)$, respectively. In this case, the root Hermite factor δ is achieved in time $2^{O(1/\log \delta)} \cdot poly(k)$ by the BKZ algorithm with block size $\beta = \Theta(\frac{1}{\log \delta})$.

From now on, we would like to present the formal definition of the CCK-ACD problem, which is a major concern of this paper.

Definition 1. **(**CCK-ACD**)** *Let γ, n, η, ρ be positive integers such that χ_ρ be an uniform distribution over $\mathbb{Z} \cap (-2^\rho, 2^\rho)$. For η-bit primes $p_1, \cdots, p_n$, the sampleable distribution $\mathcal{D}_{\gamma,\eta,\rho,n}(p_i)$ is defined as*

$$\mathcal{D}_{\gamma,\eta,\rho,n}(p_i) = \{T \cdot \prod_{i=1}^{n} p_i + \mathsf{CRT}_{(p_i)}(r_i) \mid T \leftarrow \mathbb{Z} \cap [2^{\gamma-1}/\prod_{i=1}^{n} p_i, 2^\gamma/\prod_{i=1}^{n} p_i), \ r_i \leftarrow \chi_\rho\}.$$

*The (γ, η, ρ)-*CCK-ACD *problem is: Given $N = p_0 \prod_{i=1}^{n} p_i$ for uniformly chosen $p_0 \in \mathbb{Z} \cap [2^{\gamma-1}/\prod_{i=1}^{n} p_i, \ 2^\gamma/\prod_{i=1}^{n} p_i)$ and polynomially many samples from $\mathcal{D}_{\gamma,\eta,\rho,n}(p_i)$ or χ_γ, distinguish* CCK-ACD *samples from random samples.*

In the CCK-ACD problem, we use $r_{0,j}$ to denote $b_j \bmod p_0$ for each $j \in \{1, \cdots, k\}$, where $b_j \in \mathcal{D}_{\gamma,\eta,\rho,n}(p_i)$'s are given as CCK-ACD samples. We remark that $r_{0,j}$ may not be small, unlike other $r_{i,j}$ for $i \in \{1, \cdots, n\}$.

3 OLA for the CCK-ACD Problem

In this section, we revisit the orthogonal lattice attack method (OLA) and explain how to guarantee the upper bound of the OLA proposed in [9] for the CCK-ACD problem in time $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$.

Our extended OLA algorithm outputs a determinant of certain lattice, which is constructed by CCK-ACD samples or random integers. In this section, for the CCK-ACD samples, we show that the size of determinant is bounded by $2^{\frac{n+1}{4}+n(\rho+\log k)}$, where k denotes the optimized number of CCK-ACD samples, under the Gaussian Heuristic. In the case of random elements, our algorithm outputs a determinant larger than the value. From the results, we can solve the CCK-ACD problem by checking the determinant. The full details of our OLA algorithm shall be given in full in the below.

3.0.1 Analysis of CCK-ACD instances.

Assume that we have k CCK-ACD samples $\{b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})\}_{1\le j\le k}$ with $N = \prod_{i=0}^{n} p_i$, and let $\mathbf{b} = (b_1, \cdots, b_k)^T$, $\mathbf{r}_i = (r_{i,1}, \cdots, r_{i,k})^T$ for $0 \le i \le n$.

The first step of OLA, which is described in [9, Section 5.1], is to find the set of short vectors $\{\mathbf{u}_1, \cdots, \mathbf{u}_{k-n-1}\}$ in a k-dimensional lattice

$$\Lambda_N^{\perp}(\mathbf{b}) = \{\mathbf{u} \in \mathbb{Z}^k \mid \langle \mathbf{u}, \mathbf{b}\rangle \equiv 0 \mod N\}.$$

Since $b_j \equiv r_{i,j} \bmod N$, we observe the relations using the CRT structure

$$\begin{pmatrix} r_{0,1} & r_{1,1} & \cdots & r_{n,1} \\ r_{0,2} & r_{1,2} & \cdots & r_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{0,k} & r_{1,k} & \cdots & r_{n,k} \end{pmatrix} \cdot \begin{pmatrix} (\hat{p}_0^{-1} \bmod p_0)\cdot \hat{p}_0 \\ (\hat{p}_1^{-1} \bmod p_1)\cdot \hat{p}_1 \\ \vdots \\ (\hat{p}_n^{-1} \bmod p_n)\cdot \hat{p}_n \end{pmatrix} \equiv \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} \bmod N.$$

If a vector $\mathbf{u} \in \mathbb{Z}^k$ satisfies $\langle \mathbf{u}, \mathbf{r}_i\rangle = 0$ in integers for all $i = 0, \cdots, n$, then $\langle \mathbf{u}, \mathbf{b}\rangle \equiv 0 \bmod N$ because of the above relations. Thus, it holds that

$$\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\}) \subset \Lambda_N^{\perp}(\mathbf{b})$$

Moreover, we observe $\lambda_i(\Lambda_N^{\perp}(\mathbf{b})) \le \lambda_i(\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\}))$ by the definition of successive minima for all $1 \le i \le k-n-1$.

We assume the Gaussian heuristic holds on the lattice $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$ since all components of $\mathbf{r}_i$ with $0 \le i \le n$ are uniformly chosen from each set. Therefore, it holds that

$$\log|\lambda_i(\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\}))| = \frac{\gamma - n\eta + n\rho}{k-n-1}$$

for all $i = 1, 2, \cdots, k-n-1$. Note that we omit the small values including $\log k$ for the convenience of writing.

We aim at recovering generators of $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$. To obtain such vectors $\mathbf{u}_j$'s, we run a lattice reduction algorithm $\mathcal{A}_\delta$ with root Hermite factor δ on the lattice $\Lambda_N^{\perp}(\mathbf{b})$. By the approximate factor δ of a lattice reduction algorithm $\mathcal{A}_\delta$, the j-th output vector $\mathbf{u}_j$ of $\mathcal{A}_\delta$ on the lattice $\Lambda_N^{\perp}(\mathbf{b})$ satisfies $\|\mathbf{u}_j\| \le \delta^{2k}\cdot\lambda_j(\Lambda_N^{\perp}(\mathbf{b}))$. Thus, for all $j = 1, 2, \cdots, k-n-1$, $\mathbf{w}_j$ is bounded as follows

$$\begin{aligned}\|\mathbf{u}_j\| &\le \delta^{2k}\cdot\lambda_j(\Lambda_N^{\perp}(\mathbf{b})) \le \delta^{2k}\cdot\lambda_i(\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})) \\ &\le \delta^{2k}\cdot 2^{\frac{\gamma-n\eta+n\rho}{k-n-1}}.\end{aligned}$$

We now argue that the vector $\mathbf{u}_j$ is in $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$ under some condition. Since we know $\|\mathbf{r}_i\| \le k\cdot 2^{\rho}$, it holds for $1 \le i \le n$, $1 \le j \le k-n-1$

$$\begin{aligned}|\langle \mathbf{u}_j, \mathbf{r}_i\rangle| &\le \|\mathbf{w}_j\|\cdot\|\mathbf{r}_i\| \\ &< (\delta^{2k}\cdot 2^{\frac{\gamma-n\eta+n\rho}{k-n-1}})\cdot(k\cdot 2^{\rho}) \\ &= 2^{2k\log\delta+\frac{\gamma-n\eta+n\rho}{k-n-1}}\cdot 2^{\rho+\log k}.\end{aligned}$$

The last value of the equation includes $\log k$, which is however much smaller than the current value. Therefore, for simplicity purposes, we omit this term.

By the CRT construction, we have $\langle \mathbf{u}_j, \mathbf{b}\rangle \equiv_{p_i} \langle \mathbf{u}_j, \mathbf{r}_i\rangle$ and it is zero in modulo p_i for all i. Since p_i's are η-bit primes, we can therefore ensure the vector $\mathbf{u}_j \in \Lambda_N^{\perp}(\mathbf{b})$ if $|\langle \mathbf{u}_j, \mathbf{r}_i\rangle| < p_i/2$ for all i. This condition can be written as

$$2k \cdot \log\delta + \frac{\gamma - n\eta + n\rho}{k-n-1} + \rho \le \eta, \tag{2}$$
$$2(k-n-1)\cdot\log\delta + \frac{\gamma}{k-n-1} \le \frac{(k-1)(\eta-\rho)}{k-n-1} - 2(n+1)\log\delta.$$

When we choose $k - n - 1 = \sqrt{\frac{\gamma}{2\log\delta}}$ and apply the AM-GM inequality, it is enough to satisfy $\log\delta$ as following inequality

$$2\sqrt{2\gamma\log\delta} \le \eta - \rho.$$

Therefore, when we obtain $k = n + 1 + \sqrt{\frac{\gamma}{2\log\delta}}$ CCK-ACD samples and choose δ satisfying $\log\delta < \frac{(\eta-\rho)^2}{8\gamma}$, $|\langle \mathbf{u}_j, \mathbf{r}_i\rangle| \le p_i/2$ is established for any $1 \le i \le n$. Thus, $\{\mathbf{u}_j\}_{1\le j\le k-n-1}$ are the generator set of $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$ under the condition.

The overall time complexity to recover the generator of $\Lambda^{\perp}(\{\mathbf{r}_0, \cdots, \mathbf{r}_n\})$ is $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)} \cdot poly(k) = 2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ up to polynomial factors.

As the second step, we construct a new lattice that is orthogonal to the lattice generated by $\{\mathbf{u}_j\}$, instead of opting in direct calculation of $\mathbf{r}_i$. More precisely, let $\tilde{\mathbf{U}}$ denote a matrix $(\mathbf{u}_1 \mid \cdots \mid \mathbf{u}_{k-n-1})$ and consider the orthogonal lattice

$$\Lambda^{\perp}(\tilde{\mathbf{U}}) = \{\mathbf{v} \in \mathbb{Z}^{\mathbf{k}} \mid \langle \mathbf{v}, \mathbf{u_j}\rangle = \mathbf{0} \textit{ for all } \mathbf{1} \le \mathbf{j} \le \mathbf{k} - \mathbf{n} - \mathbf{1}\}.$$

Due to the CRT-structure of CCK-ACD samples, $\{\mathbf{r}_i\}_{1\le i\le n}$ are short linearly independent vectors that belong to $\Lambda^{\perp}(\tilde{\mathbf{U}})$.[2] The lattice $\Lambda^{\perp}(\tilde{\mathbf{U}}) \subset \mathbb{Z}^k$ has rank $n+1$. We apply the LLL algorithm on the lattice $\Lambda^{\perp}(\tilde{\mathbf{U}})$ to obtain $\mathbf{b}' = \{\mathbf{b}'_1, \cdots, \mathbf{b}'_{n+1}\}$, the LLL-reduced basis. In this case, the time complexity is $poly(n, size(\Lambda^{\perp}(\tilde{\mathbf{U}})))$, which is dominated by $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$.

We now show that determinant of $\Lambda(\{\mathbf{b}'_1, \ldots, \mathbf{b}'_n\})$ is bounded. Since $\{\mathbf{r}_i\}_{1\le i\le n}$ are n linearly independent vectors in $\Lambda^{\perp}(\tilde{\mathbf{U}})$, there exists a vector $\mathbf{b}'_j$ such that $\{\mathbf{r}_1, \cdots, \mathbf{r}_n, \mathbf{b}'_j\}$ are $n+1$ linearly independent vectors in $\Lambda^{\perp}(\tilde{\mathbf{U}})$. Additionally, $\|\mathbf{r}_i\|$ is smaller than $k \cdot 2^{\rho}$ for all i, we note that $\lambda_n(\Lambda^{\perp}(\tilde{\mathbf{U}})) \le 2^{\rho+\log k}$. Let $\tilde{\mathbf{b}}' = \{\mathbf{b}'^{\star}_1, \cdots, \mathbf{b}'^{\star}_{n+1}\}$ be Gram-Schmidt basis of $\mathbf{b}'$. Then we calculate the determinant of lattice Λ' spanned by $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$.

$$\begin{aligned}
\det(\Lambda') &= \prod_{i=1}^{n} \|\mathbf{b}'^{\star}_i\| = \frac{\prod_{i=1}^{n+1}\|\mathbf{b}'^{\star}_i\|}{\|\mathbf{b}'^{\star}_{n+1}\|} \\
&= \frac{\det(\Lambda^{\perp}(\tilde{\mathbf{U}}))}{\|\mathbf{b}'^{\star}_{n+1}\|} \le \frac{\|\mathbf{b}'_j\| \cdot \prod_{i=1}^{n}\|\mathbf{r}_i\|}{\|\mathbf{b}'^{\star}_{n+1}\|} \\
&\le \|\mathbf{b}'_j\| \cdot \prod_{i=1}^{n}\|\mathbf{r}_i\| \cdot \frac{2^{\frac{n+1}{4}}}{\|\mathbf{b}'_j\|} \text{ (By inequality (1))} \\
&\le 2^{\frac{n+1}{4} + n(\rho+\log k)}
\end{aligned}$$

According to the analysis above, the log-size of determinant of the rest n column vectors after LLL algorithm is smaller than $\frac{n+1}{4} + n(\rho + \log k)$ with $k - n - 1 = \sqrt{\frac{\gamma}{2\log\delta}} = \frac{2\gamma}{\eta-\rho}$.

3.0.2 Heuristic analysis of random instances.

Assume that we have k random samples and run the same algorithm on the random samples. To analyze the size of determinant heuristically, we first assume that the logarithm of determinant of rank-n lattice is

2 We can assume that $\{\mathbf{r}_i : 1 \le i \le n\}$ are n linearly independent vectors because their entries are randomly chosen in χ_ρ.

approximately $n \log B$, when each entry of a basis matrix is uniformly sampled from $[-2^B, 2^B]$. This approximation agrees the bound from Hadamard inequality, and for square matrix it is known to hold up to difference $\Theta(n \log n)$ assuming that entries are uniform [18]. In our case, $n \log n$ is negligibly small compared to other terms.

– Random instances: As a former cases, we consider a lattice

$$\Lambda_N^{\perp}(\mathbf{b}) = \{\mathbf{u} \in \mathbb{Z}^k \mid \langle \mathbf{u}, \mathbf{b} \rangle \equiv 0 \mod N\}$$

with random integers $\mathbf{b}_i$. Next we run a lattice reduction algorithm on $\Lambda_N^{\perp}(\mathbf{b})$. The expected size of $\mathbf{u}_j$, the j-th output of the lattice reduction algorithm, are $\delta^k \cdot N^{1/k}$ for all $1 \le j \le k - n - 1$. We may suppose these vectors are random, given that the instances are random. Then, the logarithm of the determinant of a lattice $\Lambda(\tilde{\mathbf{U}})$ generated by $\{\mathbf{u}_1, \ldots, \mathbf{u}_{k-n-1}\}$ is approximately

$$\frac{k-n-1}{k} \log N + (k-n-1)k \log \delta \approx \frac{k-n-1}{k} \cdot \gamma.$$

Since the second term is relatively smaller than the first term, we will only handle the last term. The assumption that the basis vector of $\Lambda(\tilde{\mathbf{U}})$ is random also allows that $\det(\Lambda(\tilde{\mathbf{U}}))$ and $\det(\Lambda^{\perp}(\tilde{\mathbf{U}}))$ are the same. Then we obtain the desired result that the logarithm of determinant of $\Lambda^{\perp}(\tilde{\mathbf{U}})$ is approximately $\gamma \cdot \frac{k-n-1}{k}$. Then, the expected size of vectors obtained as a result of the LLL algorithm shall be $2^{n/4} \cdot \det(\Lambda^{\perp}(\tilde{\mathbf{U}}))^{\frac{1}{n+1}}$. Then the logarithm of determinant of the matrix composed by any n vectors is approximately

$$\frac{n}{n+1} \cdot \frac{k-n-1}{k} \cdot \gamma + \frac{n^2}{4}.$$

In summary, under Gaussian Heuristic and assumption from Hadamard inequality, we show that the logarithm of the determinant is less than $\frac{n+1}{4} + n(\rho + \log k) = O(n \cdot \rho)$ if the given instances are the CCK-ACD instances whereas it is asypmtotically $\gamma \cdot \frac{k-n-1}{k} \cdot \frac{n}{n+1} = \Omega(\gamma)$ for the random instances. Hence, if those two values do not overlap, we can solve the CCK-ACD problem in $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ time complexity. We will later see if the experimental results fit well with this approximation in Section 5. From the analysis, we have the following result,

Theorem 2 (Heuristic). *Let n, η, ρ be parameters of the* CCK-ACD *problem and $k = n + 1 + \sqrt{\frac{\gamma}{2 \log \delta}}$* CCK-ACD *samples are given with $\log \delta < \frac{(\eta-\rho)^2}{8\gamma}$. When the following equation holds*

$$\frac{n+1}{4} + n(\rho + \log k) < \gamma \cdot \frac{k-n-1}{k} \cdot \frac{n}{n+1},$$

one can solve the CCK-ACD *problem in $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ time complexity.*

The following is our extended OLA algorithm.

4 SDA Algorithm for the CCK-ACD problem

In this section, we first describe a lattice-based algorithm to solve the CCK-ACD problem by applying the Simultaneously Diophantine approximation (SDA) algorithm which has served as a useful method to solve the ACD problem. Compared to the OLA algorithm, SDA algorithm allows us to recover all secret primes p_i of CCK-ACD problem. Therefore, in this section, we will take into account a search CCK-ACD problem instead of decisional one.

In the paper [14], Galbraith *et al.* try to apply the SDA algorithm in the context of CCK-ACD and comment that this attack is not directly applicable to the CCK-ACD problem. In order to review this work, one can con-

Algorithm 1 OLA Algorithm for the CCK-ACD problem

Input: γ-bit integer $N = \prod_{i=0}^{n} p_i$

Input: Root Hermite factor δ

Input: $\mathbf{b} = (b_1, b_2, \cdots, b_k)$, where $k = n + \left\lfloor \sqrt{\frac{\gamma}{2\log\delta}} \right\rceil$

Output: distinguish whether b_i's are sampled from $\mathcal{D}_{\gamma,\eta,\rho,n}(p_i)$ or a χ_γ.

1: Construct a lattice $\Lambda_N^{\perp}(\mathbf{b})$ with the following basis matrix

$$\mathbf{U} = \begin{pmatrix} N & [-b_2/b_1]_N & \cdots & [-b_k/b_1]_N \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

2: Call the lattice reduction algorithm with a Hermite factor δ on $\mathbf{U}$, and let $(\mathbf{u}_i)$ denote the output.

3: Compute an orthogonal lattice $\Lambda^{\perp}(\tilde{\mathbf{U}})$ for $\tilde{\mathbf{U}} = (\mathbf{u}_1 \mid \cdots \mid \mathbf{u}_{k-n-1})$ and let $\tilde{\mathbf{b}}$ denote the its basis matrix.

4: Call the LLL algorithm on $\tilde{\mathbf{b}}$ and let $(\tilde{\mathbf{b}}_i)$ denote the output.

5: Find $\tilde{\mathbf{b}}_j \in \{\tilde{\mathbf{b}}_i\}$ such that $\|\tilde{\mathbf{b}}_j\| = \max_i \|\tilde{\mathbf{b}}_i\|$ and let $\Lambda(\tilde{\mathbf{b}})$ denote a lattice generated by $\{\tilde{\mathbf{b}}_i\} \setminus \tilde{\mathbf{b}}_j$.

6: **if** $\log(\det(\Lambda(\tilde{\mathbf{b}}))) \le \frac{n+1}{4} + n(\rho + \log k)$ **then**

7: **return** $\mathcal{D}_{\gamma,\eta,\rho,n}(p_i)$

8: **else**

9: **return** χ_γ.

10: **end if**

sider a column lattice Λ generated by a matrix $\mathbf{B}$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ b_1 & N & 0 & \cdots & 0 \\ b_2 & 0 & N & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_k & 0 & 0 & \cdots & N \end{pmatrix}$$

with given CCK-ACD samples $b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})$ for each $1 \le j \le k$ and $N = \prod_{i=0}^{n} p_i$. It follows that the lattice contains the short vectors

$$\mathbf{v}_i = \hat{p}_i \cdot (1, r_{i,1}, r_{i,2}, \cdots, r_{i,k})^T.$$

for all $1 \le i \le n$ and these all have similar lengths. Once we compute $\hat{p}_i$ from the first entry of the vector, we can recover the prime factors $p_i = N/\hat{p}_i$. But if $\mathbf{u} = (u_0, u_1, \cdots, u_k) \in \Lambda$ is a short linear combination of several of these vectors, (*i.e.*, $\mathbf{u} = \sum_{i=1}^{n} e_i \cdot \mathbf{v}_i$), we cannot expect that $\lfloor N/u_0 \rceil$ is one of the primes of N, where $u_0 = \sum_{i=1}^{n} e_i \cdot \hat{p}_i$. That is why the origianl SDA algorithm is not directly applicable to the CCK-ACD problem.

However, an instance of the form $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ with small d_i's has a special property. This integer is called dual instance in this paper. More precisely, if we can ensure that d_i's are sufficiently small, the instance $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ allows the below modular equations to be established without modulus N due to the CRT-structure of CCK-ACD samples.

$$[d \cdot b_j]_N = [\sum_{i=1}^{n} d_i \cdot b_j \cdot \hat{p}_i]_N = \sum_{i=1}^{n} d_i \cdot r_{i,j} \cdot \hat{p}_i \in \mathbb{Z}$$

$$[d \cdot b_j \cdot b_l]_N = \sum_{i=1}^{n} d_i \cdot r_{i,j} \cdot r_{i,l} \cdot \hat{p}_i \in \mathbb{Z}$$

This property plays a crucial role in solving the CCK-ACD problem and even recovering the secret primes in our algorithm. In Section 4.1, we give a formal definition of a dual instance to give a standard for how small d_i's should be in an instance $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$. Once we obtain such dual instances, we modify Cheon *et*

al.'s algorithm in [5] to solve the CCK-ACD problem using the dual instances, which is the second step of our algorithm for solving the CCK-ACD problem.

All in all, we first obtain a dual instance from the original SDA algorithm. Next we recover any secret primes p_i by applying the modified Cheon's algorithm. For convenience purposes, the second step will be firstly described and the first step will be suggested later. In the below, the full details of an extended SDA algorithm will be explained.

4.1 Revisiting the Algorithm of Cheon *et al.*

In this section, we revisit the Cheon *et al.*'s algorithm in [5] to solve the CCK-ACD problem. In the original paper, the authors presented an algorithm when an auxiliary input $\mathsf{CRT}_{(p_i)}(\hat{p}_i) = \sum_{i=1}^{n} \hat{p}_i$ is given.

However, in order to use an instance $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ in Cheon *et al.*'s algorithm, all of d_i's does not necessarily be 1. If d_i's are sufficiently small, $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ can also play the same role as an auxiliary input. From this, we define a dual instance for the CCK-ACD problem, which is a generalization of an auxiliary input and introduce a polynomial-time algorithm to solve the CCK-ACD problem when two dual instances are given instead of one auxiliary input by slightly modifying Cheon *et al.*'s algorithm.

Definition 3 (Dual Instance). *Let n, η, ρ be positive integers. For given η-bit primes $p_1, \cdots, p_n$ and $p_0 \in \mathbb{Z} \cap [2^{\gamma-1}/\prod_{i=1}^{n} p_i, 2^{\gamma}/\prod_{i=1}^{n} p_i)$ in* CCK-ACD, *define $N = \prod_{i=0}^{n} p_i$ and $\hat{p}_i = N/p_i$, for $0 \le i \le n$. We define a dual instance d as the integer which can be written as $d = \sum_{i=0}^{n} d_i \cdot \hat{p}_i$ for some integers d_i's satisfying $|d_i| \le p_i \cdot 2^{-2\rho-\log n-1}$ for each $1 \le i \le n$ and $d_0 = 0$.*

An algorithm to generate a dual instance when given polynomially many CCK-ACD samples will be described in Section 4.2.

For an integer $d = \sum_{i=0}^{n} d_i \cdot \hat{p}_i$ and CCK-ACD samples $b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})$ and $b_l = \mathsf{CRT}_{(p_i)}(r_{i,l})$, one can see the followings

$$[d]_N \equiv \sum_{i=0}^{n} d_i \cdot \hat{p}_i \pmod{N}, \tag{3}$$

$$[d \cdot b_j]_N \equiv \sum_{i=0}^{n} d_i \cdot r_{i,j} \cdot \hat{p}_i \pmod{N}, \tag{4}$$

$$[d \cdot b_j \cdot b_l]_N \equiv \sum_{i=0}^{n} d_i \cdot r_{i,j} \cdot r_{i,l} \cdot \hat{p}_i \pmod{N}. \tag{5}$$

Under the condition in which each size of d_i is sufficiently small for $1 \le i \le n$ and $d_0 = 0$, the above equations hold over the integers, not modulo N. In other words, for a dual instance $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ defined as above, the following inequalities hold

$$|d_i \cdot \hat{p}_i| = |d_i| \cdot \frac{N}{p_i} < N \cdot 2^{-2\rho-\log n-1},$$

$$|\sum_{i=0}^{n} d_i \cdot r_{i,j} \cdot r_{i,k} \cdot \hat{p}_j| \le \sum_{i=1}^{n} |r_{i,j}| \cdot |r_{i,k}| \cdot |d_i \cdot \hat{p}_i| \le \sum_{i=1}^{n} N \cdot 2^{-\log n-1} \le N/2.$$

Thus, we observe the right of the three equations (3), (4) and (5) have the size less than $N/2$ so that those equations hold over the integer. Now we show how to solve the CCK-ACD when given polynomially many CCK-ACD samples and two distinct dual instances $d = \sum_{i=0}^{n} d_i \cdot \hat{p}_i$ and $d' = \sum_{i=0}^{n} d'_i \cdot \hat{p}_i$. This computation is quite similar to the Cheon's algorithm [5]. More precisely, we are $2n$ CCK-ACD samples: $b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})$ and $b'_\ell = \mathsf{CRT}_{(p_i)}(r'_{i,\ell})$ for $1 \le j, \ell \le n$. We denote $w_{j,\ell}$ and $w'_{j,\ell}$ as $[d \cdot b_j \cdot b'_\ell]_N$ and $[d' \cdot b_j \cdot b'_\ell]_N$, respectively. Thanks

to the dual instance properties, then it can be written as

$$w_{j,\ell} = \sum_{i=1}^{n} r_{i,j} \cdot (d_i \cdot \hat{p}_i) \cdot r'_{i,\ell} = \begin{pmatrix} r_{1,j} & r_{2,j} & \cdots & r_{n,j} \end{pmatrix} \begin{pmatrix} d_1 \cdot \hat{p}_1 & 0 & \cdots & 0 \\ 0 & d_2 \cdot \hat{p}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \cdot \hat{p}_n \end{pmatrix} \begin{pmatrix} r'_{1,\ell} \\ r'_{2,\ell} \\ \vdots \\ r'_{n,\ell} \end{pmatrix},$$

$$w'_{j,\ell} = \sum_{i=1}^{n} r_{i,j} \cdot (d'_i \cdot \hat{p}_i) \cdot r'_{i,\ell} = \begin{pmatrix} r_{1,j} & r_{2,j} \cdots & r_{n,j} \end{pmatrix} \begin{pmatrix} d'_1 \cdot \hat{p}_1 & 0 & \cdots & 0 \\ 0 & d'_2 \cdot \hat{p}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d'_n \cdot \hat{p}_n \end{pmatrix} \begin{pmatrix} r'_{1,\ell} \\ r'_{2,\ell} \\ \vdots \\ r'_{n,\ell} \end{pmatrix}.$$

By collecting the above values of several $1 \le j, \ell \le n$, we can construct two matrices $\mathbf{w} = (w_{j,\ell})$ and $\mathbf{w}' = (w'_{j,\ell}) \in \mathbb{Z}^{n \times n}$, which can be written as

$$\mathbf{w} = \mathbf{r}^T \cdot \mathsf{diag}(d_1 \cdot \hat{p}_1, \cdots, d_n \cdot \hat{p}_n) \cdot \mathbf{r}'$$

$$\mathbf{w}' = \mathbf{r}^T \cdot \mathsf{diag}(d'_1 \cdot \hat{p}_1, \cdots, d'_n \cdot \hat{p}_n) \cdot \mathbf{r}'$$

for $\mathbf{r} = (r_{j,i})$ and $\mathbf{r}' = (r'_{i,\ell}) \in \mathbb{Z}^{n \times n}$. By computing $(\mathbf{w}')^{-1}$ over $\mathbb{Q}$, we obtain the matrix $\mathbf{Y}$ as following form

$$\mathbf{Y} = \mathbf{w} \cdot (\mathbf{w}')^{-1} = \mathbf{r}^T \cdot \mathsf{diag}(d_1/d'_1, \cdots, d_n/d'_n) \cdot (\mathbf{r}^T)^{-1}$$

whose eigenvalues are exactly the set $\{d_1/d'_1, \cdots, d_n/d'_n\} \subset \mathbb{Q}$. We can compute those rational eigenvalues in polynomial-time of η, n and ρ from $\mathbf{Y}$. Since the modular equations $d \equiv d_i \cdot \hat{p}_i \pmod{p_i}$ and $d' \equiv d'_i \cdot \hat{p}_i \pmod{p_i}$ hold, one can check that p_i divides $d \cdot d'_i - d' \cdot d_i$ for each i. Thus, by computing $\gcd(N, d \cdot d'_i - d' \cdot d_i)$, we can find the p_i for each $1 \le i \le n$. Considering the required cost of the computations required, we obtain the following theorem.

Theorem 4. *(Adapted from [5, Section 3.2]) For given $O(n)$* CCK-ACD *samples from $\mathcal{D}_{\eta,\rho,n}(p_i)$ with $N = \prod_{i=1}^{n} p_i$ and two distinct* dual instances*, one can recover secret primes $p_1, \cdots, p_n$ in $\widetilde{O}(n^{2+\omega} \cdot \eta)$ time with $\omega \le 2.38$ and overwhelming probability in ρ.*

4.2 Generating a Dual Instance from SDA

In this section, we present an algorithm to generate a dual instance from polynomially many given CCK-ACD samples $b_j = \mathsf{CRT}_{(p_i)}(r_{ij})$ and $N = \prod_{i=0}^{n} p_i$.

Consider the column lattice Λ generated by the following basis matrix.

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ b_1 & N & 0 & \cdots & 0 \\ b_2 & 0 & N & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_k & 0 & 0 & \cdots & N \end{pmatrix}.$$

We confirm that any lattice vector $\mathbf{c} \in \Lambda$ with $\|\mathbf{c}\| \le \frac{N}{2}$ can be written in the form of $([d]_N, [d \cdot b_1]_N, \cdots, [d \cdot b_k]_N)^T$, where $d = \sum_{i=0}^{n} d_i \cdot \hat{p}_i$ for some d_i's and the modular equation $[d \cdot b_j]_N \equiv \sum_{i=0}^{n} r_{i,j} \cdot [d_i]_{p_i} \cdot \hat{p}_i \pmod{N}$ holds for each j. In the next theorem, we prove that if $\mathbf{c} \in \Lambda$ is a sufficiently short vector for a proper integer k, the first entry of the vector $\mathbf{c}$, $\sum_{i=1}^{n} [d_i]_{p_i} \cdot \hat{p}_i$ is a dual instance. Then we will be able to solve the CCK-ACD problem by combining it with the Theorem 4.

Theorem 5. *Let n, η, ρ be parameters of the* CCK-ACD *problem. When $O(\gamma/\eta)$* CCK-ACD *samples are given, one can find a* dual instance *in $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ time up to polynomial factors.*

Proof. Suppose that $k > n$ CCK-ACD samples $b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})$ and $N = \prod_{i=0}^{n} p_i$ are given. We denote $r_{0,j}$ as $[b_j]_{p_0}$. Consider the column lattice Λ generated by the following basis matrix $\mathbf{b}$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ b_1 & N & 0 & \cdots & 0 \\ b_2 & 0 & N & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_k & 0 & 0 & \cdots & N \end{pmatrix},$$

where b_j's are given CCK-ACD samples and $N = \prod_{i=0}^{n} p_i$. Note that any vector $\mathbf{v}$ in the lattice Λ can be represented as the following form

$$\mathbf{v} \equiv a_0 \cdot \hat{p}_0 \begin{pmatrix} 1 \\ r_{0,1} \\ r_{0,2} \\ \vdots \\ r_{0,k} \end{pmatrix} + a_1 \cdot \hat{p}_1 \begin{pmatrix} 1 \\ r_{1,1} \\ r_{1,2} \\ \vdots \\ r_{1,k} \end{pmatrix} + \cdots + a_n \cdot \hat{p}_n \begin{pmatrix} 1 \\ r_{n,1} \\ r_{n,2} \\ \vdots \\ r_{n,k} \end{pmatrix} \pmod{N}.$$

for some integers a_i's. We denote $\hat{p}_i \cdot (1, r_{i,1}, r_{i,2}, \cdots, r_{i,k})^T$ by $\mathbf{v}_i$ for each i. Then, $\mathbf{v}_i$'s are linearly independent and $\|\mathbf{v}_i\| \le B := \sqrt{k+1} \cdot N \cdot 2^{-\eta+\rho+1}$ for all $i \neq 0$, so $\lambda_i(\Lambda) \le B$ holds for $1 \le i \le n$.

We apply Gaussian Heuristic to estimate $\lambda_{n+1}(\Lambda)$ which is approximately $\sqrt{\frac{k+1}{2\pi e}} \cdot (\det \Lambda)^{\frac{1}{k+1}}$. Suppose the size of a vector $\mathbf{c} \in \Lambda$ obtained by the lattice reduction algorithm $\mathcal{A}_\delta$ is shorter than $\delta^{2(k+1)} \cdot \lambda_1(\Lambda) \le \delta^{2(k+1)} \cdot B < \sqrt{\frac{k+1}{2\pi e}} \cdot (\det \Lambda)^{\frac{1}{k+1}} \approx \lambda_{n+1}(\Lambda)$. Then, we conclude $\mathbf{c} \in \langle \mathbf{v}_1, \cdots, \mathbf{v}_n \rangle$ and p_0 divides $\gcd(N, d)$, where d is the first entry of the vector $\mathbf{c}$. Hence, it is required that the length of vector $\mathbf{c}$, the first output of the lattice reduction algorithm, is shorter than $\sqrt{\frac{k+1}{2\pi e}} \cdot (\det \Lambda)^{\frac{1}{k+1}}$. It can be written as

$$\|\mathbf{c}\| \le \delta^{2(k+1)} \cdot \lambda_1(\Lambda) < \delta^{2(k+1)} \cdot B < \sqrt{\frac{k+1}{2\pi e}} \cdot (\det \Lambda)^{\frac{1}{k+1}}.$$

Taking logarithm to both sides of the inequality, we obtain the following:

$$2(k+1)\log\delta + \log N - \eta + \rho + 1 \le \frac{k}{k+1}\log N - \frac{1}{2}\log 2\pi e$$

$$2(k+1)\log\delta + \frac{1}{k+1}\log N < \eta - \rho - \log 2\sqrt{2\pi e} \tag{6}$$

In particular, when applying the AM-GM inequality on the left side of (6), we obtain the following inequality

$$2\sqrt{2\log\delta \cdot \log N} \le \eta - \rho - O(1)$$

where equality holds if and only if $(k+1)^2 = \frac{\gamma}{2\log\delta}$ and $\gamma \cdot \log\delta = \frac{(\eta-\rho)^2}{8}$.

Thus, when we choose δ satisfying $\log\delta < \frac{(\eta-\rho)^2}{8\gamma}$ and $k = \frac{2\gamma}{\eta-\rho} = O(\frac{\gamma}{\eta})$, we can conclude that output vector $\mathbf{c}$ of $\mathcal{A}_\delta$ can be written as $\mathbf{c} = \sum_{i=1}^{n} d_i \cdot \mathbf{v}_i$ for some d_i's. If we denote the first entry of $\mathbf{c}$ as d, the vector $\mathbf{c}$ is the form of $(d, [d \cdot b_1]_N, \cdots, [d \cdot b_k]_N)^T$. Then $d = \sum_{i=1}^{n} d_i \cdot \hat{p}_i$ and $[d \cdot b_j]_N = \sum_{i=1}^{n} r_{i,j} \cdot d_i \cdot \hat{p}_i$ hold for each j. In this case, d is a multiple of p_0 so that one can recover the factor p_0 by computing $\gcd(d, N)$. Since the root Hermite factor

δ is achieved in time $\mathrm{poly}(k)\cdot 2^{O(\beta)}$ times by the BKZ algorithm with $\beta = \Theta(1/\log\delta)$, we conclude that one can recover the factor p_0 in $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ time up to polynomial factors using the BKZ algorithm with $\beta \approx O\left(\frac{\gamma}{(\eta-\rho)^2}\right)$.

Next, we propose the condition for terms d_i's to be sufficiently bounded so that it can be regarded as a dual instance. We denote $\tilde{\mathbf{c}}$ as k-dimensional vector which can be obtained by removing the first coordinate of $\mathbf{c}$ (*i.e.* $\tilde{\mathbf{c}} = ([d\cdot b_1]_N, \cdots, [d\cdot b_k]_N)^T$). Using the property $[d\cdot b_j]_N = \sum_{i=1}^{n} r_{i,j}\cdot d_i\cdot \hat{p}_i$ for each j, $\tilde{\mathbf{c}}$ can be decomposed as follows:

$$\begin{aligned}\tilde{\mathbf{c}} &= (d_1\cdot\hat{p}_1, \cdots, d_n\cdot\hat{p}_n)\cdot\begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k}\\ r_{2,1} & r_{2,2} & \cdots & r_{2,k}\\ \vdots & \vdots & \ddots & \vdots\\ r_{n,1} & r_{n,2} & \cdots & r_{n,k}\end{pmatrix}\\ &= \mathbf{d}\cdot\hat{\mathbf{P}}\cdot\mathbf{r},\end{aligned}$$

where $\mathbf{d} = (d_1, \cdots, d_n)$, $\hat{\mathbf{P}} = \mathrm{diag}(\hat{p}_1, \ldots, \hat{p}_n)$, and $\mathbf{R} = (r_{i,j}) \in \mathbb{Z}^{n\times k}$.

We will later show that there is a right inverse $\mathbf{R}^\star \in \mathbb{Z}^{k\times n}$ such that $\mathbf{R}\cdot\mathbf{R}^\star = \mathbf{I}_n$, where $\mathbf{I}_n$ is the $n\times n$ identity matrix. Then, for each i, $|d_i\cdot\hat{p}_i|$ can be bounded as follows:

$$|d_i\cdot\hat{p}_i| \le \|\mathbf{d}\cdot\hat{\mathbf{P}}\|_\infty = \|\tilde{\mathbf{c}}\cdot\mathbf{R}^\star\|_\infty \le \|\tilde{\mathbf{c}}\|\cdot\|\mathbf{R}^\star\|_\infty.$$

If there is a matrix $\mathbf{R}^\star$ which satisfies $\|\mathbf{c}\|\cdot\|\mathbf{R}^\star\|_\infty \le N\cdot 2^{-2\rho-\log n-1}$, it implies that each d_i is smaller than $N\cdot 2^{-2\rho-\log n-1}/\hat{p}_i$. Thus, under the above condition, the integer $d = \sum_{i=1}^{n} d_i\cdot\hat{p}_i$, the first entry of output vector $\mathbf{c}$, can be regarded as a dual instance.

Thus, it is enough to show the existence of matrix $\mathbf{R}^\star$ which ensures that the size of $\|\mathbf{c}\|\cdot\|\mathbf{r}^\star\|_\infty$ is less than $N\cdot 2^{-2\rho-\log n-1}$ with $\|\mathbf{c}\| \le \delta^{2(k+1)}\cdot\sqrt{k+1}\cdot N\cdot 2^{-\eta+\rho+1}$ to obtain a dual instance by using the lattice reduction algorithm.

Construction of $\mathbf{R}^\star$

Now, we construct the right inverse matrix $\mathbf{R}^\star$ and estimate the size of $\|\mathbf{r}^\star\|_\infty$ using Babai's nearest plane algorithm [2] and Gaussian Heuristic assumption.

More precisely, let q_1 be a prime integer, which is independent from $\prod_{i=1}^{n} p_i$, and $\mathbf{z}_1 \in \mathbb{Z}^k$ be any vector with $\mathbf{r}\cdot\mathbf{z}_1 \equiv \mathbf{e}_1 \pmod{q_1}$, where $\mathbf{e}_1$ is a n-dimensional standard vector. Consider a full rank lattice $\Lambda_1 = \{\mathbf{x}\in\mathbb{Z}^k : \mathbf{r}\cdot\mathbf{x} \equiv \mathbf{0} \pmod{q_1}\}$, whose determinant is q_1^n and the set of linearly independent vectors $\{\mathbf{x}_i\}_{1\le i\le k} \subset \mathbb{Z}^k$ such that $\|\mathbf{x}_i\| \le \lambda_k(\Lambda_1)$ for each i. We accept Gaussian heuristic to estimate $\lambda_k(\Lambda_1) \approx \sqrt{\frac{k}{2\pi e}}\cdot\det(\Lambda_1)^{1/k} = \sqrt{\frac{k}{2\pi e}}\cdot {q_1}^{n/k}$ so that we can bound $\|\mathbf{x}_i\| \le \sqrt{\frac{k}{2\pi e}}\cdot {q_1}^{n/k}$ for each i.

Using the Babai's nearest plane algorithm on vector $\mathbf{z}$, we obtain the vector $\sum_{i=1}^{k} u_i\cdot\mathbf{x}_i$ so that $\|\mathbf{z} - \sum_{i=1}^{k} u_i\cdot\mathbf{x}_i\| \le \sqrt{\frac{1}{4}\sum_{i=1}^{k}\|\mathbf{x}_i^\star\|^2}$ holds, where each $\mathbf{x}_i^\star$ is Gram-Schmidt vector of $\mathbf{x}_i$. We denote ${\mathbf{z}_1}'$ as $\mathbf{z}_1 - \sum_{i=1}^{k} u_i\cdot\mathbf{x}_i$ and we obtain the following:

$$\|{\mathbf{z}_1}'\| = \|\mathbf{z}_1 - \sum_{i=1}^{k} u_i\cdot\mathbf{x}_i\| \le \sqrt{\frac{1}{4}\sum_{i=1}^{k}\|\mathbf{x}_i^\star\|^2} \le \frac{1}{2}\sqrt{\sum_{i=1}^{k}\|\mathbf{x}_i\|^2} \le \frac{k}{2}\cdot {q_1}^{n/k}.$$

For the modular equation

$$0 \equiv \mathbf{r}\cdot{\mathbf{z}_1}' - \mathbf{e}_1 \equiv \left([\mathbf{r}]_1\cdot{\mathbf{z}_1}' - 1, [\mathbf{r}]_2\cdot{\mathbf{z}_1}', \cdots, [\mathbf{r}]_n\cdot{\mathbf{z}_1}'\right)^T \pmod{q_1},$$

if $|[\mathbf{R}]_i \cdot \mathbf{z}_1{}'| \le \|[\mathbf{R}]_i\| \cdot \|\mathbf{z}_1{}'\| \le \sqrt{k} \cdot 2^{\rho} \cdot \frac{k}{2} \cdot q_1{}^{\frac{n}{k}}$ is less than $\frac{1}{2}q_1$ for all i (*i.e.* $q_1 > (k^{\frac{3}{2}} \cdot 2^{\rho})^{\frac{k}{k-n}}$), the equation $\mathbf{r} \cdot \mathbf{z}_1{}' = \mathbf{e}_1$ holds over the integers.

By setting the size of prime q_1 to be similar with $(k^{\frac{3}{2}} \cdot 2^{\rho})^{\frac{k}{k-n}}$, we can conclude that there exists a vector $\mathbf{z}_1{}'$ which satisfies the equation $\mathbf{r} \cdot \mathbf{z}_1{}' = \mathbf{e}_1$ and the following condition

$$\|\mathbf{z}_1{}'\|_1 \le \sqrt{k} \cdot \|\mathbf{z}_1'\|_2 \le \frac{1}{2} \cdot k^{\frac{3}{2}} \cdot q_1{}^{\frac{n}{k}} \approx \frac{1}{2} \cdot k^{\frac{3k}{2(k-n)}} \cdot 2^{\frac{n}{k-n}\rho}.$$

Similarly, we can also apply it to other $\mathbf{z}_i$'s to construct $\mathbf{r}^\star = (\mathbf{z}_1{}', \cdots, \mathbf{z}_k{}')$ with the vectors $\mathbf{z}_i{}'$ satisfying $\mathbf{r} \cdot \mathbf{z}_i{}' = \mathbf{e}_i$, so we can bound $\|\mathbf{r}^\star\|_\infty$ as follows

$$\|\mathbf{r}^\star\|_\infty = \max_{1 \le i \le k} \|\mathbf{z}_i{}'\|_1 \le \frac{1}{2} \cdot k^{\frac{3k}{2(k-n)}} \cdot 2^{\frac{n}{k-n}\rho}.$$

Hence, we can obtain the upper bound of $\|\mathbf{c}\| \cdot \|\mathbf{R}^\star\|_\infty$ as follows

$$\|\mathbf{c}\| \cdot \|\mathbf{R}^\star\|_\infty \le {\delta'}^{2(k+1)} \cdot \sqrt{k+1} \cdot N \cdot 2^{-\eta+\rho+1} \cdot \frac{1}{2} \cdot k^{\frac{3k}{2(k-n)}} \cdot 2^{\frac{n}{k-n}\rho}.$$

We remind that the size of $\|\mathbf{c}\| \cdot \|\mathbf{R}^\star\|_\infty$ needs to be less than $N \cdot 2^{-2\rho - \log n - 1}$. Therefore the following inequality should be satisfied:

$$\delta^{2(k+1)} \cdot \sqrt{k+1} \cdot N \cdot 2^{-\eta+\rho} \cdot k^{\frac{3k}{2(k-n)}} \cdot 2^{\frac{n}{k-n}\rho} \le N \cdot 2^{-2\rho - \log n - 1}$$

Taking logarithm to both sides of the inequality, we obtain as follows

$$2(k+1)\log\delta \le \eta - 3\rho - \frac{n}{k-n}\rho - \frac{3k}{2(k-n)}\log k - \log(2n\sqrt{k+1}). \tag{7}$$

Since we set $k = \frac{2\gamma}{\eta-\rho} > 2n$, the condition $\frac{k}{k-n} = O(1)$ holds so we can rewrite the above equality and obtain the following condition for n, k, η, and ρ

$$2(k+1)\log\delta \le \eta - 3\rho - \frac{n}{k-n}\rho - O(\log k).$$

The left side of the above inequality $2(k+1)\log\delta$ is approximated as $\frac{4\gamma}{\eta-\rho} \cdot \frac{(\eta-\rho)^2}{8\gamma} = \frac{\eta-\rho}{2}$ so that the equality holds with our optimized parameters $k = \frac{2\gamma}{\eta-\rho}$ and $\log\delta < \frac{(\eta-\rho)^2}{8\gamma}$ for the condition (6). Thus we can conclude that using the lattice reduction $\mathcal{A}_\delta$ with $\log\delta < \frac{(\eta-\rho)^2}{8\gamma}$ and about $\frac{2\gamma}{\eta-\rho}$ CCK-ACD samples to construct the lattice Λ satisfies the conditions (6) and (7). In other words, we can obtain a dual instance from the first entry of output vector in $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ time up to polynomial factors. □

Remark 1. *The time* $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ *up to polynomial factors required for the above algorithm does not depend on the number of secret primes n and bit-length of the multiple of n secret primes* $n \cdot \eta$ *but depends on the bit-length of* CCK-ACD *samples* γ.

By putting together the two theorem, we have the following result,

Theorem 6 (Heuristic)**.** *Let* n, η, ρ *be parameters of the* CCK-ACD *problem and* $k = n + 1 + \sqrt{\frac{\gamma}{2\log\delta}}$ CCK-ACD *samples are given with* $\log\delta < \frac{(\eta-\rho)^2}{8\gamma}$. *When the following equation holds*

$$2(k+1)\log\delta \le \eta - 3\rho - \frac{n}{k-n}\rho,$$

one can recover any secret primes of the CCK-ACD *problem in* $2^{O\left(\frac{\gamma}{(\eta-\rho)^2}\right)}$ *time complexity.*

Now we give our SDA algorithm.

Algorithm 2 SDA algorithm for the CCK-ACD problem

Input: $N = \prod_{i=0}^{n} p_i$
Input: Root Hermite factor δ_0
Input: CCK-ACD samples $b_j = \mathsf{CRT}_{(p_i)}(r_{i,j})$ for $1 \le j \le 2k$ with $k = \lfloor \sqrt{\frac{\gamma}{2 \log \delta_0}} \rceil$.
Output: prime factors p_i's of N

$m \leftarrow 0$
while $m \le 1$ **do**
  Set $\mathbf{b} \leftarrow (b_{1+mk}, b_{2+mk}, \cdots, b_{k+mk})$
  Construct a lattice $\Lambda = \Lambda(\mathbf{b})$ with a basis matrix $\mathbf{B} = \begin{pmatrix} 1 & 0 \\ \mathbf{b}^T & N \cdot \mathbf{I}_k \end{pmatrix}$
  Call the lattice reduction algorithm with a Hermite factor δ on Λ, and let v_0 denote the first entry of the shortest output vector.
  if $2(k+1)\log \delta_0 + \frac{1}{k+1} \log N < \eta - \rho - \log 2\sqrt{2\pi e}$ **then**
    $d^{(m)} \leftarrow v_0$
    $m \leftarrow m + 1$
  end if
end while
Construct matrices $\mathbf{w} = ([d^{(0)} \cdot b_i \cdot b_{n+j}]_N) \in \mathbb{Z}^{n \times n}$ and $\mathbf{w}' = ([d^{(1)} \cdot b_i \cdot b_{n+j}]_N) \in \mathbb{Z}^{n \times n}$.
Calculate $(\mathbf{w}')^{-1}$ over $\mathbb{Q}$ and $\mathbf{Y} = \mathbf{w} \cdot (\mathbf{w}')^{-1} \in \mathbb{Q}^{n \times n}$
Compute eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_n\} \subset \mathbb{Q}$ of $\mathbf{Y}$
Compute pairs of integers $(d_i^{(0)}, d_i^{(1)})$ from $\lambda_i = \frac{d_i^{(0)}}{d_i^{(1)}}$ for $1 \le i \le n$.
$p_i \leftarrow \gcd(N, d^{(0)} \cdot d_i^{(1)} - d^{(1)} \cdot d_i^{(0)})$ for $1 \le i \le n$
return p_i's.

5 Experiments

In this section, we provide the experimental results of OLA, SDA for the CCK-ACD problem. All experiments were carried out on a single Intel Core i5 running at 2.1GHz processor and 16GB memory.

We remark that we use a few simplifications for the experiments to run our algorithm; we run fplll algorithm [11] instead of BKZ algorithm. For the efficiency of the experiment, we choose the number of samples, k, to satisfy the required conditions for attack instead of the asymptotic optimum.

According to our experiments in Table1, from various parameters, we can see that the determinant of the orthogonal lattice is very similar to our prediction. Thus, our assumptions of OLA are reasonable for CCK-ACD and random instances. Particularly in the actual use of parameters, the difference of determinant between CCK-ACD and random is more stark because n and ρ are set much smaller than γ.

Experimental results of OLA refer that our expectation of the condition for OLA is accurate. OLA works well even when the ρ is quite large as long as the condition (2) is satisfied.

We also experimented with a toy parameter in [8]. OLA is slower than conventional attacks, GCD attack in [3], in toy parameters. Since conventional attacks that are the GCD algorithms in [3] are $\tilde{O}(2^{\rho/2})$ polynomial-time operation, they largely depend on the size of ρ unlike OLA. If ρ is larger than current parameters, then OLA can be the faster than other direct algorithms for the CCK-ACD problem.

When the number of secret primes, n, is small, OLA can even find the exact some $\mathbf{r}_i$ through LLL algorithm on $\Lambda^{\perp}(\tilde{\mathbf{U}})$. But if n is more than 100, the outputs of the LLL algorithm are linear combinations of $\mathbf{r}_i$'s with a high probability. For the above reason, we find it difficult to find out an exact $\mathbf{r}_i$ when n is large.

In Table2, we can see SDA experimental results with regards to the CCK-ACD problem. According to our results, we have confirmed that experimental results of SDA are above our expectation, even in parameters that do not satisfy our condition. In SDA, we cannot only distinguish them from a uniform distribution but also find the factor of N and recover the secret primes.

Table 1: Experiments about OLA on the CCK-ACD problem. Random means that we do the OLA with random instances whose size is γ–bits. Parameters* is the toy parameters in [8] with $\lambda = 42$ and our attack cost is 2^{47}. Parameters** is increasing the size of ρ to withstand the GCD attack in [3], although our attack cost is almost the same.

OLA CCK-ACD									
Experimental parameters						Experimental Det		Expected Det	
n	k	η	ρ	$\gamma/10^4$	time(min)	CCK-ACD	Random	CCK-ACD	Random
20	65	1500	500	6	3	10022	38707	10000	38682
30	90	1000	100	8	14	3040	50804	3000	50753
40	120	600	120	5.4	21	5661	34785	5600	34683
50	150	1000	300	10	128	15085	64871	15000	64706
80	150	400	70	4.7	36	5727	21530	5600	21354
80	240	400	100	6.7	615	8162	44239	8000	43840
90	270	200	40	3.8	490	3790	25433	3600	24916
100	240	400	50	8	790	5199	46306	5000	45875
*10	320	988	26	29	14600	284	254770	260	254574
**10	325	988	80	29	15540	824	254813	800	254713

Table 2: Experiments about SDA on the CCK-ACD problem.

SDA CCK-ACD					
n	k	η	ρ	$\gamma/10^4$	time(min)
20	60	1500	500	6	59
30	90	1000	100	8	550
40	120	600	120	5.4	692
50	150	1000	300	10	3650
80	150	400	70	4.7	760
80	240	400	100	6.7	9300
90	270	200	40	3.8	5900
100	300	120	10	2.5	8870
100	250	400	50	8	13950

In the CCK-ACD problem, OLA is much faster than SDA like the ACD problem. The result is not surprising though, considering the size of the determinant of lattice applying lattice reduction algorithm.

6 Conclusion

OLA and SDA are best known attacks for the ACD problem to decide the size of the γ. In this paper, we exteneded those two algorithms to the variants of the ACD problem, CCK-ACD.

Our results show that the extension of the ACD problem using CRT structure has the same security as the ACD problem under the same γ and η. In other words, according to the known reduction [4], it can be said that CCK-ACD is more difficult than PACD, but at least in terms of SDA and OLA, both problems have the same difficulty.

On the other hand, our algorithms were applicable only when there was an exact multiple of the secret primes. Therefore, advancing our current algorithms and enabling to solve the problem without no exact multiple of secret primes would be another interesting point for further exploration.

Acknowledgement: The authors of Seoul National University were supported by Institute for Information & communication Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-6-00598, The mathematical structure of functional encryption and its analysis), and the ARO and DARPA under Contract No.W911NF-15-C-0227. The author of ENS de Lyon was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

References

[1] M. Ajtai. Generating random lattices according to the invariant distribution. draft, (2006).
[2] L. Babai. On lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
[3] Y. Chen and P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 502–519, 2012.
[4] 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. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–335, 2013.
[5] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–12, 2015.
[6] J. H. Cheon and D. Stehlé. Fully homomorphic encryption over the integers revisited. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 513–536, 2015.
[7] J. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. *Advances in Cryptology - EUROCRYPT*, pages 446–464, 2012.
[8] J. S. Coron, T. Lepoint, and M. Tibouchi. Batch fully homomorphic encryption over the integers. *IACR Cryptology ePrint Archive*, page 36, 2013.
[9] J. S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. *Annual Cryptology Conference*, pages 476–493, 2013.
[10] J. S. Coron and H. V. Pereira. On kilian's randomization of multilinear map encodings. *Cryptology ePrint Archive*, page 1129, 2018.
[11] T. F. development team. fplll, a lattice reduction library. Available at https://github.com/fplll/fplll, (2016).
[12] J. Ding and C. Tao. A new algorithm for solving the approximate common divisor problem and cryptanalysis of the fhe based on gacd. *IACR Cryptology ePrint Archive*, page 42, 2014.
[13] S. D. Galbraith, S. W. Gebregiyorgis, and S. Murphy. Algorithms for the approximate common divisor problem. *LMS J. Comput. Math.*, 19(A):58–72, 2016.
[14] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.
[15] G. Hanrot, X. Pujol, and D. Stehlé. Terminating bkz. *IACR Cryptology ePrint Archive*, page 198, 2011.
[16] N. Howgrave-Graham. Approximate integer common divisors. *Cryptography and lattices*, pages 51–66, 2001.
[17] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.
[18] H. H. Nguyen and V. Vu. Random matrices: Law of the determinant. *Ann. Probability*, 42(1):146–167, 2014.
[19] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43, 2010.