creative commons

C O M M O N S D E E D

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

Ph.D. DISSERTATION

Reduction from Module-SIS to Ring-SIS and Sampling Reduction in Multi-Key Homomorphic Encryption by Reusing Error

Module-SIS로부터 Ring-SIS로의 환원과 에러를
재사용한 다중 키 동형암호에서의 샘플링 감소

BY

ZaHyun Koo

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Reduction from Module-SIS to Ring-SIS and Sampling Reduction in Multi-Key Homomorphic Encryption by Reusing Error

Module-SIS로부터 Ring-SIS로의 환원과 에러를 재사용한 다중 키 동형암호에서의 샘플링 감소

BY

ZaHyun Koo

AUGUST 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Reduction from Module-SIS to Ring-SIS and Sampling Reduction in Multi-Key Homomorphic Encryption by Reusing Error

Module-SIS로부터 Ring-SIS로의 환원과 에러를 재사용한 다중 키 동형암호에서의 샘플링 감소

지도교수 노 종 선

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

전기 정보 공학부

구 자 현

구자현의 공학박사 학위 논문을 인준함

2022년 8월

위 원 장: __________________________

부위원장: __________________________

위 원: __________________________

위 원: __________________________

위 원: __________________________

Abstract

In this dissertation, four contributions are given as i) the reduction from module-short integer solution problem (MSIS) to ring-short integer solution problem (RSIS), ii) the improved reduction from MSIS to RSIS, iii) the introduction to the variant of RLWE (Re-RLWE) and the hardness of Re-RLWE, and iv) the variant of the compact multi-key homomorphic encryption (ReCMK-HE) based on Re-RLWE.

First, we propose the reduction from MSIS to RSIS under some condition on RSIS. To demonstrate this reduction, we derive two reductions. We first show that there is a reduction from RSIS$_{q^k,m^k,\beta^k}$ to RSIS$_{q,m,\beta}$. Second, we propose the reduction from MSIS$_{q^k,m^k,\beta_1}$ to RSIS$_{q,m,\beta}$ under some norm constraint of RSIS. Combining these two results implies that RSIS for a specified modulus and the number of samples is more difficult than MSIS under norm constraint of RSIS, which provides the range of possible module rank for MSIS.

Second, we propose the improved reduction from MSIS to RSIS. To prove this reduction, we show that RSIS is more difficult than MSIS with the same modulus and ring dimension under some constraint of RSIS. Also, we show that through the reduction from MSIS to RSIS with the same modulus, the rank of the module is extended as much as the number of instances of RSIS from half of the number of instances of RSIS. Next, we show that MSIS is more difficult than MSIS defined in the previous one. Also, we propose that MSIS with the modulus prime q^k is more difficult than MSIS with the composite modulus c, such that c is divided by q. Through the three reductions, we conclude that RSIS with the modulus q is more difficult than MSIS with the composite modulus c.

Third, we propose the variant of RLWE, denoted by Re-RLWE by reusing the error x as a secret when generating the RLWE sample $(a, b = a \cdot s + x)$. That is, the Re-RLWE sample is generated in the form $(a, b = a \cdot s + x, c = a \cdot x + e)$. To define this problem, we define the Re-RLWE distribution and prove the hardness of Re-RLWE.

Lastly, we propose the variant of the compact multi-key homomorphic encryption ReCMK-HE based on Re-RLWE. This scheme has the modified multiplication keys and the modified rotation keys with the reduced size of key compared to the original CMK-HE.

keywords: Learning with errors (LWE), module-lerarning with errors (MLWE), module-short integer solution problem (MSIS), multi-key homomorphic encryption (MK-HE), ring-learning with errors (RLWE), ring-short integer solution problem (RSIS), short integer solution problem (SIS)

student number: 2018-38217

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Background

Many cryptographic schemes are based on problems that are difficult to solve on computers, including the Rivest-Shamir-Adleman (RSA) based on prime factor decomposition and the elliptic curve cryptographic (ECC) scheme based on the discrete logarithm problem (DLP). Since the prime factor decomposition problem and DLP take a long time to solve on computers, cryptographic schemes based on these problems have been considered secure. However, due to the quantum computer's development, it is known that many cryptographic schemes can be broken using quantum algorithms operated on quantum computers [2]. Therefore, candidates of cryptographic schemes that are resistant to quantum computers have been actively researched. The representative candidates are lattice-based cryptography, code-based cryptography, multivariate polynomial-based cryptography, and isogeny-based cryptography. Among them, the diverse forms of lattice-based cryptography such as public-key cryptographic schemes, signature schemes, and key encapsulation mechanisms are submitted to NIST post-quantum cryptography (PQC) standardization competition for the advantages of small-sized key and efficiency as well as security [3].

Lattice-based cryptographic schemes are based on hard problems such as the *short-*

est independent vector problem (SIVP), which is known to reduce to *short integer solution* (SIS) problem and *learning with errors* (LWE) problem. The SIS problem introduced by Ajtai in 1996 [4] has been used to construct many lattice-based cryptographic schemes. The SIS problem is defined as follows: Let $\mathbb{Z}$ and $\mathbb{R}$ denote the sets of integers and real numbers, respectively. Let $\mathbb{Z}_q$ denote the set of integers modulo q. For any positive integers m, n, given positive real number $\beta \in \mathbb{R}$, and positive integer q, the SIS problem is to find solution $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$ for uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. A one-way function can be constructed from the SIS problem [5], and then many cryptographic schemes can be constructed from one-way function [6], [7], [8].

The LWE problem has two versions, that is, the search LWE and the decision LWE problems. The search LWE problem is defined as follows: For given dimension n and positive integer q and the error distribution χ on $\mathbb{Z}$, the search LWE problem is to find $\mathbf{s}$ for many given independent pairs $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s} \rangle + e)$ for $\mathbf{a} \in \mathbb{Z}_q^n$ chosen uniformly at random and error $e \leftarrow \chi$. The decision LWE problem is to distinguish between many arbitrarily independent pairs $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s} \rangle + e)$ and the same number of samples $(\mathbf{c}, d)$, $\mathbf{c} \in \mathbb{Z}_q^n$ and $d \in \mathbb{Z}_q$ from the uniform distribution over $\mathbb{Z}_q^{n+1}$.

Most public key cryptosystems and homomorphic encryption algorithms on a lattice are constructed based on the LWE [8], [9], [10]. However, cryptographic schemes based on LWE or SIS are inefficient because the size of the key is too large. To overcome this problem, we use the ring-LWE (RLWE) and the ring-SIS (RSIS), which are defined over the ring, that is, the polynomial ring [11]. These problems are also as hard as Id-SIVP, where Id-SIVP is the SIVP problem defined on the ideal lattice with a ring structure.

The module structure is an algebraic structure that generalizes ring structure. Then the module lattice can be seen as a generalized structure of an arbitrary lattice and ideal lattice. Therefore, LWE and SIS, both of which can also be defined on the module lattice, are termed as the module-LWE (MLWE) problem and the module-SIS (MSIS) problem, respectively. Similar to the ideal lattice, both problems are as difficult as the Mod-SIVP [12].

Generally, MSIS is more difficult than RSIS in the polynomial ring. If there is an algorithm $\mathcal{A}$ for solving MSIS, the instance of MSIS becomes the instance of RSIS when the module rank is one. Then the algorithm $\mathcal{A}$ can be used to find the solution of RSIS. This method can similarly be used to the reduction from RLWE to MLWE. Thus, when a lattice-based cryptographic scheme is constructed, MLWE and MSIS having a module structure are preferred as fundamental difficulties of the scheme because of the reduced key-size and security reason.

However, the problem with the module structure is not always more difficult than the problem with the ring structure. In Asiacrypt 2017, Albrecht and Deo showed that there is a reduction from MLWE to R-LWE [13], by handling the error rate and modulus in the M-LWE and R-LWE problems. Specifically, M-LWE with error rate α, modulus q, and the rank of module d reduces to RLWE with error $\alpha \cdot n^2\sqrt{d}$ and modulus q^d. Unlike the LWE problems, the SIS problems do not have an error rate; instead, there is the upper bound β on the norm of the solution of RSIS and we can use the upper bound while retaining the same parameters q^k and m for the reduction from M-SIS to R-SIS.

Also, the cloud computing service that provides on-demand resources for computation through a network is actively used, and AIaaS (AI as a Service), which provides various AI-based functions to customers, is also attracting much attention. However, when an outsourcing server processes customer information, privacy problems arise in processing sensitive personal information. To overcome this problem, the cloud computing service and AIaaS use a cryptographic scheme. In particular, homomorphic encryption (HE), which enables computations on encrypted messages, has been developed over the past few years. HE is developed by Gentry [14], but it is impractical. Many HE schemes have been made practical with various improvements and optimizations [15, 16, 17, 18, 19, 20]. And thus the cloud computing service and AIaaS use the

HE scheme to protect sensitive information [21, 22, 23].

However, HE is not always an appropriate solution when many users are involved in a server. In the conventional single-key HE for multiple users and a single server, the public key generated by one user with the secret key should be shared between users, and each uses encrypted private data using the shared public key. However, there is a possibility that a dishonest user with a secret key corresponding to the shared public key can access other users' data. To solve this problem, the multi-key HE (MK-HE) [24, 25] allows each user to generate its own secret/public key pair, and a server performs homomorphic operations using all users' public keys. Therefore, when many users simultaneously participate in the cloud computing service and AIaaS, MK-HE is more appropriate than HE [26, 27]

Even though there are research papers to implement practical MK-HE schemes [24, 25], we still have two problems to be solved. First, ciphertext expansion occurs as homomorphic operation proceeds. This expansion is proportional to the number of users. Second, MK-HE is possible only when all users' public keys are possessed in the server.

In [28], the ciphertext size in MK-HE is significantly reduced. However, the computation and memory costs are still higher than those of underlying single-key HE. As a partial solution to overcome the ciphertext expansion and the large public key size, there are variants of MK-HE schemes with the pre-defined number of users to create a common public key. And thus, the ciphertext expansion is not depending on the number of users, and the public key size possessed in the server can be reduced [29, 30, 31, 1]. The MK-HE that achieves multi-key security and no ciphertext expansion that depends on the number of users is called the compact MK-HE (CMK-HE).

To perform CMK-HE, each user should generate multiplication keys and rotation keys. However, when the user's computer resources are limited, it may be difficult to generate a large amount of multiplication keys and rotation keys. Also, it can be difficult for the server to hold a huge amount of multiplication keys and rotation keys for each user.

1.2 Overview of Dissertation

This dissertation is organized as follows.

In Chapter 2, basic notations of ideals, modules, canonical embedding, and lattices are presented as preliminaries for understanding the whole of this dissertation. Then, the definitions of lattice problems, RLWE, RSIS, and MSIS are introduced. Also, we present the CMK-HE schemes, which are CMK-CKKS and CMK-BFV.

In Chapter 3, we propose one of main contributions that there is a reduction from MSIS to RSIS under some norm constraint of RSIS, This means that RSIS is more difficult than MSIS. To prove the statement, we derive two reductions, that is, the reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$ and the reduction from $\mathsf{MSIS}_{q^k,m,\beta_1}$ to $\mathsf{RSIS}_{q^k,m,\beta}$ under some condition on the upper bound β on the norm of the solution of $\mathsf{RSIS}_{q^k,m,\beta}$ for any $k > 1$. Due to the condition of RSIS, we also include an analysis of the range of the module rank defining the MSIS. Figure 3.4 summarizes the overview of the contributions for Chapter 3.

In Chapter 4, we propose the improved reduction from MSIS to RSIS. To improve the reduction, we propose a new method to find m distinct solutions for RSIS. Using the new method, we derive the reduction from $\mathsf{MSIS}_{q,m,(t\sqrt{m})^{d-1}\beta^d}$ to $\mathsf{RSIS}_{q,m,\beta}$. Also, we propose the various reduction among the MSIS problems, which lead to the reduction from $\mathsf{MSIS}_{c,m^k,\frac{c}{q^k}(t\sqrt{m})^{k(d-1)}\beta^{kd}}$ to $\mathsf{RSIS}_{q,m,\beta}$ for the modulus c such that q^k divides c for some $k \geq 1$. Figure 4.5 summarizes the overview of the contributions for Chapter 4.

In Chapter 5, we propose the variant of RLWE. This problem reuses the error used in the RLWE. To define the variant of RLWE, we first define the variant of RLWE distribution. Then, we define the variant of RLWE problem, and prove the hardness of this problem.

In Chapter 6, we propose the variant of CMK-HE scheme based on the variant of RLWE. This scheme has the modified multiplication keys and the rotation keys, which are the reduced size of keys compared to the original scheme. Also, we show the correctness, and security and compares them with previous work [1].

Finally, the conclusion is given in Chapter 7.

Chapter 2

PRELIMINARIES

2.1 Notation

2.1.1 Ideal and Module

Let $\Phi(X)$ be a monic irreducible polynomial of degree n and $\mathbb{Q}$ be the set of rational numbers. We will use the $2n$-th cyclotomic polynomial $\Phi(X) = X^n + 1$ with $n = 2^r$ for some positive integer r. Consider the cyclotomic field $K = \mathbb{Q}[X]/\langle\Phi(X)\rangle$ and define R as the ring of integer polynomial modulo $\Phi(X)$, that is, $R = \mathbb{Z}[X]/\langle\Phi(X)\rangle$. Conveniently, we refer to R as the polynomial ring. A non-empty set $I \subseteq R$ is termed as an ideal of R if I is an additive subgroup of R and for all $r \in R$ and all $x \in I$, $r \cdot x \in I$. The quotient R/I is the set of equivalence classes $r + I$ of R modulo I. Let q be a positive integer and let $R_q = R/qR$. In [11], it is shown that R_q is isomorphic to I/qI for a given ideal I of R using the Chinese remainder theorem. A subset $M \subseteq K^d$ is an R-module if M is closed under addition and under scalar multiplication by elements of R. The module M generalizes the ring and the vector space. It is known that M/qM is isomorphic to R_q^d [12]. Hereinafter, vectors are denoted in bold and if $\mathbf{a}$ is a vector, then its i-th coordinate is denoted by a_i. A matrix is denoted by uppercase letter in bold.

2.1.2 Canonical Embedding and Norm

In [12], the canonical embeddings are the n ring homomorphisms $\sigma_j : K \to \mathbb{C}$ for all $j = 1, \ldots, n$, where $\mathbb{C}$ is the set of the complex numbers. They are defined by $\sigma_j(X) = \xi^j$, where ξ is the solution of $X^n + 1$ for any $j \in \mathbb{Z}_{2n}^{\times}$ with $n = 2^r$ for some positive integer r, where $\mathbb{Z}_{2n}^{\times}$ denotes the set of integer j module $2n$ such that $\gcd(j, 2n) = 1$. We define the canonical embedding vector as the ring homomorphism $\sigma_C : K \to \mathbb{C}^n$ as $\sigma_C(x) = (\sigma_j(x))_{j \in \mathbb{Z}_{2n}^{\times}}$ under component-wise addition and multiplication. The trace $\mathrm{Tr} : K \to \mathbb{Q}$ is defined as $\mathrm{Tr}(x) = \sum_{j \in \mathbb{Z}_{2n}^{\times}} \sigma_j(x)$. For any $x, y \in K$, $\mathrm{Tr}(x \cdot y) = \sum_{j \in \mathbb{Z}_{2n}^{\times}} \sigma_j(x) \cdot \sigma_j(y) = \langle \sigma_C(x), \overline{\sigma_C}(y) \rangle$, where $\langle \cdot, \cdot \rangle$ is the Hermitian product on $\mathbb{C}^n$.

For any $a \in K$, we define the norm of a as

$$\|a\| = \|\sigma_C(a)\| = \left(\sum_{j \in \mathbb{Z}_{2n}^{\times}} |\sigma_j(a)|^2 \right)^{1/2}.$$

Also, for any $\mathbf{a} = (a_1, \ldots, a_d) \in K^d$, we define the norm of $\mathbf{a}$ as

$$\|\mathbf{a}\| = \left(\sum_{i=1}^{d} \|a_i\|^2 \right)^{1/2} = \left(\sum_{i=1}^{d} \sum_{j \in \mathbb{Z}_{2n}^{\times}} |\sigma_j(a_i)|^2 \right)^{1/2}.$$

2.1.3 Space H

Let $\mathbb{J}$ denote $[-\frac{n}{2}, \frac{n}{2}] \cap \mathbb{Z}_{2n}^{\times}$. We define the space H as the subspace of $\mathbb{C}^n$ such that

$$H = \{(x_j)_{j \in \mathbb{Z}_{2n}^{\times}} \in \mathbb{C}^n \ : \ \forall j \in \mathbb{J}, x_{2n-j} = \overline{x_j}\}.$$

Let $\mathbf{h}_j = \frac{1}{\sqrt{2}}(\mathbf{e}_j + \mathbf{e}_{2n-j})$ and $\mathbf{h}_{2n-j} = \frac{i}{\sqrt{2}}(\mathbf{e}_j - \mathbf{e}_{2n-j})$ for $j \in \mathbb{J}$, where $\mathbf{e}_j$ denotes the standard basis vector. Then $\mathbf{h}_j$'s are the basis of H. For $x \in K$, we define $\sigma_H(x)$ by $\sigma_H(x) = (x_j)_{j \in \mathbb{J}} \in \mathbb{R}^n$ such that $\sigma_C(x) = \sum_j x_j \cdot \mathbf{h}_j$.

2.1.4 Gaussian Measure

For the center $\mathbf{c} \in \mathbb{R}^n$ and real number $s > 0$, the Gaussian function is defined by $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \| \frac{\mathbf{x}-\mathbf{c}}{s} \|^2)$ for all $\mathbf{x} \in \mathbb{R}^n$. We can obtain the Gaussian probability distribution by using the normalization, that is, $D_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n$. If the center $\mathbf{c}$ is to be zero, we omit the subscript $\mathbf{c}$. A sample from $D_{\mathbf{r}}$ over $\mathbb{R}^n$ is given by $(D_{r_i})_{i=1,\ldots,n}$ for $\mathbf{r} = (r_1, \ldots, r_n)^T \in (\mathbb{R}^+)^n$, where $\mathbb{R}^+$ denotes the set of non-negative real numbers. For $\alpha > 0$, we write $\Psi_{\leq \alpha}$ to denote the set of Gaussian distributions that satisfy $r_i \leq \alpha$ for all i.

2.1.5 Lattices

An n-dimensional lattice is a discrete subgroup of $\mathbb{R}^m$, where $\mathbb{R}$ is the set of real numbers. Specifically, for linearly independent vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$, $\mathbf{b}_i \in \mathbb{R}^m$, for all $i = 1, \ldots, m$, the set

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_m) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \ : \ x_i \in \mathbb{Z} \right\}$$

is a lattice in $\mathbb{R}^m$ with the basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$. Also the dual lattice of $\mathcal{L}^*$ is defined as

$$\mathcal{L}^* = \{\mathbf{x} \in \mathrm{span}(\mathcal{L}) \mid \forall \mathbf{v} \in \mathcal{L}, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}.$$

A lattice is an *ideal lattice* if it is isomorphic to some ideal I of R. Similarly, a lattice is a *module lattice* if it is isomorphic to some R-module M [12].

The $i\text{-}th\ successive\ minimum$ $\lambda_i(\mathcal{L})$ is the smallest radius r such that $\mathcal{L}$ contains i linearly independent vectors of norm at most r.

2.2 Lattice Problems

In this section, we introduce the lattice problems, *learning with errors* (LWE) and *short integer solution* (SIS). First, we consider the *shortest independent vector problem* (SIVP).

Definition 2.1 ([12])**.** SIVP *is defined as follows: Given a lattice $\mathcal{L}$ of dimension n,* SIVP *is to find the n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}$ such that $\max_i \|\mathbf{v}_i\| \leq \gamma \cdot \lambda_n(\mathcal{L})$, where $\gamma \geq 1$ is a function of dimension n.*

This problem is known to be NP-hard for any approximation factor $\gamma \geq O(1)$ [32]. SIVP can be extended to the polynomial ring R if the lattice $\mathcal{L}$ is the ideal lattice, denoted as Id-SIVP. Similarly, if the lattice is the module lattice, we can extend this problem to the module, denoted as Mod-SIVP.

In [33], it is proved that there is a reduction from SIVP to LWE and SIS. This means that LWE and SIS are also NP-hard. And thus, many lattice based cryptographic scheme is based on LWE and SIS. However, lattice cryptographic scheme based on LWE and SIS is inefficient. To overcome the inefficiency, there are the ring variant of LWE and SIS, called RLWE and RSIS. It is shown that RSIS and RLWE are as hard as Id-SIVP defined on the ideal lattice [11]. Also, there are the module variant of LWE and SIS, called MLWE and MSIS. It is shown that MSIS and MLWE are as hard as Mod-SIVP defined on the module lattice [12]. In the next subsections, we introduce RLWE, MLWE, RSIS, and MSIS.

2.2.1 Learning with Errors

First, we define the LWE problem. This problem was introduced by Regev in 2005 [34]. To define the problem, we define the LWE distribution.

Definition 2.2 (LWE distribution)**.** *For given dimension n, positive integer q, $\vec{s} \in \mathbb{Z}_q^n$, and the error distribution ψ on $\mathbb{Z}$, a sample from the* LWE *distribution $A_{\vec{s},\psi}$ over*

$\mathbb{Z}^n \times \mathbb{Z}_q$ *is generated by choosing* $\vec{a} \leftarrow \mathbb{Z}_q^n$ *uniformly at random, choosing* $e \leftarrow \psi$*, and outputting* $(\vec{a}, b = \langle \vec{a}, \vec{s} \rangle + e \mod q) \in \mathbb{Z}_q^{n+1}$.

Definition 2.3 (LWE problem)**.** *The average case decision version of the* LWE *problem, denoted* $\mathsf{LWE}_{q,\psi}$ *is to distinguish with non-negligible advantage between independent samples from* $A_{\vec{s},\psi}$ *and the same number of uniformly at random and independent samples from* $\mathbb{Z}_q^{n+1}$*, where* $\vec{s} \leftarrow \mathbb{Z}_q^n$ *is uniformly at random.*

We can extend LWE to a matrix version.

Definition 2.4 (LWE problem, matrix version)**.** *For given* n, m, q *positive integers, and the error distribution* ψ *on* $\mathbb{Z}$*, the decisional* LWE *problem asks to distinguish between distribution* $(\mathbf{A}, \vec{s}\mathbf{A} + \vec{e})$ *and the uniform distribution over* $\mathbb{Z}_q^{n+1}$*, where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$*, and* $\vec{e} \leftarrow \psi^m$*, and* $\vec{s} \leftarrow \mathbb{Z}_q^n$*.*

The following lemma means that it satisfy that reduces the LWE problem to one in which secret itself it chosen from the error distribution ψ.

Lemma 2.1. *There is a deterministic polynomial transformation* T *that, for* $\vec{s} \leftarrow \mathbb{Z}_q^n$ *and error distribution* ψ*, maps* $A_{\vec{s},\psi}$ *to* $A_{\vec{x},\psi}$*, where* $\vec{x} \leftarrow \psi^n$*.*

Although many lattice-based cryptographic schemes are constructued based on LWE, they are quite inefficient in terms of key size. To overcome this inefficiency, we define the ring learning with errors (RLWE).

To define the RLWE problem, we define the RLWE distribution. For polynomial ring R in cyclotomic field K, its dual is defined as $R^\vee = \{x \in K \ : \ \mathrm{Tr}(xR) \subseteq \mathbb{Z}\}$. Let $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R}$ and $\mathbb{T}_{R^\vee} = K_\mathbb{R}/R^\vee$, where $\otimes$ denotes the tensor product. Let ψ be a distribution on $\mathbb{T}_{R^\vee}$. Let Ψ be a family of distribution over $K_\mathbb{R}$ and D be a distribution over $R_q^\vee$.

Definition 2.5 (RLWE distribution)**.** *For* $s \in R_q^\vee$*, let* $A_{q,s,\psi}^{(R)}$ *denote the distribution on* $R_q \times \mathbb{T}_{R^\vee}$ *obtained by choosing* $a \in R_q$ *from* $U(R_q)$ *and* $e \leftarrow \psi$*, and returning* $(a, \frac{1}{q}(a\cdot$

$s)+e$), where $U(R_q)$ denotes the uniform distribution over R_q. This distribution $A^{(R)}_{q,s,\psi}$ is referred to as the RLWE *distribution.*

Definition 2.6 (RLWE problem, [12], [13])**.** *The decision and search $\mathsf{RLWE}^{(R)}_{m,q,\Psi}(D)$ problems are defined as follows: Let $s \in R_q^\vee$ be uniformly random. $\mathsf{RLWE}^{(R)}_{m,q,\Psi}(D)$ is to distinguish between arbitrarily many independent m samples from $A^{(R)}_{q,s,\psi}$ and the same number of independent samples from the uniform distribution over $R_q \times \mathbb{T}_{R^\vee}$, where ψ is an arbitrary distribution in Ψ and $s \leftarrow D$. The search $\mathsf{RLWE}^{(R)}_{m,q,\Psi}(D)$, denoted by $\mathsf{S\text{-}RLWE}^{(R)}_{m,q,\Psi}$, is to find the secret $s \leftarrow D$ from many samples of $A^{(R)}_{q,s,\psi}(D)$.*

Similarly, we define the LWE problem on module $M = R^d$, which is the generalization of ring and vector space.

Definition 2.7 (MLWE distribution)**.** *For $\mathbf{s} \in (R_q^\vee)^d$, we define $A^{(M)}_{d,q,\mathbf{s},\psi}$ as the distribution on $(R_q)^d \times \mathbb{T}_{R^\vee}$ obtained by choosing a vector $\mathbf{a}$ from distribution $U((R_q)^d)$ and $e \leftarrow \psi$, and returning $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e)$.*

Definition 2.8 (MLWE problem, [12], [13])**.** *The decision and search $\mathsf{MLWE}^{(M)}_{m,q,\Psi}(D)$ problems are defined as follows: Let $\mathbf{s} \in R_q^\vee$ be uniformly random. $\mathsf{MLWE}^{(M)}_{m,q,\Psi}(D)$ is to distinguish between many arbitrarily independent samples from $A^{(M)}_{d,q,\mathbf{s},\psi}$ and the same number of independent samples from the uniform distribution over $(R_q^\vee)^d \times \mathbb{T}_{R^\vee}$, where ψ is an arbitrary distribution in Ψ and $s \leftarrow D$. The search $\mathsf{MLWE}^{(M)}_{m,q,\Psi}(D)$, denoted by $\mathsf{S\text{-}MLWE}^{(M)}_{m,q,\Psi}(D)$, is to find the secret $\mathbf{s} \leftarrow D^d$ of many samples from $A^{(M)}_{d,q,\mathbf{s},\psi}(D)$.*

Generally, the MLWE ($\mathsf{S\text{-}MLWE}$) problem is known to be harder than the RLWE ($\mathsf{S\text{-}RLWE}$). However, under some condition, the RLWE problem is more difficult than the MLWE problem [13] as follows.

Theorem 2.1 ([13], Corollary 3)**.** *Let m be a positive integer and χ be a distribution* *over $R^\vee$ satisfying*

$$\Pr_{s \leftarrow \chi}\left[\|\sigma_H(s)\| > B_1\right] \leq \delta_1 \text{ and}$$
$$\Pr_{s \leftarrow \chi}\left[\max_j \frac{1}{|\sigma_j(s)|} \geq B_2\right] \leq \delta_2$$

for some (B_1, δ_1) and (B_2, δ_2). For $\alpha > 0$ and any $k > 1$ that divides $d > 1$ and

$$r \geq \left(\frac{\max\{\sqrt{n}, B_1B_2\}}{q}\right) \cdot \sqrt{2\ln(2nd(1+m(d+3)))/\pi},$$

there exists a reduction from $\mathsf{S\text{-}MLWE}^{(R^d)}_{m,q,\Psi_{\leq\alpha}}(\chi^d)$ to $\mathsf{S\text{-}MLWE}^{(R^{d/k})}_{m,q^k,\Psi_{\leq\alpha'}}(U(R_q^\vee))$ for $(\alpha')^2 \geq \alpha^2 + 2r^2B_1^2d$.

Corollary 2.1 ([13]). *If we take $k = d$, then there exists an efficient reduction from $\mathsf{S\text{-}MLWE}^{R^d}_{m,q,\Psi_{\leq\alpha}}(\chi^d)$ to $\mathsf{S\text{-}RLWE}^{R}_{m,q,\Psi_{\leq\alpha\cdot n^2\cdot\sqrt{d}}}(U(R_q^\vee))$ with controlled error rate α.*

Definition 2.6 is a very interesting problem, but it is difficult to use the cryptographic scheme. Now, we introduce the discretized version of RLWE. In particular, if the error distribution ψ is supported on R_q, then the secret s can also be chosen from ψ without affecting the hardness of the problem [35].

Definition 2.9 (RLWE distribution). *For a secret $s \in R_q$ and a distribution ψ over R_q, a sample from the* RLWE *distribution $A_{s,\psi}$ over $R_q \times R_q$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $e \leftarrow \psi$, and outputting $(a, b = a \cdot s + e \mod qR)$.*

Definition 2.10 (RLWE problem). *The average-case decision version of the* RLWE *problem, denoted $\mathsf{RLWE}_{q,\psi}$, is to distinguish with non-negligible advantage between independent samples from $A_{q,\psi}$ and the same number of uniformly random and independent samples from $R_q \times R_q$, where $s \leftarrow R_q$ is uniformly random.*

Hereafter, we refer to the discretized version of RLWE as RLWE for convenience.

2.2.2 Short Integer Solution Problem

We define the *short integer solution problem*, which is used in many lattice-based cryptographic schemes such as signature schemes and identification schemes. This problem, which was introduced by Ajtai [4], is defined as follows:

Definition 2.11 ([4], [12]). *The* SIS *problem is defined as follows: Given* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *chosen from the uniform distribution,* SIS *is to find* $\mathbf{z} = (z_1, \ldots, z_m)^T \in \mathbb{Z}^m$ *such that* $\mathbf{A} \cdot \mathbf{z} = \mathbf{0}$ *mod* q *and* $0 < \|\mathbf{z}\| \leq \beta$.

In particular, to guarantee the non-trivial solution $\mathbf{z} \in \mathbb{Z}^m$ for the SIS problem, it is clear that β is less than the modulus q. Indeed, if $\beta \geq q$ and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, then we take the solution $\mathbf{z} = (q, 0, \ldots, 0)^T \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{z} = 0 \bmod q$.

It is proved [33] that there is a reduction from SIVP to the SIS problem. Thus, the SIS problem is also NP-hard. The SIS problem is one of the most important problems pertaining to lattices. Therefore, it is necessary to know the relationship among SIS problems for various parameters. The following theorem shows the hardness of the SIS problem in the integer ring, based on the modulus and the number of samples in a previous work [36].

Theorem 2.2 ([36], Proposition 3.2). *Let* m, n *be integers,* q *be a prime, and* β *be a given real number such that* $q \geq \beta \cdot \omega(\sqrt{n \log n})$. *Then for any positive integer* k, *there is a deterministic reduction from* $\mathsf{SIS}_{q^k, m^k, \beta^k}$ *to* $\mathsf{SIS}_{q,m,\beta}$.

Theorem 2.2 means that the SIS problem with modulus q and m samples is more difficult than the SIS problem with modulus q^k and m^k samples for any positive integer k.

We recall the RSIS and MSIS. RSIS was introduced by Peikert and Rosen and is defined on R. Since the instance of RSIS is polynomial, the key size of the cryptographic scheme based on RSIS can be smaller than that of the cryptographic scheme based on SIS [37], [38].

Definition 2.12 (RSIS problem, [12], [37]). *The problem* $\mathsf{RSIS}_{q,m,\beta}$ *is defined as follows: Given* $a_1, \ldots, a_m \in R_q$ *chosen independently from the uniform distribution, the* RSIS *problem is to find* $z_1, \ldots, z_m \in R$ *such that* $\sum_{i=1}^{m} a_i \cdot z_i = 0$ *mod* q *and* $0 < \|\mathbf{z}\| \leq \beta$, *where* $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$.

The module structure is a generalized structure of ring. Thus, RSIS can be extended to the module lattice, which is termed as the MSIS problem [12].

Definition 2.13 (MSIS problem, [12]). *The problem* $\mathsf{MSIS}_{q,m,\beta}$ *is defined as follows: Given* $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_q^d$ *chosen independently from the uniform distribution,* MSIS *is to find* $z_1, \ldots, z_m \in R$ *such that* $\sum_{i=1}^{m} \mathbf{a}_i \cdot z_i = \mathbf{0}$ *mod* q *and* $0 < \|\mathbf{z}\| \leq \beta$, *where* $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$.

MSIS is known to be more difficult than RSIS. Indeed, suppose that an algorithm $\mathcal{A}$ exists for solving MSIS and let $a_1, \ldots, a_m \in R_q$ be independently uniform instances of RSIS. Also, we choose $a_2^{(j)}, \ldots, a_d^{(j)} \in R_q$ from uniform distribution over R_q for all $j = 1, \ldots, m$, where d is a module rank. Then $\mathbf{a}_j = (a_j, a_2^{(j)}, \ldots, a_d^{(j)})$ and $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are instances of MSIS. Using the algorithm $\mathcal{A}$ for solving MSIS, we obtain a solution $\mathbf{z} = (z_1, \ldots, z_m)^T$ such that

$$\begin{aligned}\sum_{i=1}^{m} \mathbf{a}_i \cdot z_i &= (\sum_{i=1}^{m} a_i \cdot z_i, \sum_{i=1}^{m} a_2^{(i)} \cdot z_i, \ldots, \sum_{i=1}^{m} a_d^{(i)} \cdot z_i) \\ &= \mathbf{0} \mod q\end{aligned}$$

with $\|\mathbf{z}\| \leq \beta$. Since $\sum_{i=1}^{m} a_i \cdot z_i = 0 \mod q$ and $\|\mathbf{z}\| \leq \beta$, we find the solution of the instance of RSIS.

2.3 Multi-Key Homomorphic Encryption

In this section, we fisrt define the multi-key homomorphic encryption (MK-HE). This scheme is a cryptosystem which allows us to evaluate an arithmetic circuit on ciphertexts, possibly encrypted under different keys. To define the MK-HE, we assume that each participating users has an index to its public and secret keys. A multi-key ciphertext implicitly contains an ordered set $T = \{id_0, \dots, id_{k-1}\}$ of associated indices.

Definition 2.14 (Multi-Key Homomorphic Encryption)**.** *Let $\mathcal{M}$ be the message space with arithmetic structure.* $\mathsf{MK\text{-}HE}$ *consists of five probabilistic polynomial time algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$.

- $\textbf{Setup}$*:* $pp \leftarrow \mathsf{MK\text{-}HE.Setup}$. *Take the security parameter as an input and returns the public parameterization. We assume that all other algorithms implicitly take* pp *as an input.*

- $\textbf{Key Generation}$*:* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{MK\text{-}HE.KeyGen}$. *Output a pair of secret and public keys.*

- $\textbf{Encryption}$*:* $\mathsf{ct} \leftarrow \mathsf{MK\text{-}HE.Enc}(\mu; \mathsf{pk})$. *Encrypt a plaintext* $\mu \in \mathcal{M}$ *and outputs a ciphertext* $\mathsf{ct} \in \{0,1\}^*$.

- $\textbf{Decryption}$*:* $\mu \leftarrow \mathsf{MK\text{-}HE.Dec}(\bar{\mathsf{ct}}; \{\mathsf{sk}_{id}\}_{id \in T})$. *Given a ciphertext* $\bar{\mathsf{ct}}$ *with the corresponding sequence of secret keys, outputs a plaintext* μ.

- $\textbf{Homomorphic evaluation}$*:*

  $$\bar{\mathsf{ct}} \leftarrow \mathsf{MK\text{-}HE.Eval}(\mathcal{C}, (\bar{\mathsf{ct}}_1, \dots, \bar{\mathsf{ct}}_\ell), \{\mathsf{pk}_{id}\}_{id \in T}).$$

  Given a circuit $\mathcal{C}$*, a tuple of multi-key ciphertexts* $(\bar{\mathsf{ct}}_1, \dots, \bar{\mathsf{ct}}_\ell)$*, and the corresponding set of public keys* $\{\mathsf{pk}_{id}\}_{id \in T}$*, output a ciphertext* $\bar{\mathsf{ct}}$. *Its index set is the union* $T = T_1 \cup \cdots \cup T_\ell$ *of index sets* T_j *of the input ciphertexts* $\bar{\mathsf{ct}}_j$ *for* $1 \le j \le \ell$.

The following is the security, correctness, and compactness of $\mathsf{MK\text{-}HE}$.

- $\textbf{SemanticSecurity}$: For any two messages $\mu_0, \mu_1 \in \mathcal{M}$, the distribution

  $$\mathsf{MK\text{-}HE.Enc}(\mu_i; \mathsf{pk})$$

for $i = 0, 1$ should be computationally indistinguishable, where $pp \leftarrow \textsf{MK-HE.Setup}(1^\lambda)$ and $(\textsf{sk}, \textsf{pk}) \leftarrow \textsf{MK-HE.KeyGen}(pp)$.

- **Compactness**: MK-HE scheme is compact if the size of a ciphertext relevant to k users is bounded by $\textsf{poly}(\lambda, k)$ for a fixed polynomial $\textsf{poly}(\cdot, \cdot)$.
- **Correctness** : For $1 \leq j \leq \ell$, let $\bar{\textsf{ct}}_j$ be a ciphertext with index set T_j such that $\textsf{MK-HE.Dec}(\bar{\textsf{ct}}_j, \{\textsf{sk}_{id}\}_{id \in T}) = \mu_j$. Let $\mathcal{C} : \mathcal{M}^\ell \to \mathcal{M}$ be a circuit and $\bar{\textsf{ct}} \leftarrow \textsf{MK-HE.Eval}(\mathcal{C}, (\bar{\textsf{ct}}_1, \ldots, \bar{\textsf{ct}}_\ell), \{\textsf{pk}_{id}\}_{id \in T})$ for $T = T_1 \cup \cdots \cup T_\ell$. Then,

$$\textsf{MK-HE.Dec}(\bar{\textsf{ct}}, \{\textsf{sk}_{id}\}_{id \in T}) = \mathcal{C}(\bar{\textsf{ct}}_1, \ldots, \bar{\textsf{ct}}_\ell) \tag{2.1}$$

with an overwhelming probability.

Note that (2.1) can be substituted by approximated equality similar to the CKKS scheme for approximate arithmetic [17].

Now, we introduce the multi-key CKKS (MK-CKKS)and the multi-key BFV (MK-BFV) [24]. The main difference from CKKS and BFV is as follows:

In CKKS and BFV, the homomorphic multiplication of RLWE ciphertexts consists of two steps, tensor product and relinearization. Let $\textsf{sk} = (s, 1)$ for the secret $s \in R$. For input ciphertexts $\textsf{ct}_1$ and $\textsf{ct}_2$, we first compute their tensor product $\textsf{ct} = \textsf{ct}_1 \otimes \textsf{ct}_2$ that satisfies

$$\langle \textsf{ct}, \textsf{sk} \otimes \textsf{sk} \rangle = \langle \textsf{ct}_1, \textsf{sk} \rangle \cdot \langle \textsf{ct}_2, \textsf{sk} \rangle.$$

In $\textsf{sk} \otimes \textsf{sk}$, the nonlinear entry s^2 exists. Thus, it requires to perform the relinearization technique which transforms the extended ciphertext to a canonical ciphertext encrypting the same message. To perform the relinearization, we publish a multiplication key which is some kind of ciphertxt encrypting s^2 under sk.

In MK-CKKS and MK-BFV, a ciphertext related to k different users is of the form $\bar{\textsf{ct}} = (c_0, \ldots, c_k) \in R_q^{k+1}$, which is decryptable by the concatenated secret $\bar{\textsf{sk}} =$

$(s_0, \ldots, s_{k-1}, 1)$. Thus, the decryption is computed by

$$\mu = \langle \bar{\mathsf{ct}}, \bar{\mathsf{sk}} \rangle = \sum_{i=0}^{k-1} c_i \cdot s_i + c_k.$$

Since MK-CKKS and MK-BFV follow the same pipeline for homomorphic operation as in the single-key setting, the tensor product returns an extended ciphertext corresponding $\bar{\mathsf{sf}} \otimes \bar{\mathsf{sk}}$. Hence, we need to generate a relinearization key which consists of multiple ciphertexts encrypting the entries $s_i \cdot s_j$ of $\bar{\mathsf{sf}} \otimes \bar{\mathsf{sk}}$. It requires some additional computations since the term $s_i \cdot s_j$ depends on two secret keys which are independently generated by different users. First, the following operations are commonly used in MK-CKKS and MK-BFV.

- $\mathsf{MK\text{-}HE.Setup}(1^\lambda)$: Given a security parameter λ, set the RLWE dimension n, ciphertext modulus q, key distribution χ, and error distribution ψ over R. Generate a random polynomial $a \leftarrow R_q$. Return the public parameter $pp = (n, q, \chi, \psi, a)$.

- $\mathsf{MK\text{-}HE.UniEnc}(\mu; s)$: For an input plaintext $\mu \in R$, generate a ciphertext $\mathbf{d} = (d_0, d_1, d_2) \in R_q^3$ as follows:

  (i) Sample $r \leftarrow \chi$.

  (ii) Sample $d_1 \leftarrow R_q$ and $e_1 \leftarrow \psi$, and set $d_0 = -s \cdot d_1 + e_1 + r \mod q$.

  (iii) Sample $e_2 \leftarrow \psi$ and set $d_2 = r \cdot a + e_2 + \mu \mod q$.

- $\mathsf{MK\text{-}HE.KeyGen}(pp)$: Each user i samples the secret key $s_i \leftarrow \chi$, an error $e_i \leftarrow \psi$ and sets the public key as $b_i = -s_i \cdot a + e_i \mod q$. Set the multiplication key $\mathbf{d}_i = (d_{i,0}, d_{i,1}, d_{i,2}) \leftarrow \mathsf{MK\text{-}HE.UniEnc}(s_i; s_i)$.

- $\mathsf{MK\text{-}HE.Relin}(\bar{\mathsf{ct}}; \{(b_i, \mathbf{d}_i)\}_{0 \le i \le k-1})$: Given an extended ciphertext $\bar{\mathsf{ct}} = (c_{i,j})_{0 \le i,j \le k} \in R_q^{(k+1)^2}$ and k pairs of multiplication and public keys $\{(b_i, \mathbf{d}_i)\}_{0 \le i \le k-1}$, generate a ciphertext $\bar{\mathsf{ct}}' \in R_q^{k+1}$ as described in Algorithm 2.1

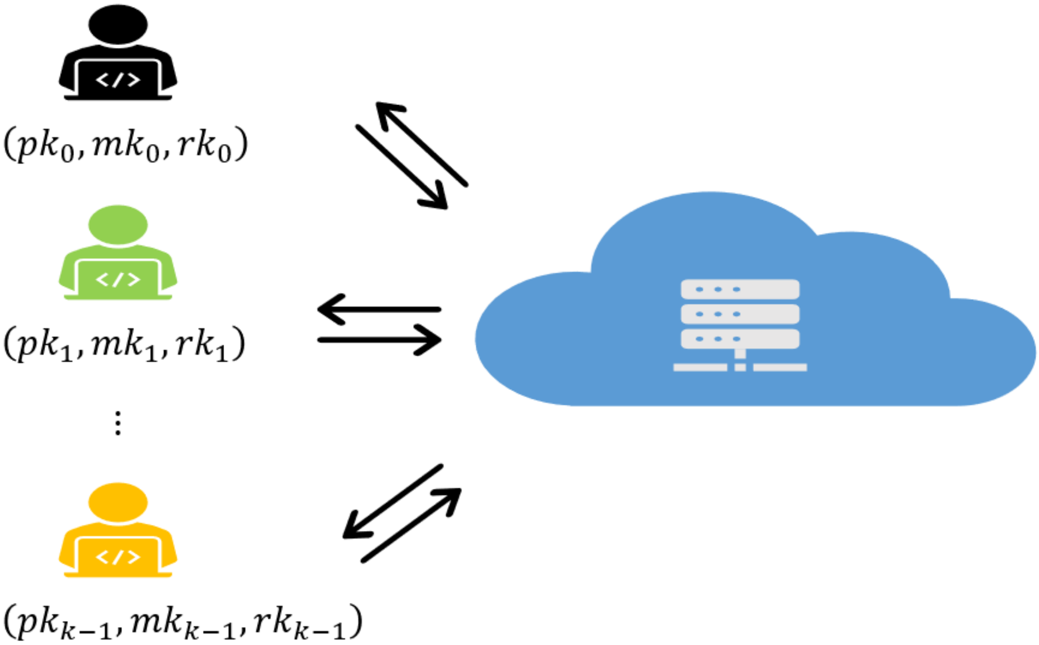

Figure 2.1: Overview of the multi-key homomorphic encryption.

Algorithm 2.1 Relinearization for MK-HE

```
Input : $\bar{\mathsf{ct}} = (c_{i,j})_{0 \leq i,j \leq k}, \{(b_i, \mathbf{d}_i = (d_{i,0}, d_{i,1}, d_{i,2}))\}_{0 \leq i \leq k-1}$
Output : $\bar{\mathsf{ct}}' \in R_q^{k+1}$
$c'_k \leftarrow c_{k,k}$
for $0 \leq i \leq k-1$ do
    $c'_i \leftarrow c_{k,i} + c_{i,k} \mod q$
end for
for $0 \leq i \leq k-1$ do
    $c'_{i,j} \leftarrow c_{i,j} \cdot b_j \mod q$
    $(c'_k, c'_i) \leftarrow (c'_k, c'_i) + c'_{i,j} \cdot (d_{i,0}, d_{i,1}) \mod q$
    $c'_j \leftarrow c'_j + c_{i,j} \cdot d_{i,2} \mod q$
end for
```

2.3.1 Multi-Key CKKS

The CKKS scheme [17] is a leveled HE with support for fixed-point arithmetic. Assume that $q = \prod_{i=0}^{L} q_i$ for some prime q_i, $q_\ell = \prod_{i=0}^{\ell} q_i$, and k is the number of users. This scheme supports the rescaling algorithm to handle the magnitude of encrypted messages. $\mathsf{MK\text{-}CKKS}$ scheme [24] is defined as the following operations together with the algorithm defined in Section 2.3. Figure 2.1 is the overview of the $\mathsf{CMK\text{-}HE}$ scheme:

- $\mathsf{MK\text{-}CKKS.Enc}(m; (a, b))$: Let $m \in R$ be an input plaintext and sample $v \leftarrow \chi$, and $e_0, e_1 \leftarrow \psi$. Return the ciphertext $\mathsf{ct} = (c_0, c_1) \in R_q^2$, where $c_0 = v \cdot a + e_0 \mod q$ and $c_1 = v \cdot b + e_1 + m \mod q$.

- $\mathsf{MK\text{-}CKKS.Dec}(\bar{\mathsf{ct}}; s_0, \ldots, s_{k-1})$: Let $\bar{\mathsf{ct}} = (c_0, \ldots, c_k) \in R_{q_\ell}^{k+1}$ be a ciphertext at level ℓ associated to k users and $s_0, \ldots, s_{k-1}$ be their secret keys. Set $\bar{\mathsf{sk}} = (s_0, \ldots, s_{k-1}, 1)$ and return $\langle \bar{\mathsf{ct}}, \bar{\mathsf{sk}} \rangle \mod q_\ell$.

- $\mathsf{MK\text{-}CKKS.Add}(\bar{\mathsf{ct}}_1, \bar{\mathsf{ct}}_2)$: Given two ciphertexts $\bar{\mathsf{ct}}_i \in R_{q_\ell}^{k+1}$ at level ℓ, return the ciphertext

$$\bar{\mathsf{ct}}' = \bar{\mathsf{ct}}_1 + \bar{\mathsf{ct}}_1 \mod q_\ell.$$

- $\mathsf{MK\text{-}CKKS.Mult}(\bar{\mathsf{ct}}_1, \bar{\mathsf{ct}}_2; \{(b_i, \mathbf{d}_i)\}_{0\le i\le k-1})$: Given two ciphertexts $\bar{\mathsf{ct}}_i \in R_{q_\ell}^{k+1}$ at level ℓ, compute $\bar{\mathsf{ct}} = \bar{\mathsf{ct}}_1 \otimes \bar{\mathsf{ct}}_2 \in R_{q_\ell}^{(k+1)^2}$ and return the ciphertext

$$\bar{\mathsf{ct}}' \leftarrow \mathsf{MK\text{-}HE.Relin}(\bar{\mathsf{ct}}; \{(b_i, \mathbf{d}_i)\}_{0\le i\le k-1}) \in R_{q_\ell}^{k+1}.$$

- $\mathsf{MK\text{-}CKKS.Rescale}(\bar{\mathbf{ct}})$: Given the ciphertext $\bar{\mathsf{ct}} = (c_0, \dots, c_k) \in R_{q_\ell}^{k+1}$ at level ℓ, compute $c_i' = \lfloor q_\ell^{-1} \cdot c_i \rceil$ for $0 \le i \le k$ and return the ciphertext $\bar{\mathsf{ct}}' = (c_0', \dots, c_0') \in R_{q_{\ell-1}}^{k+1}$.

2.3.2 Multi-Key BFV

The BFV scheme is a scale-invariant HE which supports exact computation on a discrete space with a finite characteristic. Let t denote as the plaintext modulus and $\Delta = \lfloor \frac{q}{t} \rceil$ be a scaling factor of the BFV scheme. $\mathsf{MK\text{-}BFV}$ scheme [24] is defined as following operations together with the algorithm defined in Subsection 2.3:

- $\mathsf{MK\text{-}BFV.Enc}(m; (a, b))$: The standard BFV encryption takes a polynomial $m \in R_t$ as the input. Sample $v \leftarrow \chi$, and $e_0, e_1 \leftarrow \psi$. Return the ciphertext $\mathrm{ct} = (c_0, c_1) \in R_q^2$, where $c_0 = v \cdot a + e_0 \mod q$ and $c_1 = v \cdot b + e_1 + \Delta \cdot m \mod q$.

- $\mathsf{MK\text{-}BFV.Dec}(\bar{\mathsf{ct}}; s_0, \dots, s_{k-1})$: Let $\bar{\mathsf{ct}} = (c_0, \dots, c_k) \in R_q^{k+1}$ be a ciphertext associated with k users and $s_0, \dots, s_{k-1}$ be their secret keys. Set $\bar{\mathsf{sk}} = (s_0, \dots, s_{k-1}, 1)$ and return $\lfloor (t/q) \cdot \langle \bar{\mathsf{ct}}, \bar{\mathsf{sk}} \rangle \rceil \mod t$.

- $\mathsf{MK\text{-}BFV.Add}(\bar{\mathsf{ct}}_1, \bar{\mathsf{ct}}_2)$: Given two ciphertexts $\bar{\mathsf{ct}}_i \in R_q^{k+1}$, return the ciphertext

$$\bar{\mathsf{ct}}' = \bar{\mathsf{ct}}_1 + \bar{\mathsf{ct}}_1 \mod q.$$

- $\mathsf{MK\text{-}BFV.Mult}(\bar{\mathsf{ct}}_1, \bar{\mathsf{ct}}_2; \{(b_i, \mathbf{d}_i)\}_{0\le i\le k-1})$: Given two ciphertexts $\bar{\mathsf{ct}}_i \in R_q^{k+1}$, compute $\bar{\mathsf{ct}} = \lfloor (t/q) \cdot (\bar{\mathsf{ct}}_1 \otimes \bar{\mathsf{ct}}_2) \rceil \in R_q^{(k+1)^2}$ and return the ciphertext

$$\bar{\mathsf{ct}}' \leftarrow \mathsf{MK\text{-}HE.Relin}(\bar{\mathsf{ct}}; \{(b_i, \mathbf{d}_i)\}_{0 \le i \le k-1}) \in R_q^{k+1}.$$

2.4 Compact Multi-Key Homomorphic Encryption

In this section, we focus on the CMK-CKKS and CMK-BFV schemes [1], which are variants of MK-CKKS and MK-BFV schemes [24] with the pre-defined number of users. Since CMK-CKKS and CMK-BFV are the variants of those in [24], we assume that CMK-CKKS and CMK-BFV are the common reference string model. The following operations are commonly used in CMK-CKKS and CMK-BFV. Figure 2.2 is the overview of the CMK-HE scheme.

- $\mathsf{CMK\text{-}HE.Setup}(1^\lambda)$: Given a security parameter λ, set the RLWE dimension n, ciphertext modulus q, key distribution χ, and error distribution ψ over R. Generate a random polynomial $a \leftarrow R_q$. Return the public parameter $pp = (n, q, \chi, \psi, a)$.

- $\mathsf{CMK\text{-}HE.KeyGen}(pp)$: Each user i samples the secret key $s_i \leftarrow \chi$ and an error $x_i \leftarrow \psi$ and sets the public key (a, b_i), where $b_i = -s_i \cdot a + x_i \mod q$.

- $\mathsf{CMK\text{-}HE.ComPK}(pk_0, \ldots, pk_{k-1})$: Given all users' public keys $pk_i = (a, b_i = -a \cdot s_i + x_i)$, output a common public key $\hat{pk} = (a, b)$, where

$$\begin{aligned} b &= \sum_{i=0}^{k-1} b_i \mod q \\ &= -a \cdot \sum_{i=0}^{k-1} s_i + \sum_{i=0}^{k-1} x_i \mod q \\ &:= -a \cdot s + x \mod q. \end{aligned}$$

- $\mathsf{CMK\text{-}HE.MultKeyGen}(\hat{pk} = (a, b); s_i)$: For each user i, generate the multipli-

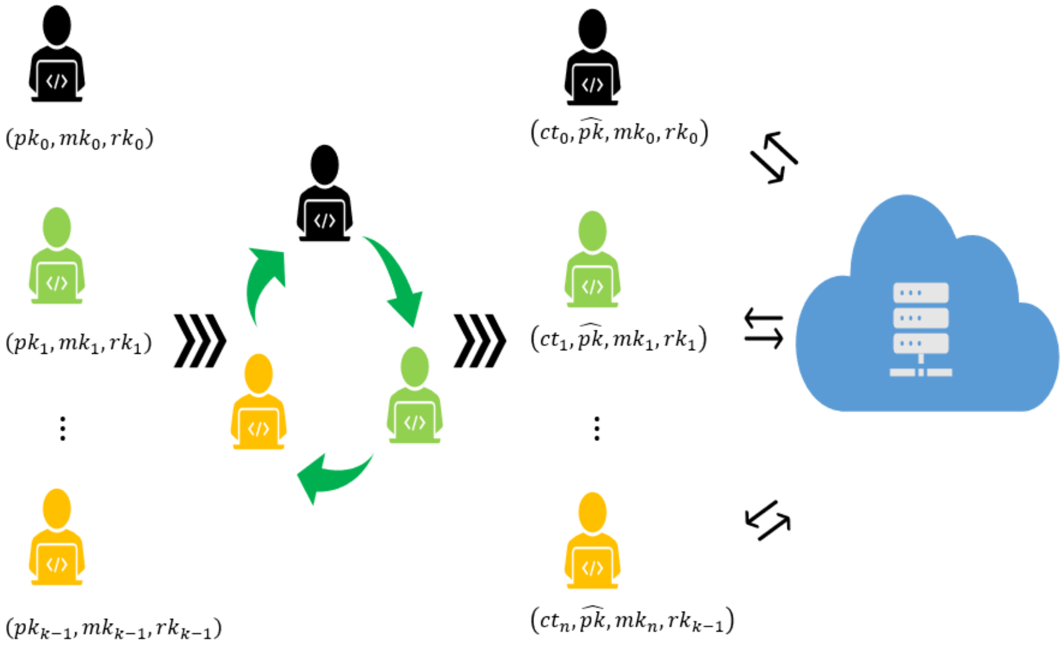
(pk_0, mk_0, rk_0)
(pk_1, mk_1, rk_1)
$(pk_{k-1}, mk_{k-1}, rk_{k-1})$
$(ct_0, \widehat{pk}, mk_0, rk_0)$
$(ct_1, \widehat{pk}, mk_1, rk_1)$
$(ct_n, \widehat{pk}, mk_n, rk_{k-1})$

Figure 2.2: Overview of the compact multi-key homomorphic encryption.

cation key $mk_i = (mk_{i,0}, mk_{i,1})$ as follows:

(i) Sample $r_i \leftarrow \chi$.

(ii) Sample $e_i \leftarrow \psi$ and $mk_{i,0} = a \cdot r_i + s_i + e_i \mod q$.

(iii) Sample $e'_i \leftarrow \psi$ and $mk_{i,1} = b \cdot r_i + e'_i \mod q$.

- $\mathsf{CMK\text{-}HE.ComMultKey}(mk_0, \ldots, mk_{k-1})$: Compute and return a common multiplication key $mk = \sum_{i=0}^{k-1} mk_i$.
- $\mathsf{CMK\text{-}HE.RotKeyGen}(\hat{pk} = (a, b); \tau_t(s_i))$: For each user i and $t \in \mathbb{Z}_{2N}^*$, generate the rotation key $rk_i = (rk_{i,0}, rk_{i,1})$ as follows:

(i) Sample $r_i \leftarrow \chi$.

(ii) Sample $e_i \leftarrow \psi$ and $rk_{i,0} = a \cdot r_i + e_i \mod q$.

(iii) Sample $e'_i \leftarrow \psi$ and $rk_{i,0} = b \cdot r_i + \tau_t(s_i) + e'_i \mod q$.

- $\mathsf{CMK\text{-}HE.ComRotKey}(rk_0, \ldots, rk_{k-1})$: Compute and return a common rotation key $rk = \sum_{i=0}^{k-1} rk_i$, where

$$\begin{aligned} rk_0 &= a \cdot \sum_i^{k-1} r_i + \sum_i^{k-1} e_i \mod q \\ &:= a \cdot r + e \\ rk_1 &= b \cdot \sum_i^{k-1} r_i + \sum_i^{k-1} \tau_t(s_i) + \sum_i^{k-1} e'_i \mod q \\ &= b \cdot r + \tau_t(\sum_i^{k-1} s_i) + e' \mod q \\ &:= b \cdot r + \tau_t(s) + e' \mod q. \end{aligned}$$

2.4.1 Compact Multi-Key CKKS Scheme

As in Subsection 2.3.1, we assume that $q = \prod_{i=0}^{L} q_i$ for some prime q_i, $q_\ell = \prod_{i=0}^{\ell} q_i$, and k is the number of users. CMK-CKKS scheme [1] is defined as the following operations together with the algorithm defined in Subsection 2.4.

- $\mathsf{CMK\text{-}CKKS.Enc}(m; \hat{pk} = (a, b))$: Let $m \in R$ be an input plaintext. Sample $v \leftarrow \chi$ and $e_0, e_1 \leftarrow \psi$ and return $\mathsf{ct} = (c_0, c_1) \in R_q^2$, where $c_0 = v \cdot a + e_0 \mod q$ and $c_1 = v \cdot b + m + e_1 \mod q$.

- $\mathsf{CMK\text{-}CKKS.Add}(\mathsf{ct}_0, \mathsf{ct}_1)$: Given two ciphertexts $\mathsf{ct}_i \in R_{q_\ell}^2$ at level ℓ, return the ciphertext $\mathsf{ct}' = \mathsf{ct}_0 + \mathsf{ct}_1 \mod q_\ell$.

- $\mathsf{CMK\text{-}CKKS.Mult}(\mathsf{ct}_0, \mathsf{ct}_1; mk)$: Given two ciphertexts ct_0 and ct_1 at level q_ℓ, compute $\hat{\mathsf{ct}} = \mathsf{ct}_0 \otimes \mathsf{ct}_1 \in R_{q_\ell}^4$ and return the ciphertext

$$\bar{\mathsf{ct}} \leftarrow \mathsf{CMK\text{-}CKKS.Relin}(\hat{\mathsf{ct}}; mk) \in R_{q_\ell}^2$$

as described in Algorithm 2.2, where $\otimes$ is a tensor product.

Algorithm 2.2 Relinearization for CMK-CKKS

```
Input : $\hat{\mathsf{ct}} = (\hat{c}_0, \hat{c}_1, \hat{c}_2, \hat{c}_3), mk = (mk_0, mk_1)$
Output : $\bar{\mathsf{ct}} \in R_{q_\ell}^2$
$\bar{c}_0 \leftarrow \hat{c}_1 + \hat{c}_2 + \hat{c}_0 \cdot mk_0 \mod q_\ell$
$\bar{c}_1 \leftarrow \hat{c}_3 + \hat{c}_0 \cdot mk_1 \mod q_\ell$
```

- $\mathsf{CMK\text{-}CKKS.Rescale}(\bar{\mathsf{ct}})$: Given a ciphertext $\bar{\mathsf{ct}} = (c_0, c_1) \in R_{q_\ell}^2$, compute $c_i' = \lfloor q_\ell^{-1} \cdot c_i \rceil$ for $i = 0, 1$ and return the ciphertext $\bar{\mathsf{ct}}' = (c_0', c_1') \in R_{q_{\ell-1}}^2$.

For the decryption with multiple secret keys, each party partially decrypts the ciphertext with errors. Then we merge partially decrypted results with c_1 to recover the message.

- $\mathsf{CMK\text{-}CKKS.PartDec}(\mathsf{ct}; s_i)$ For each user i, given a ciphertext $\mathsf{ct} = (c_0, c_1)$, and a secret s_i, sample an error $e_i \leftarrow \psi$ and return $\mu_i = c_0 \cdot s_i + e_i \mod q_0$.

- $\mathsf{CMK\text{-}CKKS.Merge}(\mu_0, \ldots, \mu_{k-1}; \mathsf{ct} = (c_0, c_1))$: Compute and return $\mu = \sum_{i=0}^{k-1} \mu_i + c_1 \mod q_0$.

2.4.2 Compact Multi-Key BFV Scheme

As in Subsection 2.3.2, let t denote as the plaintext modulus and $\varDelta = \lfloor \frac{q}{t} \rceil$ be a scaling factor of the BFV scheme. CMK-BFV scheme [1] is defined as following operations together with the algorithm defined in Subsection 2.4:

- $\mathsf{CMK\text{-}BFV.Enc}(m; \hat{pk} = (a, b))$: Let $m \in R_t$ be an input plaintext. Sample $v \leftarrow \chi$ and $e_0, e_1 \leftarrow \psi$ and return the ciphertext $\mathsf{ct} = (c_0, c_1) \in R_q^2$, where $c_0 = v \cdot a + e_0 \mod q$ and $c_1 = v \cdot b + \varDelta \cdot m + e_1 \mod q$.

- $\mathsf{CMK\text{-}BFV.Add}(\mathsf{ct}_0, \mathsf{ct}_1)$: Given two ciphertexts $\mathsf{ct}_i \in R_q^2$, return the ciphertext $\mathsf{ct}' = \mathsf{ct}_0 + \mathsf{ct}_1 \mod q$.

- $\mathsf{CMK\text{-}BFV.Mult}(\mathsf{ct}_0, \mathsf{ct}_1; mk)$: Given two ciphertexts ct_0 and ct_1, compute $\hat{\mathsf{ct}} = \mathsf{ct}_0 \otimes \mathsf{ct}_1 \in R_q^4$ and return the ciphertext

$$\bar{\mathsf{ct}} \leftarrow \mathsf{CMK\text{-}BFV.Relin}(\hat{\mathsf{ct}}; mk) \in R_q^2$$

as described in Algorithm 2.3, where $\otimes$ is a tensor product.

Algorithm 2.3 Relinearization for CMK-BFV

```
Input : $\hat{\mathsf{ct}} = (\hat{c}_0, \hat{c}_1, \hat{c}_2, \hat{c}_3), mk = (mk_0, mk_1)$
Output : $\bar{\mathsf{ct}} \in R_q^2$
Compute $\hat{c}'_i = \lfloor \frac{1}{\varDelta} \hat{c}_i \rceil \mod q$
$\bar{c}_0 \leftarrow \hat{c}'_1 + \hat{c}'_2 + \hat{c}'_0 \cdot mk_0 \mod q$
$\bar{c}_1 \leftarrow \hat{c}'_3 + \hat{c}'_0 \cdot mk_1 \mod q$
```

- $\mathsf{CMK\text{-}BFV.PartDec}(\mathsf{ct}; s_i)$: For each user i and given a ciphertext $\mathsf{ct} = (c_0, c_1)$ and a secret s_i, sample an error $e_i \leftarrow \psi$ and return $\mu_i = c_0 \cdot s_i + e_i \mod q$.

- $\mathsf{CMK\text{-}BFV.Merge}(\mu_0, \ldots, \mu_{k-1}; \mathsf{ct} = (c_0, c_1))$: Compute $\mu = \sum_{i=0}^{k-1} \mu_i + c_1$ and return $m = \lfloor (t/q) \cdot \mu \rceil$.

Chapter 3

REDUCTION FROM MODULE-SIS TO RING-SIS UNDER STRUCTURED LATTICES

In this chapter, instead of handling the error rate in Corollary 2.1, by controlling the upper bound β on the norm of the solution of SIS, we propose the reduction from MSIS with modulus q^k and m^k samples for any $k > 1$ to RSIS with modulus q and m samples by handling the upper bound β on the norm of the solution of RSIS. To demonstrate this, we first prove that there is a reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$. Second, we show the reduction from MSIS to RSIS under some condition of the upper bound β on the norm of the solution of RSIS.

These two reductions can be combined to obtain the reduction from MSIS to RSIS under the constraints of q, m, β, and k. This means that MSIS can be solved by obtaining the solution of $\mathsf{RSIS}_{q,m,\beta}$. Thus, we have to consider the condition under which $\mathsf{RSIS}_{q,m,\beta}$ can be solved. Since the upper bound β on the norm of the solution of RSIS satisfies that β is less than the modulus q, we consider the polynomial $z \in R$ such that the coefficients of z are in $\{0, 1, \ldots, q-1\}$, where q is a prime. Then, it is clear that $\gcd(z, q) = 1$. Further, for $\mathbf{z} \in R^m$, it is also clear that $\gcd(\mathbf{z}, q) = 1$. Henceforth, we assume that all $\mathsf{RSIS}_{q,m,\beta}$ solutions $\mathbf{z} \in R^m \backslash \{\mathbf{0}\}$ satisfy $\gcd(\mathbf{z}, q) = 1$.

3.1 Reduction from Ring-SIS to Ring-SIS

We propose that solving $\mathsf{RSIS}_{q,m,\beta}$ is more difficult than solving $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ for any integer $k > 1$, which corresponds to the polynomial ring R version of Theorem 2.2. First, we prove that the solution of $\mathsf{RSIS}_{q,m,\beta}$ should be guaranteed and thus we need to extend the following lemma.

Lemma 3.1 ([39], Lemma 5.2). *For any integer q, the instance $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\beta \geq \sqrt{m}q^{n/m}$, the $\mathsf{SIS}_{q,m,\beta}$ admits a solution; i.e., there exists a vector $\mathbf{z} = (z_1, \dots, z_m)^T \in \mathbb{Z}^m \backslash \{\mathbf{0}\}$ such that $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leq \beta$.*

Lemma 3.1 means that to guarantee the solution of $\mathsf{SIS}_{q,m,\beta}$, the upper bound β of the norm of the solution is at least $\sqrt{m}q^{n/m}$. we extend Lemma 3.1 to $\mathsf{RSIS}_{q,m,\beta}$ in the polynomial ring as in the following lemma, the proof of which is similar to that of Lemma 3.1.

Lemma 3.2. *For any integer q, the instances $a_1, \dots, a_m \in R_q$, and $\beta \geq \sqrt{n \cdot m}q^{1/m}$, the $\mathsf{RSIS}_{q,m,\beta}$ admits a solution; that is, there exists a vector $\mathbf{z} = (z_1, \dots, z_m)^T \in R^m \backslash \{\mathbf{0}\}$ such that $\sum_{i=1}^m a_i \cdot z_i = 0 \bmod q$ and $\|\mathbf{z}\| \leq \beta$.*

Proof. Consider all $\mathbf{z} = (z_1, \dots, z_m)^T \in R^m$ such that the coefficients of z_i are in the set $\{0, 1, \dots, \lfloor q^{1/m} \rfloor\}$. Then, there are more than q^n such vectors. Clearly, there exist q^n distinct polynomials in the polynomial ring R_q. Thus, there exist two such vectors $\mathbf{z} \neq \mathbf{z}' \in R^m$ such that $\sum_{i=1}^m a_i \cdot z_i = \sum_{i=1}^m a_i \cdot z_i' \bmod q$. It is clear that $\sum_{i=1}^m a_i \cdot (z_i - z_i') = 0 \bmod q$ and $\|\mathbf{z} - \mathbf{z}'\| \leq \sqrt{n \cdot m}\lfloor q^{1/m} \rfloor \leq \sqrt{n \cdot m}q^{1/m} \leq \beta$ because all coefficients are between $-\lfloor q^{1/m} \rfloor$ and $\lfloor q^{1/m} \rfloor$. □

Now, we propose that for any integer $k > 1$, there is a reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$ as in the following theorem, the proof of which is similar to that of Theorem 2.2.

Theorem 3.1. *Let m be a positive integer and q be a prime. Choose the upper bound of the norm, $\beta \in \mathbb{R}$ such that $\beta \geq \sqrt{n \cdot m} \cdot q^{\frac{1}{m}}$ and $q \geq \beta\sqrt{n}\omega(\log n)$. Assume that* *there exists an algorithm $\mathcal{A}_1$ for solving the $\mathsf{RSIS}_{q,m,\beta}$ problem. Then there exists an algorithm $\mathcal{A}_2$ for solving the $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ for any integer $k > 1$, which corresponds to the reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$.*

Proof. Assume that there exists an algorithm $\mathcal{A}_1$ for solving $\mathsf{RSIS}_{q,m,\beta}$. For the given instances $a_1, a_2, \ldots, a_{m^k} \in R_q$ of $\mathsf{RSIS}_{q^k,m^k,\beta^k}$, which are chosen independently from the uniform distribution $U(R_1)$, we can write $\mathbf{a} = (a_1, \ldots, a_{m^k}) = (\mathbf{a}_1, \ldots, \mathbf{a}_{m^{k-1}})$, where $\mathbf{a}_i$ is the m-tuple vector for $i = 1, \ldots, m^{k-1}$. Using algorithm $\mathcal{A}_1$, we can find a solution $\mathbf{z}_i \in R^m$ with $\|\mathbf{z}_i\| \leq \beta$ such that $\mathbf{a}_i \cdot \mathbf{z}_i = 0 \bmod q$ for all $i = 1, \ldots, m^{k-1}$. Since $\beta < q$ and q is a prime, $\gcd(\mathbf{z}_i, q) = 1$. Thus, $\mathbf{a}_i \cdot \mathbf{z}_i = q \cdot a'_i$ and $a'_i = \mathbf{a}_i \cdot \mathbf{z}_i / q \in R_{q^{k-1}}$ for some $a'_i \in R$. Set $\mathbf{a}' = (a'_1, \ldots, a'_{m^{k-1}})$ and use the induction on k. Then we find a solution $\mathbf{z}' = (z'_1, \ldots, z'_{m^{k-1}})^T \in R^{m^{k-1}}$ with $\|\mathbf{z}'\| \leq \beta^{k-1}$ such that $\mathbf{a}' \cdot \mathbf{z}' = 0 \bmod q^{k-1}$. Let $\mathbf{z} = (z'_1 \cdot \mathbf{z}_1, \ldots, z'_{m^{k-1}} \cdot \mathbf{z}_{m^{k-1}})^T \in R^{m^k}$. Then, we have

$$\begin{aligned} \mathbf{a} \cdot \mathbf{z} &= \sum_{i=1}^{m^{k-1}} z'_i \cdot \mathbf{a}_i \cdot \mathbf{z}_i \\ &= \sum_{i=1}^{m^{k-1}} z'_i \cdot q \cdot a'_i \\ &= q \cdot \sum_{i=1}^{m^{k-1}} z'_i \cdot a'_i \\ &= q \cdot \mathbf{a}' \cdot \mathbf{z}' = 0 \bmod q^k \end{aligned}$$

and $\|\mathbf{z}\| \leq \|\mathbf{z}'\| \cdot \max_i \|\mathbf{z}_i\| \leq \beta^k$. Thus, we prove it. □

In the above proof, the solution of $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ is made by the solutions of $\mathsf{RSIS}_{q,m,\beta}$. Since each solution $\mathbf{z}$ of $\mathsf{RSIS}_{q,m,\beta}$ has $\gcd(\mathbf{z}, q) = 1$, the solution of $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ is relatively prime to q.

3.2 Reduction from Module-SIS to Ring-SIS

Now, we propose that there is a reduction from MSIS to RSIS with the same q^k and m under some condition on the upper bound β on the norm of the solution of RSIS. In general, the MSIS problem is harder than the RSIS problem since the module structure is equal to the ring structure if the rank of the module is one. However, RSIS can be more difficult than MSIS under some condition on the upper bound β on the norm of the solution of RSIS. To show the reduction from MSIS to RSIS, we need to find as many distinct solutions as the number of instances for the same instances of RSIS. However, finding distinct solutions for the same instances of RSIS is difficult because details of the process of the algorithms for solving RSIS are not known. Therefore, certain algorithms may arrive at the same solution for the same instances. To resolve this problem, we use the following lemma, that is, there exist m distinct solutions.

Lemma 3.3. *Let m be a positive integer. Let $k > 1$ be a positive integer and q be a prime. Let β be a real number such that $\max(q, \sqrt{n \cdot m} \cdot q^{\frac{k}{m}}) \leq \beta$. Assume that an algorithm $\mathcal{A}_2$ exists for solving $\mathsf{RSIS}_{q^k,m,\beta}$ such that $\mathcal{A}_2$ outputs a solution $\mathbf{z} \in R^m$ with gcd$(\mathbf{z}, q) = 1$. Let $a_1, \ldots, a_m \in R_{q^k}$ be instances of $\mathsf{RSIS}_{q^k,m,\beta}$. Then we can find m solutions $\bar{\mathbf{z}}^{(j)} = (\bar{z}_1^{(j)}, \ldots, \bar{z}_m^{(j)})^T$ with $\|\bar{\mathbf{z}}^{(j)}\| \leq \beta^2$ such that $\sum_{i=1}^{m} a_i \cdot \bar{z}_i^{(j)} = 0$ mod q^k for all $j = 1, \ldots, m$.*

Proof. Let $\mathbf{a} = (a_1, \ldots, a_m)$ be an instance of $\mathsf{RSIS}_{q^k,m,\beta}$, where $a_i \in R_{q^k}$ for $i = 1, \ldots, m$. Since q is not equal to 0 in R_{q^k}, we can write $\mathbf{a}^{(j)} = (a_1, \ldots, q \cdot a_j, \ldots, a_m)$ for $j = 1, \ldots, m$. Using algorithm $\mathcal{A}_2$, it becomes possible to find the solution $\mathbf{z}^{(j)} = (z_1^{(j)}, \ldots, z_m^{(j)})^T$ with $\|\mathbf{z}^{(j)}\| \leq \beta$ such that

$$a_1 \cdot z_1 + \cdots + q \cdot a_j \cdot z_j + \cdots + a_m \cdot z_m = 0 \bmod q^k$$

for $j = 1, \ldots, m$. Let $\bar{\mathbf{z}}^{(j)} = (z_1, \ldots, q \cdot z_j, \ldots, z_m)^T = (\bar{z}_1^{(j)}, \ldots, \bar{z}_m^{(j)})^T$ for $j =$

$1, \ldots, m$. Then $\bar{\mathbf{z}}^{(j)}$ is a solution of the instance $\mathbf{a}$ with

$$\begin{aligned}
\|\bar{\mathbf{z}}^{(j)}\| &= \|(z_1, \ldots, q \cdot z_j, \ldots, z_m)\| \\
&= (z_1^2 + \cdots q^2 \cdot z_j^2 + \cdots + z_m^2)^{1/2} \\
&\leq q \cdot (z_1^2 + \cdots + z_m^2)^{1/2} \\
&= q \cdot \|\mathbf{z}\| \\
&\leq \beta^2.
\end{aligned}$$

From the property of $\mathcal{A}_2$, each $z_i^{(j)}$ is relatively prime to q. This means that the greatest common divisor of $\bar{z}_i^{(j)}$ and q is 1 if $i \neq j$ and q if $i = j$. Thus, all $\bar{\mathbf{z}}^{(j)}$, $j = 1, \ldots, m$, are distinct solutions for instance $\mathbf{a}$. □

Theorem 3.2. *Let m be a fixed positive integer. Let $k > 1$ be a positive integer and q be a prime. Choose a module rank $d \in \mathbb{Z}_{>0}$ such that*

$$\max(q, \sqrt{n \cdot m} \cdot q^{\frac{k}{m}}) < \sqrt[2d-1]{q^k / (\sqrt{m})^{(d-1)}}. \tag{3.1}$$

Let a positive real number β be an upper bound on the norm of the solution of $\mathsf{RSIS}_{q^k,m,\beta}$ such that

$$\max(q, \sqrt{n \cdot m} \cdot q^{\frac{k}{m}}) \leq \beta < \sqrt[2d-1]{q^k / (\sqrt{m})^{(d-1)}}. \tag{3.2}$$

Assume that an algorithm $\mathcal{A}_2$ exists for solving the $\mathsf{RSIS}_{q^k,m,\beta}$ problem such that $\mathcal{A}_2$ outputs a solution $\mathbf{z} \in R^m$ with gcd$(\mathbf{z}, q) = 1$. Then, an algorithm $\mathcal{A}_3$ exists for solving the $\mathsf{MSIS}_{q^k,m,\beta'}$ problem with module rank d, where $\beta' = m^{\frac{1}{2}(d-1)}\beta^{(2d-1)}$; that is, there exists a reduction from $\mathsf{MSIS}_{q^k,m,\beta'}$ to $\mathsf{RSIS}_{q^k,m,\beta}$ with $\beta' = m^{\frac{1}{2}(d-1)}\beta^{(2d-1)}$.

Proof. Let $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_{q^k}^d$ be instances of $\mathsf{MSIS}_{q^k,m,\beta}$, which are chosen independently from the uniform distribution, where $\mathbf{a}_i = (a_{i1}, \ldots, a_{id})$ and $a_{ij} \in R_{q^k}$. Then we can write the matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1d} & a_{2d} & \cdots & a_{md} \end{bmatrix} = \begin{bmatrix} - & \mathbf{a}'_1 & - \\ - & \mathbf{a}'_2 & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{a}'_d & - \end{bmatrix} \in R_{q^k}^{d\times m}.$$

Then each row $\mathbf{a}'_i$ of $\mathbf{A}$ is considered as an instance of RSIS. Consider the last row $\mathbf{a}'_d$ of $\mathbf{A}$. Then there are m distinct solutions $\bar{\mathbf{z}}_d^{(j)} = (\bar{z}_{d,1}^{(j)}, \ldots, \bar{z}_{d,m}^{(j)})^T$ with $\|\bar{\mathbf{z}}_d^{(j)}\| \leq \beta^2$ such that $\mathbf{a}'_d \cdot \bar{\mathbf{z}}_d^{(j)} = 0 \bmod q^k$ by the Lemma 3.3 for $j = 1, \ldots, m$. Now, we construct the $m \times m$ solution matrix

$$\bar{\mathbf{Z}}_d = \begin{bmatrix} | & | & \cdots & | \\ \bar{\mathbf{z}}_d^{(1)} & \bar{\mathbf{z}}_d^{(2)} & \cdots & \bar{\mathbf{z}}_d^{(m)} \\ | & | & \cdots & | \end{bmatrix}$$

and $\|\bar{\mathbf{Z}}_d\| \leq \beta^2\sqrt{m}$. Then, we have

$$\mathbf{A} \cdot \bar{\mathbf{Z}}_d = \begin{bmatrix} - & \mathbf{a}''_1 & - \\ - & \mathbf{a}''_2 & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{a}''_{d-1} & - \\ - & \mathbf{0} & - \end{bmatrix} \bmod q^k.$$

Applying the above method $d-1$ times, we obtain the solution matrix

$$\mathbf{A}^* = \mathbf{A} \cdot \bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 = \begin{bmatrix} - & \mathbf{a}^*_1 & - \\ - & \mathbf{0} & - \\ \vdots & \vdots & \vdots \\ - & \mathbf{0} & - \end{bmatrix} \bmod q^k.$$

Finally, applying the algorithm $\mathcal{A}_2$ to $\mathbf{a}_1^*$, we find a solution $\mathbf{z}'$ with $\|\mathbf{z}'\| \leq \beta$ such that $\mathbf{A}^* \cdot \mathbf{z}' = \mathbf{0} \bmod q^k$. Then, we have the solution $\mathbf{z} = \bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 \cdot \mathbf{z}'$ for $\mathbf{A}$. Then $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q^k$ and

$$\begin{aligned}\|\mathbf{z}\| &= \|\bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 \cdot \mathbf{z}'\| \\ &\leq \left(\sqrt{m} \cdot \beta^2\right)^{d-1} \cdot \beta \\ &\leq m^{\frac{1}{2}(d-1)} \beta^{(2d-1)}.\end{aligned}$$

By modifying (3.2), we have that the upper bound $\beta' = m^{\frac{1}{2}(d-1)}\beta^{(2d-1)}$ on the norm of the solution of $\mathsf{MSIS}_{q^k,m,\beta'}$ is less than q^k. Thus, we found a non-trivial solution of $\mathsf{MSIS}_{q^k,m,\beta'}$ and showed that there exists a reduction from $\mathsf{MSIS}_{q^k,m,\beta'}$ to $\mathsf{RSIS}_{q^k,m,\beta}$. □

From Theorem 3.2, it is easy to verify that there is a reduction from $\mathsf{MSIS}_{q^k,m^k,\beta'}$ to $\mathsf{RSIS}_{q^k,m^k,\beta^k}$, where $\beta' = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$. To demonstrate the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta'}$ to $\mathsf{RSIS}_{q,m,\beta}$, where $\beta' = m^{\frac{k}{2}(d-1)}\beta^k(2d-1)$, we combine Theorems 3.1 and 3.2 as follows.

Corollary 3.1. *Let m be a fixed positive integer. Let $k > 1$ be a positive integer and q be a prime. Choose a module rank $d \in \mathbb{N}$ such that*

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \sqrt[2d-1]{q^k/(\sqrt{m})^{(d-1)}}. \tag{3.3}$$

Let a positive real number β be an upper bound on the norm of the solution of $\mathsf{RSIS}_{q,m,\beta}$ such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta < \sqrt[2d-1]{q^k/(\sqrt{m})^{(d-1)}}. \tag{3.4}$$

Assume that an algorithm $\mathcal{A}_1$ exists for solving the $\mathsf{RSIS}_{q,m,\beta}$ problem. Then, an algorithm $\mathcal{A}_3$ exists for solving the $\mathsf{MSIS}_{q^k,m^k,\beta'}$ problem with module rank d, where $\beta' =$

$m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$*; that is, there exists a reduction from* $\mathsf{MSIS}_{q^k,m^k,\beta'}$ *to* $\mathsf{RSIS}_{q,m,\beta}$ *with* $\beta' = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$.

Proof. From Theorem 3.1, there exists the algorithm $\mathcal{A}_2$ for solving $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ such that $\mathcal{A}_2$ outputs a solution $\mathbf{z}$ with $\gcd(\mathbf{z}, q) = 1$. Modifying (3.4), we have

$$(\sqrt{n \cdot m} \cdot q^{\frac{1}{m}})^k \leq \beta^k < \left(\sqrt[2d-1]{q/(\sqrt{m})^{(d-1)}} \right)^k.$$

In the inequality on the left, we have

$$\begin{aligned} \beta^k &\geq (\sqrt{n \cdot m} \cdot q^{\frac{1}{m}})^k \\ &\geq \sqrt{n \cdot m^k} \cdot q^{\frac{k}{m}} \\ &\geq \sqrt{n \cdot m^k} \cdot q^{\frac{k}{m^k}}. \end{aligned}$$

In the inequality on the right, we have

$$\beta^k < \left(\sqrt[2d-1]{q/(\sqrt{m})^{(d-1)}} \right)^k = \sqrt[2d-1]{q^k/(\sqrt{m^k})^{(d-1)}}.$$

Thus, we obtain the inequality

$$\sqrt{n \cdot m^k} \cdot q^{\frac{k}{m^k}} \leq \beta^k < \sqrt[2d-1]{q^k/(\sqrt{m^k})^{(d-1)}}.$$

From Theorem 3.2, there exists the algorithm $\mathcal{A}_3$ for solving $\mathsf{MSIS}_{q^k,m^k,\beta'}$ with $\beta' = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$. Thus, there is a reduction from $\mathsf{MSIS}_{q^k,m^k\beta'}$ to $\mathsf{RSIS}_{q,m,\beta}$ with $\beta' = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$. □

3.3 Analysis of Reduction from MSIS to RSIS

In Theorem 3.2, the module rank d is determined by (3.1), in which parameter n is the dimension of the polynomial ring R and thus, n and m are fixed. Thus, d depends on the parameters prime q and k, which is an exponent of q. Modification of (3.1) enables us to find the range of possible module rank d. To obtain the modification of (3.1), we take the logarithm on both sides of (3.1) and multiply $(2d-1)$ to obtain the following equation:

$$\begin{aligned} &(2d-1)\log((mn)^{1/2}q^{k/m}) < (\log q^k - (d-1)\log m^{1/2}) \\ \implies &2d\left(\log((mn)^{1/2}q^{k/m})\right) - \log((mn)^{1/2}q^{k/m}) < \log q^k + \log m^{1/2} - d\log m^{1/2}. \end{aligned}$$

And this inequality summarized as follows for d:

$$d < \frac{\log q^k + \log m^{1/2} + \log((mn)^{1/2}q^{k/m})}{2\log((mn)^{1/2}q^{k/m}) + \log m^{1/2}}.$$

Finally, we obtain the inequality that is the range of possible module rank d as follows:

$$d < \frac{2k(m+1)\log q + 2m\log m + m\log n}{4k\log q + 3m\log m + 2m\log n}. \tag{3.5}$$

Figures 3.1 and 3.2 shows the possible ranks of the module different for parameters and $\log_2(q)$. In the case of Figure 3.1, the logarithm in modulus q of base 2 varies from 0 to 10000 with fixed $n = 2^{16}$ and $k = 2$ and in the case of Figure 3.2, the logarithm in modulus q of base 2 varies from 0 to 10000 with fixed $n = 2^{16}$ and $k = 10$. As m and $\log_2(q)$ increase, the possible module rank d is also increased.

To find the relation between prime q and module rank d, we fix the parameter k. Then we have

$$\frac{2k(m+1)\log q + 2m\log m + m\log n}{4k\log q + 3m\log m + 2m\log n} \to \frac{m+1}{2}$$

as $q \to \infty$, and thus the range of d is

$$d < \frac{m+1}{2} \tag{3.6}$$

for sufficiently large q. Similarly, to find the relation between the exponent k of q and module rank d, we fix the parameter q. Then, we have the same range of d as (3.5) for sufficiently large k.

However, the module rank d is determined by (3.3) in Corollary 3.1. In (3.3), the parameters n and m are fixed. Thus, the module rank d depends only on the parameter q. Modification of (3.3) enables us to find the range of possible module rank d, which is given as

$$d < \frac{2(m+1)\log q + 2m\log m + m\log n}{4\log q + 2m\log m + 2m\log n}. \tag{3.7}$$

The difference between (3.5) and (3.7) is that the latter does not depend on parameter k, which is responsible for the difference in the convergence speed of these two inequalities.

The parameters in Figure3.3 are equal to those in Figures 3.1 and 3.2 except for parameter k. Comparing the figures, it can be seen that the convergence speed of (3.7) is slower than that of (3.5). However, the range of module rank d is the same as that in (3.6) for sufficiently large q.

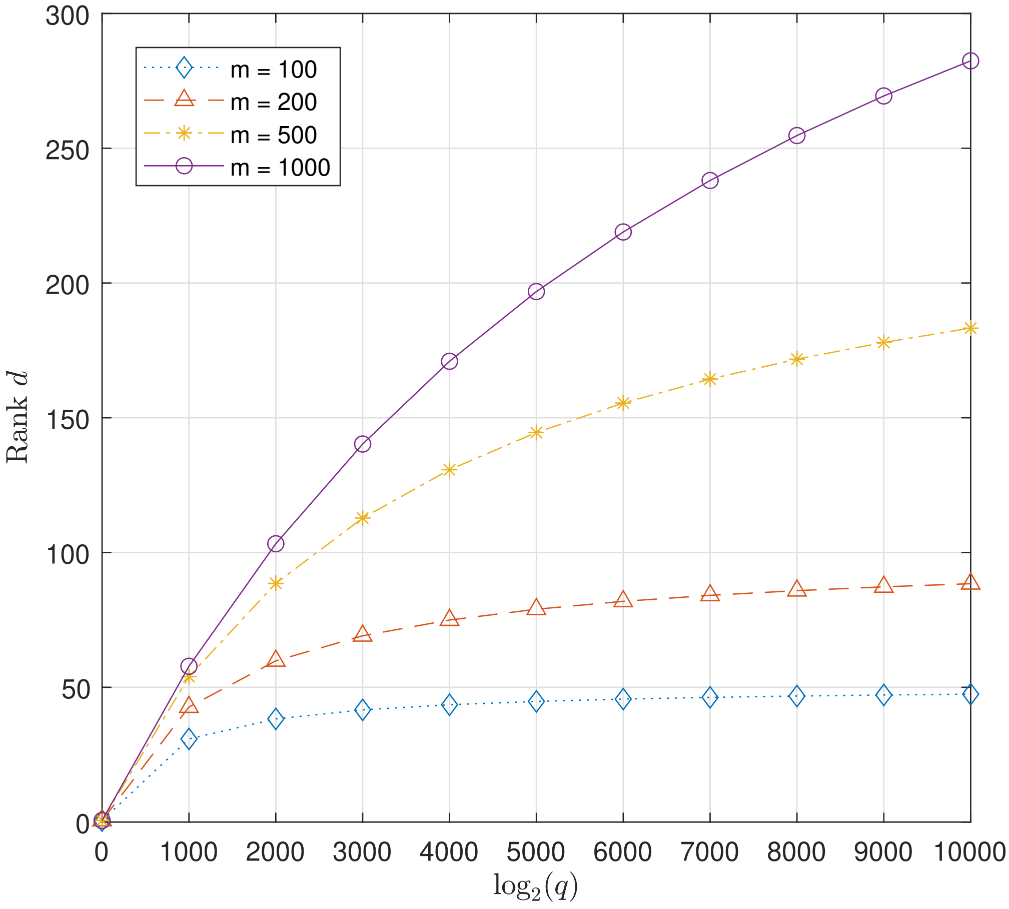

Figure 3.1: Rank of the module when $n = 2^{16}$ and exponent $k = 10$.

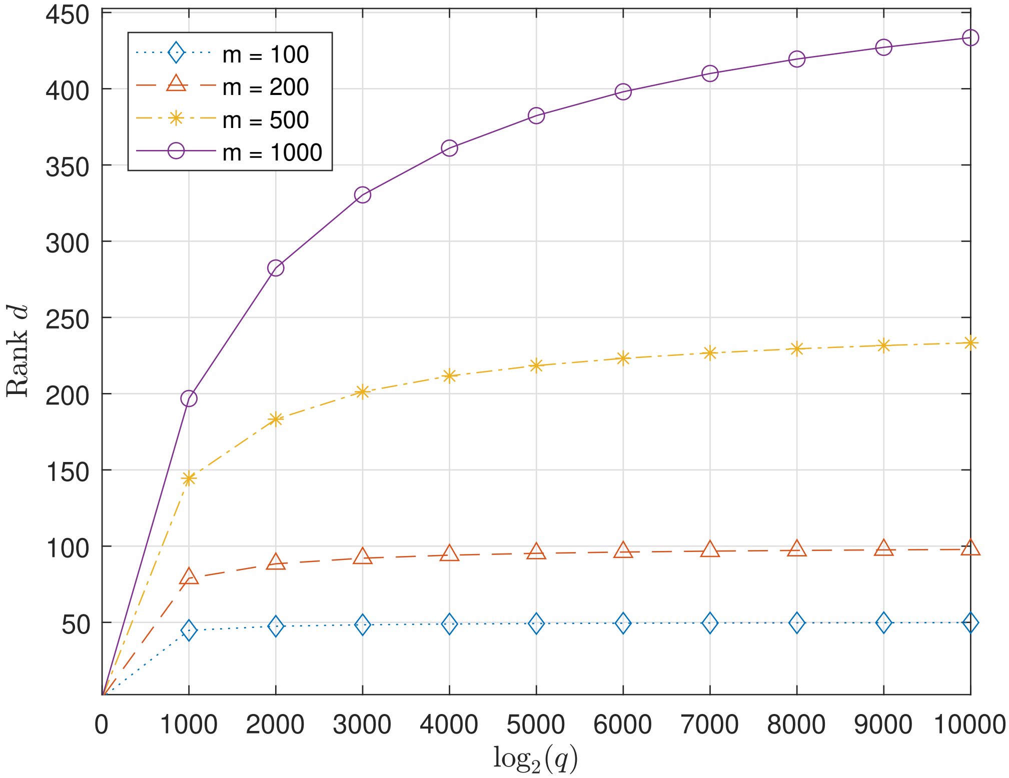

Figure 3.2: Rank of the module when $n = 2^{16}$ and exponent $k = 10$.

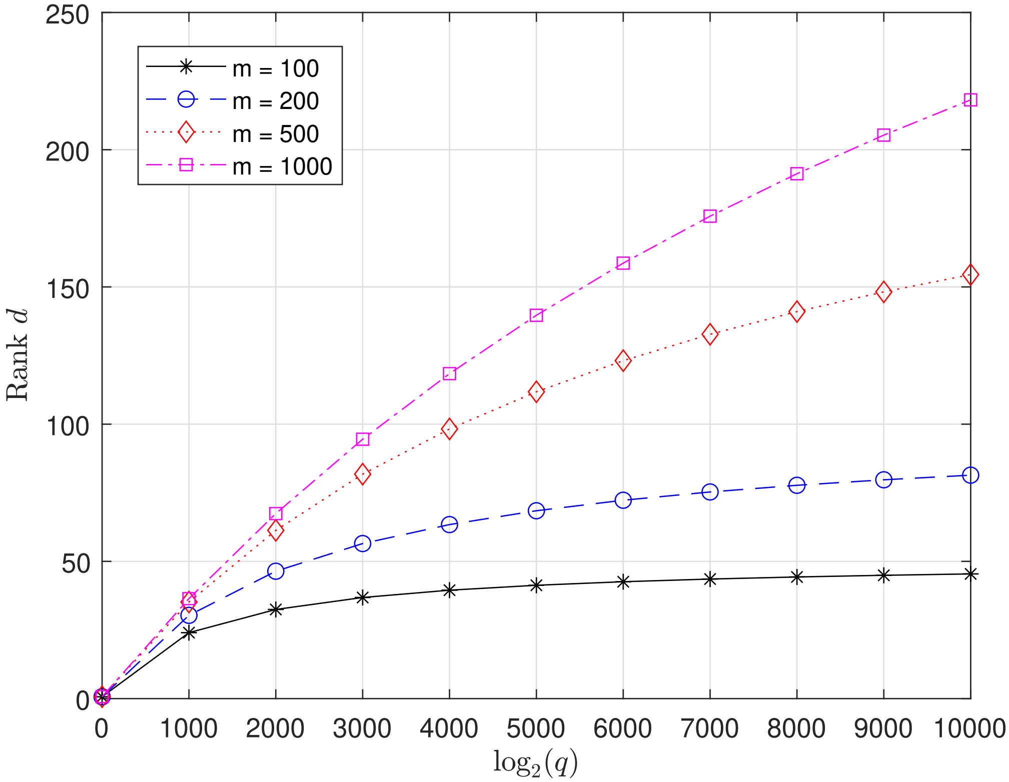

Figure 3.3: Rank of the module when $n = 2^{16}$ for (3.7).

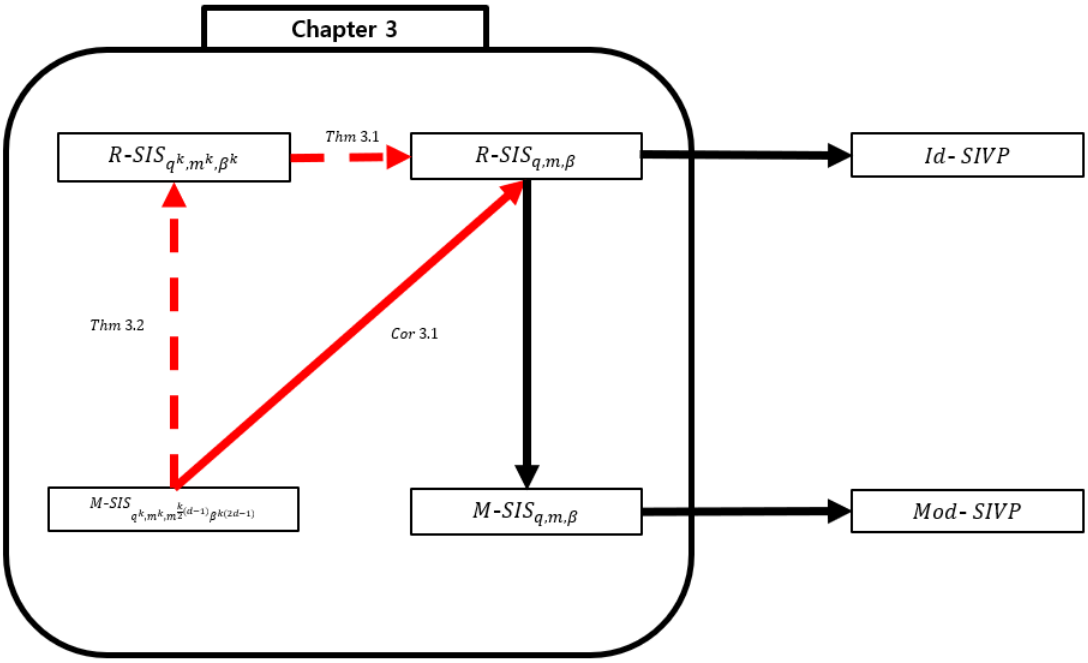
Chapter 3
$R\text{-}SIS_{q^k,m^k,\beta^k}$
Thm 3.1
$R\text{-}SIS_{q,m,\beta}$
$Id\text{-}SIVP$
Thm 3.2
Cor 3.1
$M\text{-}SIS_{q^k,m^k,m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}}$
$M\text{-}SIS_{q,m,\beta}$
$Mod\text{-}SIVP$

Figure 3.4: Overview of the contributions for Chapter 3

Chapter 4

IMPROVED REDUCTION FROM MODULE-SIS TO RING-SIS

In this chapter, we propose a new method to find m distinct solutions for instances of RSIS. In particular, the m distinct solutions are linearly independent over R_q. Using m distinct solutions, we obtain the solution for instances of MSIS. Similar to the previous Chapter 3, there is a range of module rank that allows the reduction from MSIS to RSIS. However, we show that the range of module rank is doubled compared to the previous Chapther 3.

4.1 Improved Reduction from Module-SIS to Ring-SIS

We propose a new method of finding m distinct solutions of instances of RSIS. Finding distinct solutions for the same instances of RSIS is difficult since details of the algorithms' process for solving RSIS are not known. For example, if the algorithm $\mathcal{A}$ for solving RSIS is deterministic, then this algorithm outputs the same solution for the same instance. To overcome this problem, we devise a method to add randomness before using the algorithm for solving RSIS.

Lemma 4.1. *Let m be a positive integer and let t be a positive integer. Choose a prime*

q such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \frac{q}{t}.$$

Choose a real number β such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta < \frac{q}{t}.$$

Suppose that there exists an algorithm $\mathcal{A}$ for solving $\mathsf{RSIS}_{q,m,\beta}$. Let $\mathbf{a} = (a_1, \ldots, a_m) \in R_q^m$ be chosen independently from uniform distribution. Then there exist m linearly independent solutions $\bar{\mathbf{z}}^{(j)} = (\bar{z}_1^{(j)}, \ldots, \bar{z}_m^{(j)}) \in R^m$ such that $\sum_{i=1}^m a_i \cdot \bar{z}^{(j)} = 0 \mod q$ with $\|\bar{\mathbf{z}}^{(j)}\| \leq t \cdot \beta$ for all $j = 1, \ldots, m$.

Proof. **(Step 1)** Let $r^{(1)} = (r_1^{(1)}, \ldots, r_m^{(1)}) \leftarrow U(R^m)$ with $\|r^{(1)}\| \leq t$ and let $\mathbf{a}^{(1)} = (a_1 \cdot r_1^{(1)}, \ldots, a_m \cdot r_m^{(1)})$. Then $\mathbf{a}^{(1)}$ is uniform and we can consider $\mathbf{a}^{(1)}$ as an instance of $\mathsf{RSIS}_{q,m,\beta}$. Using the algorithm $\mathcal{A}$ for solving $\mathsf{RSIS}_{q,m,\beta}$, we obtain a non-trivial solution $\mathbf{z}^{(1)} = (z_1^{(1)}, \ldots, z_m^{(1)})$ such that $\sum_{i=1}^m a_i \cdot r_i^{(1)} \cdot z_i^{(1)} = 0 \mod q$ with $\|\mathbf{z}^{(1)}\| \leq \beta$. Since $\mathbf{a}^{(1)}$ is uniform, there is a non-zero $r_i^{(1)}$(if $r_i^{(1)}$ is all zero in R, then $a_i^{(1)}$ is not uniform). Denote $\bar{\mathbf{z}}^{(1)} = (r_1^{(1)} \cdot z_1^{(1)}, \ldots, r_m^{(1)} \cdot z_m^{(1)})$ in R^m. Then $\bar{\mathbf{z}}^{(1)}$ is a non-trivial solution of $(a_1, \ldots, a_m)$ with $\|\bar{\mathbf{z}}^{(1)}\| \leq t \cdot \beta$ since $\mathbf{z}^{(1)}$ is a non-trivial solution in R^m and there is a non-zero $r_i^{(1)}$ in R. Since $t \cdot \beta$ is less than q, we consider $r_i^{(1)}, z_i^{(1)} \in R$ as $r_i^{(1)}, z_i^{(1)} \in R_q$ for all $i = 1, \ldots, m$.

(Step 2) Let $r^{(2)} = (r_1^{(2)}, \ldots, r_m^{(2)}) \leftarrow U(R^m)$ with $\|r^{(2)}\| \leq t$ and let $\mathbf{a}^{(2)} = (a_1 \cdot r_1^{(2)}, \ldots, a_m \cdot r_m^{(2)})$. Then $\mathbf{a}^{(2)}$ is uniform and we can consider $\mathbf{a}^{(2)}$ as an instance of $\mathsf{RSIS}_{q,m,\beta}$. Through the above process, we obtain a non-trivial solution $\bar{\mathbf{z}}^{(2)} = (r_1^{(2)} \cdot z_1^{(2)}, \ldots, r_m^{(2)} \cdot z_m^{(2)}) \in R^m$ with $\|\bar{\mathbf{z}}^{(2)}\| \leq t \cdot \beta$. Also, we consider $r_i^{(2)}, z_i^{(2)} \in R$ as $r_i^{(2)}, z_i^{(2)} \in R_q$ for all $i = 1, \ldots, m$.

Let $\bar{\mathbf{z}}^{(1)}$ be fixed. Since $\|\bar{\mathbf{z}}^{(1)}\| \leq t \cdot \beta < q$, each coefficient of $\bar{\mathbf{z}}^{(1)}$ is in $\mathbb{Z}_q$. Thus,

$\gcd(\bar{\mathbf{z}}^{(1)}, q) = 1$ because q is a prime. Then we can define

$$S_1 = \mathrm{span}_{R_q}(\bar{\mathbf{z}}^{(1)}) = \{k_1 \cdot \bar{\mathbf{z}}^{(1)} \mid k_1 \in R_q\}$$

and

$$\begin{aligned} T_1 = \{\bar{\mathbf{z}}^{(2)} = (r_1^{(2)} \cdot z_1^{(2)}, \ldots, r_m^{(2)} \cdot z_m^{(2)}) \\ \mid (r_1^{(2)}, \ldots, r_m^{(2)}) \leftarrow U(R^m), \\ (a_1 \cdot r_1^{(2)}, \ldots, a_m \cdot r_m^{(2)}) \to \mathcal{A}, \\ \text{and } \mathbf{z}^{(2)} = (z_1^{(2)}, \ldots, z_m^{(2)}) \leftarrow \mathcal{A}\}. \end{aligned}$$

Since S_1 is determined by an element $k_1 \in R_q$, we obtain $|S_1| = q^n$. However, $\bar{\mathbf{z}}^{(2)}$ is determined by $r_i^{(2)}$ for all $i = 1, \ldots, m$, whether $\bar{\mathbf{z}}^{(2)}$ belongs to S_1 or not. Thus, we obtain $|T_1| = q^{nm}$. Then $|S_1 \cap T_1| \leq |S_1| \ll |T_1|$. If $\bar{\mathbf{z}}^{(2)}$ is in S_1, then we repeat Step 2 until $\bar{\mathbf{z}}^{(1)}$ and $\bar{\mathbf{z}}^{(2)}$ are linearly independent, which is possible from $|S_1| \ll |T_1|$.

Now, assume that $\bar{\mathbf{z}}^{(1)}, \ldots, \bar{\mathbf{z}}^{(j-1)} \in R^m$ are linearly independent solutions of $(a_1, \ldots, a_m)$ such that $\|\bar{\mathbf{z}}^{(k)}\| \leq t \cdot \beta$ for all $k = 1, \ldots, j-1$.

(Step 3) Let $r^{(j)} = (r_1^{(j)}, \ldots, r_m^{(j)}) \leftarrow U(R^m)$ with $\|r^{(j)}\| \leq t$ and let $\mathbf{a}^{(j)} = (a_1 \cdot r_1^{(j)}, \ldots, a_m \cdot r_m^{(j)})$. Through the above process, we obtain a solution $\bar{\mathbf{z}}^{(j)} = (r_1^{(j)} \cdot z_1^{(j)}, \ldots, r_m^{(j)} \cdot z_m^{(j)})$ such that $\|\bar{\mathbf{z}}^{(j)}\| \leq t \cdot \beta$. Also, we consider $r_i^{(j)}, z_i^{(j)} \in R$ as $r_i^{(j)}, z_i^{(j)} \in R_q$ for all $i = 1, \ldots, m$. Let $\bar{\mathbf{z}}^{(1)}, \ldots, \bar{\mathbf{z}}^{(j-1)}$ be fixed and let

$$\begin{aligned} S_{j-1} =& \mathrm{span}_{R_q}(\bar{\mathbf{z}}^{(1)}, \ldots, \bar{\mathbf{z}}^{(j-1)}) \\ =& \{k_1 \cdot \bar{\mathbf{z}}^{(1)} + \cdots + k_{j-1} \cdot \bar{\mathbf{z}}^{(j-1)} \\ & \mid k_i \in R_q \text{ for } i = 1, \ldots, j-1\} \end{aligned}$$

and

$$\begin{aligned}T_{j-1} =\{&\bar{\mathbf{z}}^{(j)} = (r_1^{(j)} \cdot z_1^{(j)}, \ldots, r_m^{(j)} \cdots z_m^{(j)})\\ &\mid (r_1^{(j)}, \ldots, r_m^{(j)}) \leftarrow U(R^m),\\ &(a_1 \cdot r_1^{(j)}, \ldots, a_m \cdot r_m^{(j)}) \to \mathcal{A},\\ &\text{and } \mathbf{z}^{(j)} = (z_1^{(j)}, \ldots, z_m^{(j)}) \leftarrow \mathcal{A}\}.\end{aligned}$$

Then $|S_{j-1}| = q^{n(j-1)}$ since S_{j-1} is determined by elements $k_1, \ldots, k_{j-1} \in R_q$. However, $\bar{\mathbf{z}}^{(j)}$ is determined by $r_i^{(j)}$ for all $i = 1, \ldots, m$ whether $\bar{\mathbf{z}}^{(j)}$ belongs to S_{j-1} or not. Thus, we obtain $|T_{j-1}| = q^{nm}$. Then $|S_{j-1} \cap T_{j-1}| \leq |S_{j-1}| \ll |T_{j-1}|$. If $\bar{\mathbf{z}}^{(j)}$ is in S_{j-1}, then we repeat Step 3 until $\bar{\mathbf{z}}^{(1)}, \bar{\mathbf{z}}^{(2)}, \ldots, \bar{\mathbf{z}}^{(j)}$ are linearly independent, which is also possible from $|S_{j-1}| \ll |T_{j-1}|$. If we repeat this process m times, then we can find m linearly independent solutions $\bar{\mathbf{z}}^{(j)} = (\bar{z}_1^{(j)}, \ldots, \bar{z}_m^{(j)}) = (r_1^{(j)} \cdot z_1^{(j)}, \ldots, r_m^{(j)} \cdot z_m^{(j)})$ such that $\sum_{i=1}^m a_i \cdot r_i^{(j)} \cdot z_i^{(j)} = 0 \mod q$ with $\|\bar{\mathbf{z}}^{(j)}\| \leq t \cdot \beta$ for all $i = 1, \ldots, m$. □

The above solutions are not exact solutions of $\mathsf{RSIS}_{q,m,\beta}$, but we can use these solutions to find the solution of MSIS. Now, we prove the reduction from MSIS to RSIS using Lemma 4.1. The proof of the following theorem is the same as that of Theorem 3.2. However, the upper bound of the solution of RSIS is changed since we use Lemma 4.1. Also, the condition for β is changed as in the following theorem, where the reduction from MSIS to RSIS is satisfied.

Theorem 4.1. *Let m, t be positive integers and q be chosen as in Lemma 4.1. Choose a module rank $d \in \mathbb{Z}_{>0}$ such that*

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}. \tag{4.1}$$

Let a positive real number β be an upper bound on the norm of the solution of $\mathsf{RSIS}_{q,m,\beta}$

such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}.$$

Assume that an algorithm $\mathcal{A}$ exists for solving $\mathsf{RSIS}_{q,m,\beta}$. Then there exists an algorithm $\mathcal{A}_1$ for solving $\mathsf{MSIS}_{q,m,\beta_1}$, where $\beta_1 = (t\sqrt{m})^{d-1}\beta^d$.

Proof. Let $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_q^d$ be instances of $\mathsf{MSIS}_{q,m,\beta}$, which are chosen independently from the uniform distribution, where $\mathbf{a}_i = (a_{i1}, \ldots, a_{id})^T$ and $a_{ij} \in R_q$. Then we can write the matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1d} & a_{2d} & \cdots & a_{md} \end{bmatrix} = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{a}'_2 \\ \vdots \\ \mathbf{a}'_d \end{bmatrix} \in R_q^{d \times m},$$

where $\mathbf{a}'_i = (a_{1i}, \ldots, a_{mi})$. Then the i-th row $\mathbf{a}'_i$ of $\mathbf{A}$ is considered as an instance of RSIS. Consider the last row $\mathbf{a}'_d$ of $\mathbf{A}$. Then there are m distinct solutions $\bar{\mathbf{z}}_d^{(j)} = (\bar{z}_{d,1}^{(j)}, \ldots, \bar{z}_{d,m}^{(j)})^T$ with $\|\bar{\mathbf{z}}_d^{(j)}\| \leq t \cdot \beta$ such that $\mathbf{a}'_d \cdot \bar{\mathbf{z}}_d^{(j)} = 0 \bmod q^k$ for $j = 1, \ldots, m$ from Lemma 4.1. Now, we construct the $m \times m$ solution matrix

$$\bar{\mathbf{Z}}_d = \begin{bmatrix} \bar{\mathbf{z}}_d^{(1)} & \bar{\mathbf{z}}_d^{(2)} & \cdots & \bar{\mathbf{z}}_d^{(m)} \end{bmatrix}$$

and $\|\bar{\mathbf{Z}}_d\| \leq (t \cdot \sqrt{m}) \cdot \beta$. Then, we have

$$\mathbf{A} \cdot \bar{\mathbf{Z}}_d = \begin{bmatrix} \mathbf{a}''_1 \\ \mathbf{a}''_2 \\ \vdots \\ \mathbf{a}''_{d-1} \\ \mathbf{0} \end{bmatrix} \bmod q,$$

where $\mathbf{a}_i''$ is an m-tuple vector. Applying the above method $d-1$ times, we obtain the solution matrix

$$\mathbf{A}^* = \mathbf{A} \cdot \bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 = \begin{bmatrix} \mathbf{a}_1^* \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \bmod q.$$

Finally, applying the algorithm $\mathcal{A}$ to $\mathbf{a}_1^*$, we find a solution $\mathbf{z}'$ with $\|\mathbf{z}'\| \leq \beta$ such that $\mathbf{A}^* \cdot \mathbf{z}' = \mathbf{0} \bmod q$. Then, we have the solution $\mathbf{z} = \bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 \cdot \mathbf{z}'$ for $\mathbf{A}$. Then $\mathbf{A} \cdot \mathbf{z} = \mathbf{0}$ mod q and

$$\begin{aligned} \|\mathbf{z}\| &= \|\bar{\mathbf{Z}}_d \cdots \bar{\mathbf{Z}}_2 \cdot \mathbf{z}'\| \\ &\leq \left(t \cdot \sqrt{m} \cdot \beta\right)^{d-1} \cdot \beta \\ &= \left(t \cdot \sqrt{m}\right)^{d-1} \beta^d. \end{aligned}$$

From (4.1), we have that the upper bound $\beta_1 = (t \cdot \sqrt{m})^{d-1} \cdot \beta^d$ on the norm of the solution of $\mathsf{MSIS}_{q,m,\beta_1}$ is less than q since

$$\begin{aligned} \left(t \cdot \sqrt{m}\right)^{d-1} \beta^d &< \left(t \cdot \sqrt{m}\right)^{d-1} \left(\frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}} \right)^d \\ &= q. \end{aligned}$$

Thus, we find a non-trivial solution of $\mathsf{MSIS}_{q,m,\beta_1}$ and show that there exists a reduction from $\mathsf{MSIS}_{q,m,\beta_1}$ to $\mathsf{RSIS}_{q,m,\beta}$, where $\beta_1 = (t\sqrt{m})^{d-1}\beta^d$. □

4.2 Analysis of Improved Reduction from Module-SIS to Ring-SIS

Similar to Chapter 3, the possible range of module rank of MSIS that satisfies the reduction from $\mathsf{MSIS}_{q,m,\beta_1}$ to $\mathsf{RSIS}_{q,m,\beta}$ depends on (4.1) in Theorem 4.1, where $\beta_1 = (t\sqrt{m})^{d-1}\beta^d$. Moreover, n and m are fixed since n and m are the dimension of the polynomial ring R and the number of instances of RSIS, respectively. Also, given t, the module rank d depends on the modulus q. In this dissertation, the new range of module rank d of MSIS through (4.1) is derived as

$$d < \frac{2m\log q + m\log m + 2m\log t}{m\log n + 2m\log m + 2\log q + 2m\log t}. \tag{4.2}$$

Then, for sufficiently large q, we obtain the range of module rank as

$$d < m.$$

This result is twice as large as the range of module rank of the reduction from MSIS to RSIS [40]. Figures 4.1 and 4.2 shows the possible module ranks with the different parameters and $\log_2 q$ for $n = 2^{16}$, $t = 10$. In the case of Figure 4.1, the bits of modulus q vary from 0 to 100. In the case of Figure 4.2, the bits of modulus q vary from 0 to 10^5. As $\log_2 q$ increases, the possible range of module rank d approaches the number of instances m as in Figure 4.2. Also, as m increases, the possible range of module rank d becomes even wider.

The possible range of module rank is doubled compared to that of the previous result in (3.7). Also, the previous work considered the case that the modulus exponent k is larger than one, but in this work, we propose the reduction for the case of $k = 1$. Figure 4.3 and 4.4 show the comparison of the possible ranges of module ranks of Section 3.3 and the proposed work for $n = 2^{16}$, $t = 10$. In the case of Figure 4.3, the bits of modulus q vary from 0 to 100. The range of module rank of the previous work

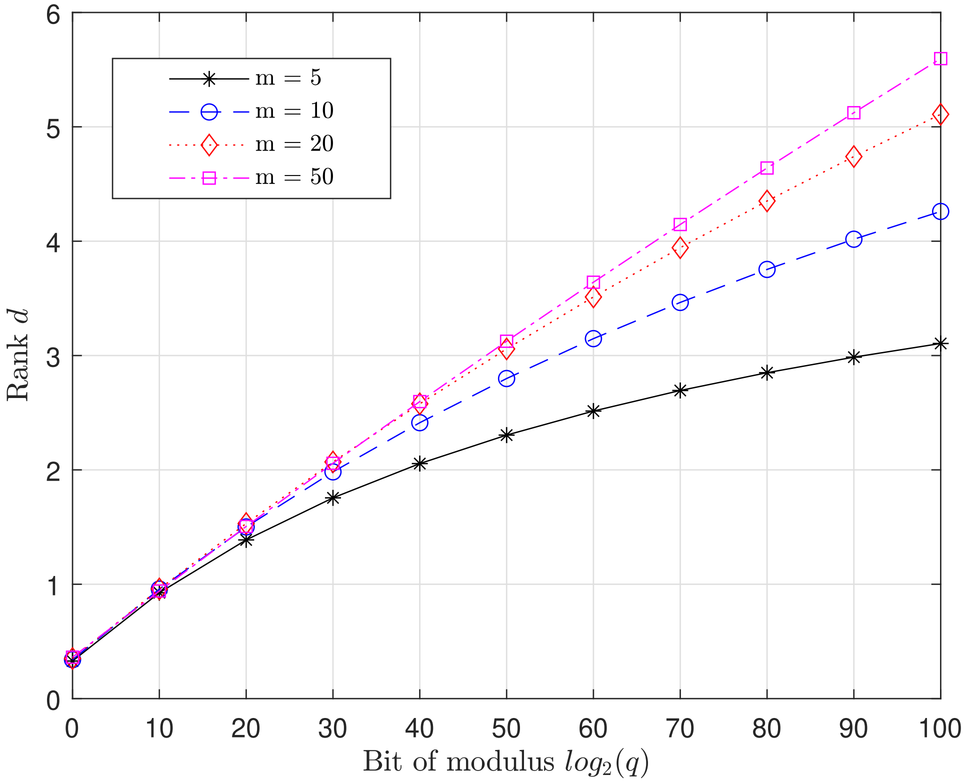

Figure 4.1: Module rank for small number bits of modulus.

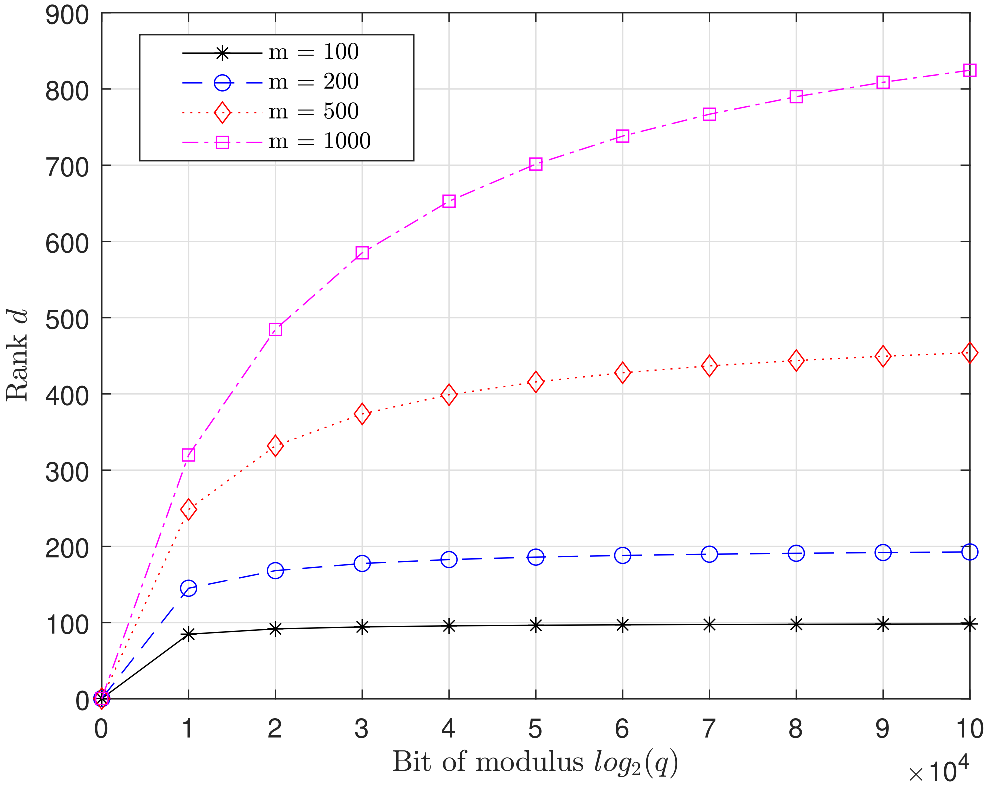

Figure 4.2: Convergence of module rank.

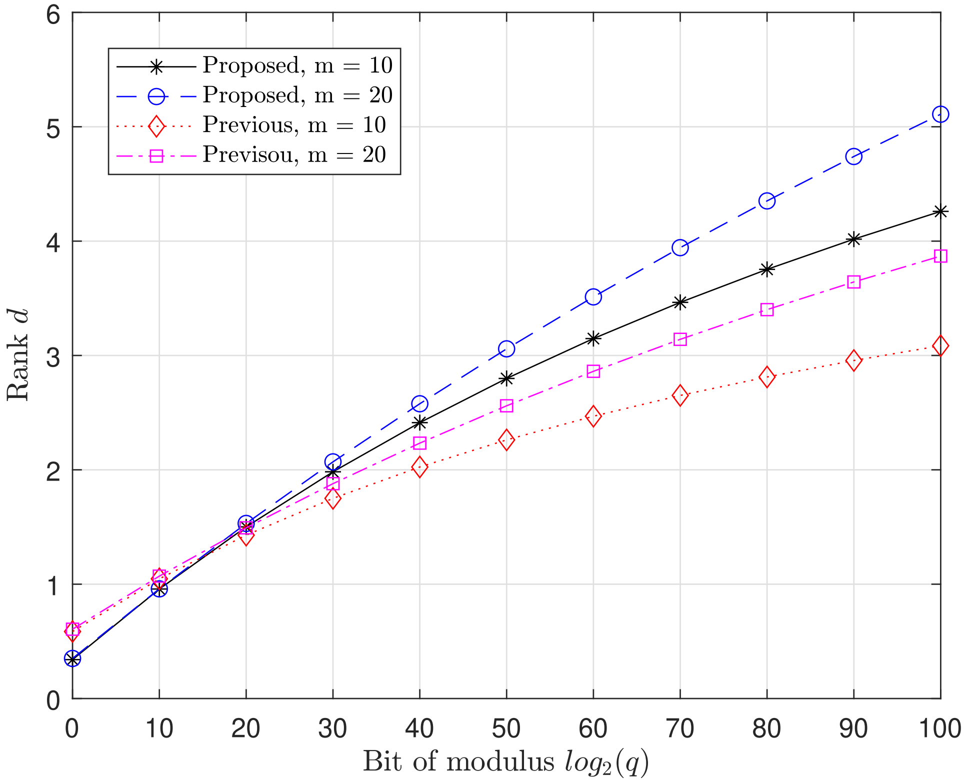

Figure 4.3: For small $\log_2 q$, comparison of the possible ranges of module ranks for Section 3.3 and the proposed works when $n = 2^{16}$.

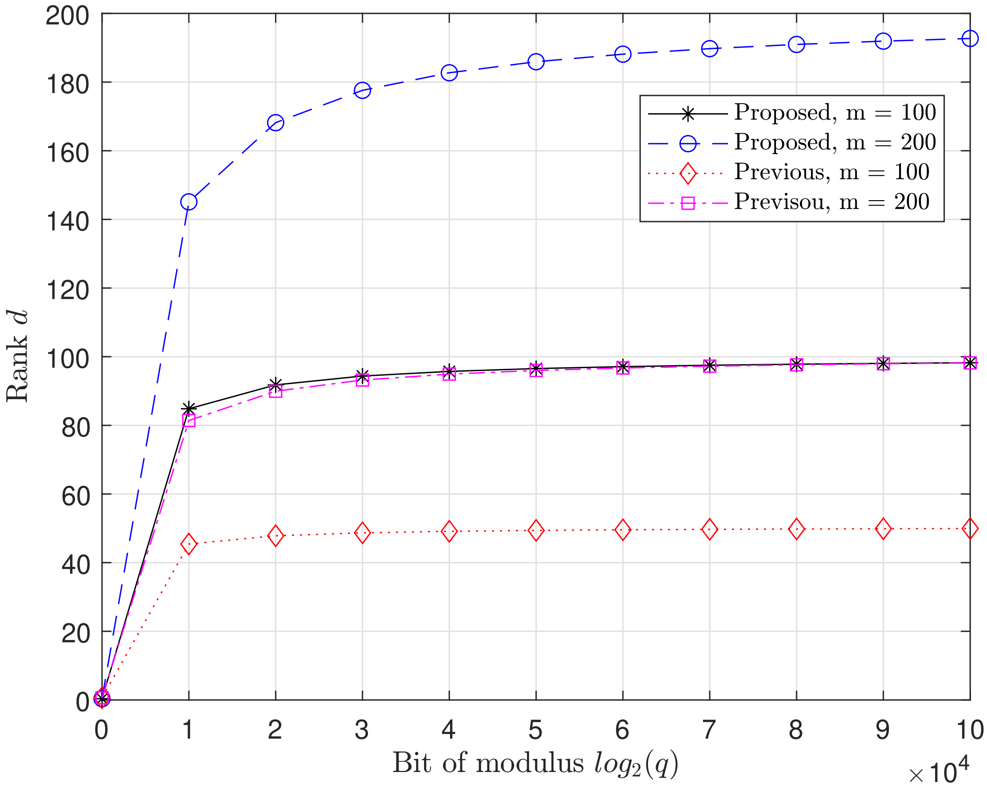

Figure 4.4: For large $\log_2 q$, comparison of the possible ranges of module ranks for Section 3.3 and the proposed works when $n = 2^{16}$.

is larger than that of the proposed work in the range 0 to 10, but, in the range 10 to 100, the range of the proposed work is larger than that of previous work. Also, the previous reduction is possible when the exponent k of the modulus of MSIS is larger than one, but the proposed reduction is also possible when the exponent of k of that of MSIS is equal to one. In the case of Figure 4.4, the bits of modulus q vary from 0 to 10^5, and it shows the convergence values of (3.7) and (4.2). Equation (3.7) converges to half of the number of instances of RSIS, which is the maximum module rank. However, (4.2) converges to the same number of instances of RSIS, which is the maximum module rank.

4.3 Reduction Between Various Module-SIS Problems

In this section, we derive several reductions among the MSIS problems, which lead to the reduction from $\mathsf{MSIS}_{c,m^k,\frac{c}{q^k}(t\cdot\sqrt{m})^{k(d-1)}\beta^{kd}}$ to $\mathsf{RSIS}_{q,m,\beta}$ for the modulus c such that q^k divides c.

4.3.1 Reduction Between Module-SIS Problems with Increased Modulus

First, we derive the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{MSIS}_{q,m,\beta}$ as in the following theorem, where its proof is the same as that of Theorem 3.1.

Theorem 4.2. *Let m be a positive integer and q be a prime. Let d be a positive integer such that d defines a rank of module defining $\mathsf{MSIS}_{q,m,\beta}$ and $\mathsf{MSIS}_{q^k,m^k,\beta^k}$. Assume that there exists an algorithm $\mathcal{A}_1$ for solving the $\mathsf{MSIS}_{q,m,\beta}$ problem. Then there exists an algorithm $\mathcal{A}_2$ for solving the $\mathsf{MSIS}_{q^k,m^k,\beta^k}$ for any integer $k \geq 1$, which corresponds to the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{MSIS}_{q,m,\beta}$.*

Proof. Assume that there exists an algorithm $\mathcal{A}_1$ for solving $\mathsf{MSIS}_{q,m,\beta}$. Assume that $\mathbf{a}_1, \ldots, \mathbf{a}_{m^k} \in R_{q^k}^d$ are chosen independently from uniform distribution over R_q^d. We can write $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_{m^k}) = (\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{m^{k-1}})$, where $\bar{\mathbf{a}}_i$ is an m tuple vector. Using the algorithm $\mathcal{A}_1$, we obtain the solution $\mathbf{z}_i \in R^m$ such that $\bar{\mathbf{a}}_i \cdot \mathbf{z}_i = \mathbf{0} \mod q$

and $\|\mathbf{z}_i\| \leq \beta$. Since $\beta < q$ and q is a prime, $\gcd(\mathbf{z}_i, q) = 1$. Thus, $\bar{\mathbf{a}}_i \cdot \mathbf{z}_i = q \cdot \mathbf{a}'_i$ and $\mathbf{a}'_i = \bar{\mathbf{a}}_i \cdot \mathbf{z}_i / q \in R^d_{q^{k-1}}$ for some $\mathbf{a}'_i \in R^d$. Set $\mathbf{A}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_{m^{k-1}})$ and use the induction on k. Then we find a solution $\mathbf{z}' = (z'_1, \ldots, z'_{m^{k-1}})^T \in R^{m^{k-1}}$ with $\|\mathbf{z}'\| \leq \beta^{k-1}$ such that $\mathbf{A}' \cdot \mathbf{z}' = 0 \mod q^{k-1}$. Let $\mathbf{z} = (z'_1 \cdot \mathbf{z}_1, \ldots, z'_{m^{k-1}} \cdot \mathbf{z}_{m^{k-1}})^T \in R^{m^k}$. Then, we have

$$\begin{aligned}
\mathbf{A} \cdot \mathbf{z} &= \sum_{i=1}^{m^{k-1}} z'_i \cdot \bar{\mathbf{a}}_i \cdot \mathbf{z}_i \\
&= \sum_{i=1}^{m^{k-1}} z'_i \cdot q \cdot \mathbf{a}'_i \\
&= q \cdot \sum_{i=1}^{m^{k-1}} z'_i \cdot \mathbf{a}'_i \\
&= q \cdot \mathbf{A}' \cdot \mathbf{z}' = 0 \mod q^k
\end{aligned}$$

and $\|\mathbf{z}\| \leq \|\mathbf{z}'\| \cdot \max_i \|\mathbf{z}_i\| \leq \beta^k$. Thus, $\mathsf{MSIS}_{q,m,\beta}$ is more difficult than $\mathsf{MSIS}_{q^k,m^k,\beta^k}$. □

Using Theorem 4.1, we can obtain the following reduction.

Corollary 4.1. *There exists the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ to $\mathsf{MSIS}_{q,m,\beta_1}$, where $\beta_1 = (t \cdot \sqrt{m})^{d-1}\beta^d$ and $\beta_2 = \beta_1^k$ as in Figure 4.5.*

4.3.2 Reduction Between Module-SIS Problems with Changed Norm Bound

In order to derive the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_3}$ to $\mathsf{MSIS}_{q,m,\beta_1}$ in Figure 4.5, we use the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_3}$ to $\mathsf{MSIS}_{q^k,m^k,\beta_2}$, where

$$\begin{aligned}\beta_1 &= (t \cdot \sqrt{m})^{d-1}\beta^d,\\ \beta_2 &= \beta_1^k\\ &= (t \cdot \sqrt{m})^{k(d-1)}\beta^{kd},\\ \beta_3 &= m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)},\end{aligned}$$

and $k \geq 1$. To derive the reduction, we need to know the following remark.

Remark 4.1. *Let m and q be positive integers. Let $\beta, \beta' \in \mathbb{R}$ such that*

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta \leq \beta' < q.$$

Assume that there exists an algorithm $\mathcal{A}$ for solving $\mathsf{RSIS}_{q,m,\beta}$. Then there exists an algorithm $\mathcal{A}'$ for solving $\mathsf{RSIS}_{q,m,\beta'}$. Similarly, assume that there exists an algorithm $\mathcal{A}$ for solving $\mathsf{MSIS}_{q,m,\beta}$. Then there exists an algorithm $\mathcal{A}'$ for solving $\mathsf{MSIS}_{q,m,\beta'}$ with the same module rank.

Thus, we derive the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_3}$ to $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ as in the following theorem.

Theorem 4.3. *Let m be a positive integer. Let t be a positive integers and q a prime such that*

$$t \leq \sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \frac{q}{t}.$$

Choose a module rank $d \in \mathbb{Z}_{>0}$ such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}.$$

Let β be a positive real number such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}.$$

Then $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ *is harder than* $\mathsf{MSIS}_{q^k,m^k,\beta_3}$*, where* $\beta_2 = (t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}$*,* $\beta_3 = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$*, and* $k \geq 1$*.*

Proof. Assume that there exists an algorithm $\mathcal{A}_2$ for solving $\mathsf{MSIS}_{q^k,m^k,\beta_2}$, where $\beta_2 = (t \cdot \sqrt{m})^{k(d-1)}\,\beta^{kd}$. Then we need to compare β_2 and β_3 as

$$\begin{aligned}\frac{\beta_3}{\beta_2} &= \frac{m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}}{(t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}} \\ &= \left(\frac{\beta}{t}\right)^{k(d-1)} \\ &\geq \left(\frac{\sqrt{n \cdot m} \cdot q^{\frac{1}{m}}}{t}\right)^{k(d-1)},\end{aligned}$$

which is larger than one if $t \leq \sqrt{n \cdot m} \cdot q^{\frac{1}{m}}$. Thus, we obtain

$$\beta_3 = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)} \geq (t \cdot \sqrt{m})^{k(d-1)}\beta^{kd} = \beta_2.$$

From Remark 4.1, there exists an algorithm $\mathcal{A}_3$ for solving $\mathsf{MSIS}_{q^k,m^k,\beta_3}$, where $\beta_3 = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$. □

From Theorems 4.1, 4.3, and Corollary 4.1, we can derive the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_3}$ to $\mathsf{RSIS}_{q,m,\beta}$, where $\beta_3 = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$ for $k \geq 1$.

4.4 Reduction from Module-SIS with Composite Number as Modulus to Ring-SIS

In this section, we observe the relationship between MSIS with modulus q^k for prime q and $k \geq 1$ and MSIS with modulus c as a composite number. In particular, composite number c is divided by prime q^k. The following theorem shows the relationship between two problems.

Theorem 4.4. *Let m, t, and q be chosen as in Theorem 4.3. Let $k \geq 1$ be a positive integer. Let c be a composite integer such that q^k divides c. Assume that there exists an algorithm $\mathcal{A}$ for solving $\mathsf{MSIS}_{q^k,m^k,\beta_2}$. Then there exists an algorithm $\mathcal{B}$ for solving $\mathsf{MSIS}_{c,m^k,\gamma}$, where $\gamma = \frac{c}{q^k}\beta_2$ and $\beta_2 = (t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}$ for $k \geq 1$.*

Proof. Let $\mathbf{a}_1, \ldots, \mathbf{a}_{m^k} \in R_c^d$ be chosen independently from uniform distribution, where $\mathbf{a}_i = (a_{i1}, \ldots, a_{id})$ for all $i = 1, \ldots, m^k$. For $i = 1, \ldots, m^k$ and $j = 1, \ldots, d$, $a_{ij} = a_{ij}^{(0)} + q^k a_{ij}^{(1)} + \cdots + q^{ks} a_{ij}^{(s)}$ for some integer s and thus we write $\mathbf{a}_i = \mathbf{a}_i^{(0)} + q^k \mathbf{a}_i^{(1)} + \cdots + q^{ks}\mathbf{a}_i^{(s)}$. Thus, $\mathbf{a}_i \equiv \mathbf{a}_i^{(0)} \bmod q^k$. From the algorithm $\mathcal{A}$ for solving $\mathsf{MSIS}_{q^k,m^k,\beta_2}$, we can find the solution $z_1, \ldots, z_{m^k} \in R$ such that

$$\mathbf{a}_1^{(0)} \cdot z_1 + \cdots + \mathbf{a}_{m^k}^{(0)} \cdot z_{m^k} = \sum_{i=1}^{m^k} \mathbf{a}_i^{(0)} \cdot z_i = 0 \bmod q^k$$

and $\|\mathbf{z}\| \leq \beta_2$, where $\mathbf{z} = (z_1, \ldots, z_{m^k})^T$. This means that $\sum_{i=1}^{m^k} \mathbf{a}_i^{(0)} \cdot z_i = q^k \cdot \alpha$ for some $\alpha \in R$. Thus, we have

$$\begin{aligned}\sum_{i=1}^{m^k} \mathbf{a}_i \cdot z_i &= \sum_{i=1}^{m^k} \mathbf{a}_i^{(0)} \cdot z_i \\ &\quad + q^k \sum_{i=1}^{m^k} \mathbf{a}_i^{(1)} \cdot z_i + \cdots + q^{ks} \sum_{i=1}^{m^k} \mathbf{a}_i^{(s)} \cdot z_i \\ &= q^k \cdot \alpha + q^k \sum_{i=1}^{m^k} \mathbf{a}_i^{(1)} \cdot z_i + \cdots + q^{ks} \sum_{i=1}^{m^k} \mathbf{a}_i^{(s)} \cdot z_i \\ &= 0 \bmod q^k.\end{aligned}$$

Thus, $\sum_{i=1}^{m^k} \mathbf{a}_i \cdot z_i = q^k \cdot \alpha'$ for some $\alpha' \in R$ and we have

$$\begin{aligned}\frac{c}{q^k}\sum_{i=1}^{m^k} \mathbf{a}_i \cdot z_i &= \sum_{i=1}^{m^k} \mathbf{a}_i \cdot (\frac{c}{q^k} z_i)\\ &= c \cdot \alpha'\\ &= 0 \bmod c.\end{aligned}$$

Since $\frac{c}{q^k}$ is an integer, $\frac{c}{q^k} z_i$ is in R for all $i = 1, \ldots, m^k$. And we obtain $\|\frac{c}{q^k}\mathbf{z}\| = \frac{c}{q^k}\|\mathbf{z}\| \leq \frac{c}{q^k}\beta_2$. Thus, $\frac{c}{q^k}\mathbf{z}$ is a solution of the instance of $\mathsf{MSIS}_{c,m^k,\gamma}$, where $\gamma = \frac{c}{q^k}\beta_2$ and $\beta_2 = (t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}$ for $k \geq 1$. □

Using Theorems 4.1, 4.5, and Corollary 4.1, we obtain the reduction from $\mathsf{MSIS}_{c,m^k,\gamma}$ to $\mathsf{RSIS}_{q,m,\beta}$, when $\gamma = \frac{c}{q^k}(t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}$ as in the following theorem.

Theorem 4.5. *Let m, t, and q be chosen as in Theorem 4.4. Let c be a composite integer such that c is divided by q^k for some $k \geq 1$. Choose a module rank $d \in \mathbb{Z}_{>0}$ such that*

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}.$$

Let a positive real number β be an upper bound on the norm of the solution of $\mathsf{RSIS}_{q,m,\beta}$ such that

$$\sqrt{n \cdot m} \cdot q^{\frac{1}{m}} \leq \beta < \frac{\sqrt[d]{q \cdot t \cdot \sqrt{m}}}{t \cdot \sqrt{m}}.$$

Assume that an algorithm $\mathcal{A}$ exists for solving $\mathsf{RSIS}_{q,m,\beta}$. Then there exists an algorithm $\mathcal{B}$ for solving $\mathsf{MSIS}_{c,m^k,\gamma}$, where $\gamma = \frac{c}{q^k}(t \cdot \sqrt{m})^{k(d-1)}\beta^{kd}$.

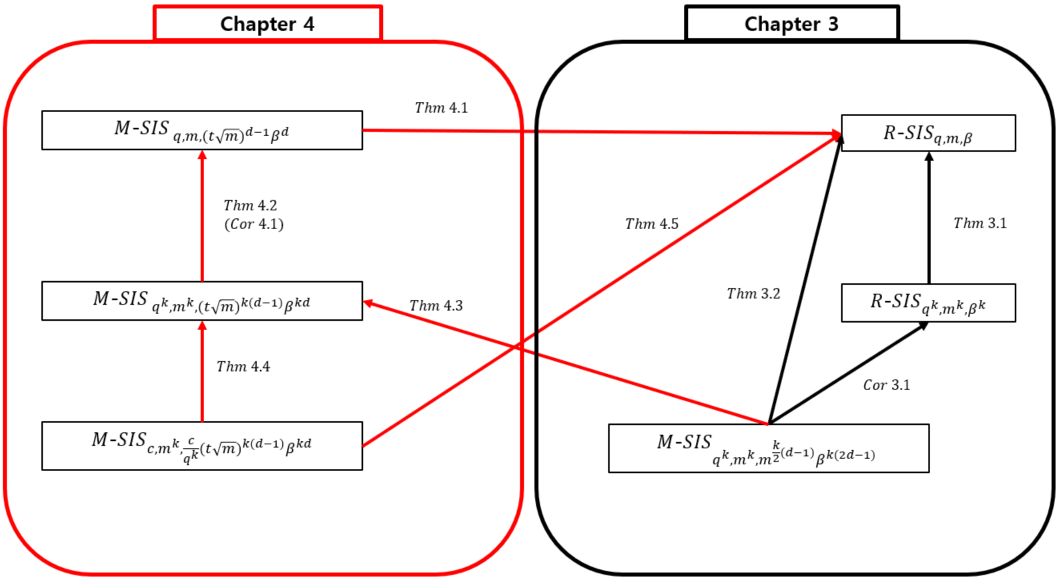
Chapter 4
Chapter 3
$M\text{-}SIS_{q,m,(t\sqrt{m})^{d-1}\beta^d}$
Thm 4.1
$R\text{-}SIS_{q,m,\beta}$
Thm 4.2
(Cor 4.1)
Thm 4.5
Thm 3.1
$M\text{-}SIS_{q^k,m^k,(t\sqrt{m})^{k(d-1)}\beta^{kd}}$
Thm 4.3
Thm 3.2
$R\text{-}SIS_{q^k,m^k,\beta^k}$
Thm 4.4
Cor 3.1
$M\text{-}SIS_{c,m^k,\frac{c}{q^k}(t\sqrt{m})^{k(d-1)}\beta^{kd}}$
$M\text{-}SIS_{q^k,m^k,m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}}$

Figure 4.5: Overview of the contributions for Chapter 4.

Chapter 5

HARDNESS OF VARIANT OF RING-LWE

In the proposed variant of RLWE, the error for generating an RLWE sample is reused as a secret of the other RLWE sample. The variant of the RLWE sample is of the form $(a, a \cdot s + x, a \cdot x + e)$, where ψ is the error distribution and $a \leftarrow R_q$, $s \leftarrow R_q$, and $x, e \leftarrow \psi$. We first define the variant of the RLWE problem (Re-RLWE) and prove the hardness of this problem.

5.1 Definition of Variant of Ring-LWE

To define the variant of the RLWE problem, we first define the variant of RLWE distribution, called Re-RLWE disitribution. This distribution is given as follows:

Definition 5.1 (Re-RLWE distribution)**.** *For a secret $s \in R_q$ and a distribution ψ over R_q, a sample from* Re-RLWE *distribution $\bar{A}_{s,\psi}$ over $R_q \times R_q \times R_q$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $x, e \leftarrow \psi$, and outputting (a, b, c), where*

$$\begin{aligned} b &= a \cdot s + x \mod qR \\ c &= a \cdot x + e \mod qR. \end{aligned}$$

Now, we define the variant of Re-RLWE problem, called Re-RLWE. Informally, Re-RLWE distribution is indistinguishable to the uniform distribution over $R_q \times R_q \times R_q$. The formal definition is given as follows:

Definition 5.2 (Re-RLWE problem)**.** *The average-case decision version of the* Re-RLWE *problem denoted* $\mathsf{Re\text{-}RLWE}_{q,\psi}$*, is to distinguish with non-negligible advantage between the sample from* $\bar{A}_{s,\psi}$ *and the uniformly at random from* $R_q \times R_q \times R_q$*.*

5.2 Hardness of Variant of Ring-LWE

In this section, we demonstrate the hardness of Re-RLWE. To prove that, we will prove that the Re-RLWE problem is harder than the RLWE problem as follows:

Theorem 5.1. *Let q be a prime and ψ be an error distribution. Assume that there exists an algorithm $\mathcal{A}$ in distinguishing the* $\mathsf{Re\text{-}RLWE}_{q,\psi}$ *distribution from the uniform distribution. Then there exists an algorithm $\mathcal{B}$ in distinguishing the* $\mathsf{RLWE}_{q,\psi}$ *distribution from the uniform distribution.*

Proof. Assume that $\mathcal{A}$ is a distinguisher of $\mathsf{Re\text{-}RLWE}_{q,\psi}$ with a non-negligible advantage. Then we can construct a distinguisher $\mathcal{B}$ against $\mathsf{Re\text{-}RLWE}_{q,\psi}$ as follows. $\mathcal{B}$ gets as inputs $a \in R_q$ and $b \in R_q$. Then $\mathcal{B}$ proceeds as follows.

- If a has no inverse, abort $\mathcal{B}$ and output reject.
- $u \leftarrow R_q$
- $c \leftarrow a^{-1} \cdot b + a \cdot u$
- Output $\mathcal{A}(a, c, b)$.

If the input of $\mathcal{B}$ is distributed according to the uniform distribution over $R_q \times R_q$, then c is also uniformly at random. If the input of $\mathcal{B}$ is distributed according to the RLWE distribution $A_{s,\psi}$ of the form $(a, b) = (a, a \cdot s + x)$, where $s, x \leftarrow \psi$, then we have

$$\begin{aligned} c &= a^{-1} \cdot b + a \cdot u \\ &= a^{-1}(a \cdot s + x) + a \cdot u \\ &= s + a^{-1} \cdot x + a \cdot u \\ &= s + a \cdot (a^{-2} \cdot x + u). \end{aligned}$$

Then $a^{-2} \cdot x + u$ is uniformly at random and independent of x as in [41]. Denote $s' = a^{-2} \cdot x + u$. Then $c = a \cdot s' + s$ and $(a, c, b) = (a, a \cdot s' + s, a \cdot s + x)$, which has the $\mathsf{Re\text{-}RLWE}_{q,\psi}$ distribution. Thus, we conclude that $\mathcal{B}$ has the same advantage as $\mathcal{A}$, which contradicts the hardness of $\mathsf{RLWE}_{q,\psi}$. □

Chapter 6

SAMPLING REDUCTION IN COMPACT MULTI-KEY HOMOMORPHIC ENCRYPTION

In this section, we introduce the modified multiplication keys and rotation keys for CMK-CKKS and CMK-BFV schemes using the Re-RLWE. In [1], to resolve the expansion of ciphertext, a common public is generated through the communication between users. In addition, by adding the multiplication keys and rotation keys generated by users in the server, a common multiplication key and rotation key are generated to lower the communication cost between users. Although many operations of the proposed scheme is similar to those in [1], the multiplication key and the rotation key generation are different. This generation method reuses the error used to generate the public key for the multiplication key. Also, to reduce the rotation key, we modify the CMK-HE.Setup, that is, we consider a more common reference string. In this way, the size of the multiplication key and the rotation key can be reduced compared to that in [1]. The following operation is the modified setup.

- ReCMK-HE.Setup(1^λ): Given a security parameter λ, set the RLWE dimension n, ciphertext modulus q, key distribution χ, and error distribution ψ over R. Let w be a half of the number of $\mathbb{Z}_{2N}^*$, that is, $w = |\mathbb{Z}_{2N}^*|/2$. Generate random polynomials $a, a_1, a_2 \ldots, a_{w-1}, a_w \leftarrow R_q$. Return the public parameter $pp =$

$(n, q, \chi, \psi, a, a_1, a_2, \ldots, a_{w-1}, a_w)$.

6.1 Variant of Compact Multi-Key CKKS Scheme

ReCMK-CKKS covers how to generate a modified multiplication key and a modified rotation key and how to operate multiplication between ciphertexts and rotation in the ciphertexts. The rest of the algorithm is the same as that in Section 2.4.

- ReCMK-CKKS.MultKeyGen(s_i, x_i): Set the modified multiplication key as

$$mk_i = a \cdot x_i + e_i + s_i \mod q,$$

  where $e_i \leftarrow \psi$.

- ReCMK-CKKS.ComMultKey$(mk_0, \ldots, mk_{k-1})$: Given all users' modified multiplication keys $mk_i = a \cdot x_i + e_i + s_i$, the server generates a common modified multiplication key

$$mk = \sum_{i=0}^{k-1} mk_i = a \cdot x + e + s \mod q.$$

- ReCMK-CKKS.Mult$(\mathsf{ct}_0, \mathsf{ct}_1; (pk, mk))$: Given two ciphertexts ct_0 and ct_1 at level q_ℓ, compute $\hat{\mathsf{ct}} = \mathsf{ct}_0 \otimes \mathsf{ct}_1 \in R_{q_\ell}^4$ and return the ciphertext

$$\bar{\mathsf{ct}} \leftarrow \text{ReCMK-CKKS.Relin}(\hat{\mathsf{ct}}; (pk, mk)) \in R_{q_\ell}^2$$

  as described in Algorithm 6.1.

- ReCMK-CKKS.RotKeyGen$(s_i, j; \tau_{t_1}(s_i), \tau_{t_2}(s_i))$: For each user i, fixed $j \in \{1, 2, \ldots, w\}$, and $t_1, t_2 \in \mathbb{Z}_{2N}^*$, set the modified rotation key

$$rk_i^{(t_1,t_2)} = (a_j, rk_{i,t_1}, rk_{i,t_2})$$

Algorithm 6.1 Relinearization of CKKS using the Re-RLWE

```
Input : $\hat{\mathsf{ct}} = (\hat{c}_0, \hat{c}_1, \hat{c}_2, \hat{c}_3)$, a common public key $(a, b)$, a common modified multiplication key $mk$
Output : $\bar{\mathsf{ct}} = (\bar{c}_0, \bar{c}_1) \in R_{q_\ell}^2$
1: $f_0 \leftarrow ab + mk$
2: $f_1 \leftarrow b^2$
3: $\bar{c}_0 \leftarrow \hat{c}_1 + \hat{c}_2 + \hat{c}_0 \cdot f_0 \mod q_\ell$
4: $\bar{c}_1 \leftarrow \hat{c}_3 + \hat{c}_0 \cdot f_1 \mod q_\ell$
```

as

(i) $x_i, e_i \leftarrow \psi$.

(ii) Set $rk_{i,t_1} = -a_j \cdot s_i + x_i + \tau_{t_1}(s_i) \mod q$.

(iii) Set $rk_{i,t_2} = -a_j \cdot x_i + e_i + \tau_{t_2}(s_i) - \tau_{t_1}(s_i) \cdot a_j \mod q$.

- ReCMK-CKKS.ComRotKey$(rk_0^{(t_1,t_2)}, \ldots, rk_{k-1}^{(t_1,t_2)})$: Given all users' modified rotation keys $rk_i^{(t_1,t_2)} = (a_j, rk_{i,t_1}, rk_{i,t_2})$, the server generates a common modified rotation key $rk^{(t_1,t_2)} = (a_j, rk_{t_1}, rk_{t_2})$ as

$$rk_{t_1} = \sum_{i=0}^{k-1} rk_{i,t_1} \mod q$$
$$rk_{t_2} = \sum_{i=0}^{k-1} rk_{i,t_2} \mod q.$$

- ReCMK-CKKS.Rot(ct; $rk^{(t_1,t_2)}, t)$ Given a ciphertext ct at level q_ℓ, compute and return the ciphertext $\bar{\mathsf{ct}}$ as described in Algorithm 6.2.

Remark 6.1. *In the* CKKS *scheme, not only rescaling method but also the special modulus technique is used to prevent the error from rapidly increasing. See [17],[18], and [24] for details.*

Algorithm 6.2 Rotation using the Re-RLWE

```
Input : ct = $(c_0, c_1)$, a common rotation key $rk^{(t_1,t_2)} = (a_j, rk_{t_1}, rk_{t_2})$ and $t \in \mathbb{Z}_{2N}^*$
Output : $\bar{\mathsf{ct}} = (\bar{c}_0, \bar{c}_1) \in R_{q_\ell}^2$
1: Compute $\hat{\mathsf{ct}} = (\hat{c}_0, \hat{c}_1) = (\tau_t(c_0), \tau_t(c_1))$
2: If $t = t_1$;
3:     $\bar{c}_0 \leftarrow \hat{c}_0 \cdot a_j \mod q_\ell$.
4:     $\bar{c}_1 \leftarrow \hat{c}_1 + \hat{c}_0 \cdot rk_{t_1} \mod q_\ell$.
5: else if $t = t_2$;
6:     $\bar{c}_0 \leftarrow \hat{c}_0 \cdot a_j^2 \mod q_\ell$.
7:     $\bar{c}_1 \leftarrow \hat{c}_1 + \hat{c}_0 \cdot (rk_{t_2} + a_j \cdot rk_{t_1}) \mod q_\ell$.
```

6.2 Variant of Compact Multi-Key BFV Scheme

The variant of the CMK-BFV (ReCMK-BFV) scheme is almost similar to ReCMK-CKKS defined in Section 6.1. However, there is only a slight difference in the algorithm of relinearization. In this section, we propose a variant of the multiplication of ReCMK-BFV as follows.

- ReCMK-BFV.Mult$(ct_0, ct_1; (pk, mk))$: Given two ciphertexts ct_0 and ct_1, compute $\hat{\mathsf{ct}} = \mathsf{ct}_0 \otimes \mathsf{ct}_1 \in R_{q_\ell}^4$ and return the ciphertext

$$\bar{\mathsf{ct}} \leftarrow \mathsf{ReCMK\text{-}BFV.Relin}(\hat{\mathsf{ct}}; (pk, mk)) \in R_{q_\ell}^2$$

as described in Algorithm 6.3.

Algorithm 6.3 Relinearization of BFV using the Re-RLWE

```
Input : $\hat{\mathsf{ct}} = (\hat{c}_0, \hat{c}_1, \hat{c}_2, \hat{c}_3)$, a common public key $(a, b)$, a common modified multiplication key $mk$
Output : $\bar{\mathsf{ct}} = (\bar{c}_0, \bar{c}_1) \in R_{q_\ell}^2$
1: Compute $\hat{c}_i' = \lfloor \frac{1}{\Delta} \hat{c}_i \rceil \mod q$
1: $f_0 \leftarrow ab + mk$
2: $f_1 \leftarrow b^2$
3: $\bar{c}_0 \leftarrow \hat{c}_1' + \hat{c}_2' + \hat{c}_0' \cdot f_0 \mod q$
4: $\bar{c}_1 \leftarrow \hat{c}_3' + \hat{c}_0' \cdot f_1 \mod q$
```

6.3 Correctness, Security, and Comparison

In this section, we show the correctness, security, and comparison of ReCMK-HE schemes. Since ReCMK-CKKS and ReCMK-BFV are defined similarly, we only consider ReCMK-CKKS. It is similar to the case of ReCMK-BFV.

6.3.1 Correctness

First, we show the correctness of multiplication in the proposed scheme. Let $s = \sum_{i=0}^{k-1} s_i \in R_q$ and $sk = (s, 1) \in R_q^2$ be a secret key. Let $\bar{\mathsf{ct}} = (c_0, c_1)$ and $\bar{\mathsf{ct}}' = (c_0', c_1')$ be ciphertexts corresponding to the messages m and m' with secret key sk, respectively. For multiplication, let $\bar{\mathsf{ct}}^{\times} = (c_0^{\times}, c_1^{\times})$ be multiplied by $\bar{\mathsf{ct}}$ and $\bar{\mathsf{ct}}'$ from Algorithm 6.1. Note that $\bar{\mathsf{ct}} \otimes \bar{\mathsf{ct}}' = (c_0c_0', c_1c_0', c_0c_1', c_1c_1')$. Since $(ab + mk, b^2)$ satisfies that

$$\begin{aligned}
\langle (ab + mk, b^2), sk \rangle &= (ab + mk) \cdot s + b^2 \\
&= a \cdot s \cdot b + mk \cdot s + b^2 \\
&= (x - b) \cdot b + mk \cdot s + b^2 \\
&= x \cdot (-a \cdot s + x) + (a \cdot x + e + s) \cdot s \\
&= x^2 + e \cdot s + s^2,
\end{aligned}$$

the ciphertext $\bar{\mathsf{ct}}^{\times}$ satisfies that

$$\begin{aligned}
\langle \bar{\mathsf{ct}}^{\times}, sk \rangle &= c_0^{\times} \cdot s + c_1^{\times} \\
&\approx c_1c_1' + (c_1c_0' + c_0c_1') \cdot s + c_0c_0' \cdot (f_0 \cdot s + f_1) \\
&= c_1c_1' + (c_1c_0' + c_0c_1') \cdot s + c_0c_0' \cdot (s^2 + x^2 + e \cdot s) \\
&= c_1c_1' + (c_1c_0' + c_0c_1') \cdot s + c_0c_0' \cdot s^2 \\
&= m \cdot m' \mod q.
\end{aligned}$$

Second, we show the correctness of rotation in the proposed scheme. For rotation, let $t \in \mathbb{Z}_{2N}^*$ and let $\bar{\mathsf{ct}}^{\mathsf{rot}_t} = (c_0^{\mathsf{rot}_t}, c_1^{\mathsf{rot}_t})$ be the rotation of $\bar{\mathsf{ct}} = (c_0, c_1)$ from Algorithm 6.2. Note that $\tau_t(\mathsf{ct}) = (\tau_t(c_0), \tau_t(c_1))$ and

$$\begin{aligned} rk_{t_2} + a_j \cdot rk_{t_1} &= -a_j \cdot x + e + \tau_{t_2}(s) - \tau_{t_1}(s) \cdot a_j + a_j \cdot (-a_j \cdot s + x + \tau_{t_1}(s)) \\ &= -a_j^2 \cdot s + e + \tau_{t_2}(s). \end{aligned}$$

If $t = t_1$, we have

$$\begin{aligned} \langle \bar{\mathsf{ct}}^{\mathsf{rot}_{t_1}}, sk \rangle &= \tau_{t_1}(c_0) \cdot a_j \cdot s + \tau_{t_1}(c_1) + \tau_{t_1}(c_0) \cdot rk_{t_1} \\ &= \tau_{t_1}(c_0) \cdot a_j \cdot s + \tau_{t_1}(c_1) + \tau_{t_1}(c_0) \cdot (-a_j \cdot s + x + \tau_{t_1}(s)) \\ &= \tau_{t_1}(c_1) + \tau_{t_1}(c_0) \cdot \tau_{t_1}(s) + \tau_{t_1}(c_0) \cdot x \\ &= \tau_{t_1}(c_1 + c_0 \cdot s) + \tau_{t_1}(c_0) \cdot x. \end{aligned}$$

If $t = t_2$, we have

$$\begin{aligned} \langle \bar{\mathsf{ct}}^{\mathsf{rot}_{t_2}}, sk \rangle &= \tau_{t_2}(c_0) \cdot a_j^2 \cdot s + \tau_{t_2}(c_1) + \tau_{t_2}(c_0) \cdot (rk_{t_2} + a_j \cdot rk_{t_1}) \\ &= \tau_{t_2}(c_0) \cdot a_j^2 \cdot s + \tau_{t_2}(c_1) + \tau_{t_2}(c_0) \cdot (-a_j^2 \cdot s + e + \tau_{t_2}(s)) \\ &= \tau_{t_2}(c_1) + \tau_{t_2}(c_0) \cdot \tau_{t_2}(s) + \tau_{t_2}(c_0) \cdot e. \\ &= \tau_{t_2}(c_1 + c_0 \cdot s) + \tau_{t_2}(c_0) \cdot e. \end{aligned}$$

6.3.2 Security

In this subsection, we prove that the proposed scheme satisfies the indistinguishability under chosen-plaintext attack (IND-CPA) security. we first show that the public key with the modified multiplication key is computationally indistinguishable from a uniform distribution over R_q^3. It is similar to the case of the rotation key.

Theorem 6.1. *The distribution of public keys with the modified multiplication keys is computationally indistinguishable to a uniform distribution over R_q^3 under the as-*

sumption of Re-RLWE *and circular security.*

Proof. Let pp be the $\mathsf{Re\text{-}RLWE}_{q,\psi}$ parameter generated in ReCMK-HE.Setup, where $w = \mathbb{Z}_{2N}^*$. We define the distribution $\mathcal{D}_0 = \{a, b, mk\}$ over R_q^3 as follows:

(i) $a \leftarrow R_q$ and $s \leftarrow \chi$, $x \leftarrow \psi$, and $b = -a \cdot s + x \mod q$

(ii) $e \leftarrow \psi$ and $mk = a \cdot x + e + s \mod q$.

Now, we consider the distribution $\mathcal{D}_1$ over R_q^3, which is obtained from $\mathcal{D}_0$ by modifying its definitions (i) and (ii) into

(i)' $a \leftarrow R_q$ and $b \leftarrow R_q$

(ii)' $mk \leftarrow R_q$.

From Theorem 5.1 and the circular security, we obtain that $\mathcal{D}_0$ and $\mathcal{D}_1$ are computationally indistinguishable. □

Now, we will show that the ReCMK-CKKS is IND-CPA secure under the Re-RLWE assumption with parameter $pp \leftarrow$ ReCMK-HE.Setup.

Theorem 6.2. *Let $pp \leftarrow$* ReCMK-HE.Setup *be the* Re-RLWE *parameter generated in the setup phase. Then the* ReCMK-CKKS *is IND-CPA secure under the* RLWE *and the* Re-RLWE *assumptions with parameter pp.*

Proof. Let $\mathcal{A}$ be an IND-CPA adversary for the ReCMK-CKKS. We consider a series of hybrids, where $\mathsf{Adv}_H[\mathcal{A}]$ denotes the success probability of $\mathcal{A}$ in hybrid H.

- **Hybrid** H_0: This is identical to the IND-CPA game, where the adversary gets a distributed public key (a, b) generated by CMK-CKKS.ComPk and the modified multiplication key mk generated by ReCMK-CKKS.ComMultKey. Also, the adversary gets encryption ct_0 and ct_1 of m_0 and m_1, respectively, computed using $\mathsf{CMK\text{-}CKKS.Enc}$. Note that the public key with the modified multiplication key consists of

$$(a, b, mk) := (a, -a \cdot s + x, a \cdot x + e + p \cdot s),$$

where $a \leftarrow R_q, s \leftarrow \chi$, and $x, e \leftarrow \psi$. Assume that there is a polynomial $t(\cdot)$ such that

$$\mathsf{Adv}_{H_0}[\mathcal{A}] := |\Pr[\mathcal{A}((a, b, mk), \mathsf{ct}_0) = 1] - \Pr[\mathcal{A}((a, b, mk), \mathsf{ct}_1) = 1]| > 1/t(\lambda), \tag{6.1}$$

where $\mathsf{ct}_i = \mathsf{CMK\text{-}CKKS.Enc}(m_i; \hat{pk} = (a, b))$ for $i = 0, 1$.

- **Hybrid** H_1: The hybrid H_1 is identical to H_0 except that b of the public key and the modified evaluation key d are chosen to be uniformly at random from R_q.

In H_1, the public key and modified multiplication key are uniformly at random. Also, mk is independent of (c_0, c_1) and (a, c_0) and (b, c_1) are computationally indistinguishable from the uniform distribution over R_q^2 since they can be viewed as two RLWE samples of secret v. Thus, we obtain that

$$\mathsf{Adv}_{H_1}[\mathcal{A}] = \mathsf{negl}(\lambda). \tag{6.2}$$

Now, we claim that

$$|\mathsf{Adv}_{H_0}[\mathcal{A}] - \mathsf{Adv}_{H_1}[\mathcal{A}]| \leq \mathsf{negl}(\lambda). \tag{6.3}$$

A ciphertext is generated by adding an encoded plaintext to a random encryption of zero. Hence we consider the random variables (a, b, d, c_0, c_1) over R_q^5 defined by

$$\begin{aligned} a &\leftarrow R_q \\ b &\leftarrow -a \cdot s + x \mod q \\ mk &\leftarrow a \cdot x + e + s \mod q, \end{aligned}$$

where $s \leftarrow \chi, x, e \leftarrow \psi$, and $(c_0, c_1) = v \cdot (a, b) + (e_0, e_1)$ for $v \leftarrow \chi$ and $e_0, e_1 \leftarrow \psi$. Now, we change the definition of (b, mk) as $b \leftarrow R_q$ and $mk \leftarrow R_q$. Then it is computationally indistinguishable by the Re-RLWE assumption with parameter pp. This means that

$$|\mathsf{Adv}_{H_0}[\mathcal{A}] - \mathsf{Adv}_{H_1}[\mathcal{A}]| \leq \mathsf{negl}(\lambda).$$

By combining (6.2) and (6.3), we obtain

$$\begin{aligned} \mathsf{Adv}_{H_0}[\mathcal{A}] &\leq \mathsf{Adv}_{H_1}[\mathcal{A}] + |\mathsf{Adv}_{H_0}[\mathcal{A}] - \mathsf{Adv}_{H_1}[\mathcal{A}]| \\ &= \mathsf{negl}(\lambda), \end{aligned}$$

which contradicts to (6.1). □

6.3.3 Comparison

In this subsection, the numerical results of the proposed scheme are compared to those in [1]. In our implementation, every number is stored as an unsigned 64-bit integer. Our implementation is performed on a computer with AMD Ryzen Threadripper PRO 3995WX CPU @ 2.70GHz processor on a multi-threaded mode. Also, the simulation utilizes the open-source in [42] and [43]. Table 6.1 shows the parameters used in CMK-CKKS and ReCMK-CKKS. In Table 6.2, the multiplication time is slightly increased by several milliseconds, but the multiplication key size is reduced by half. The rotation time performed twice also increases slightly. However, two rotation key sizes created to perform two rotations can be reduced by about 3/4. Therefore, the proposed scheme is more suitable for homomorphic encryption in limited memory environments.

Table 6.1: Parameters for compact multi-key CKKS

ID	$\log N$	$\log q$	$\log q_i$	No. of q_i's
I	13	218	49–60	4
II	14	438	53–60	8
III	15	881	54–60	16

Table 6.2: Comparison of keys and operations for each parameter in [1] and the proposed one

	ID	Mult. key size (MB)	Mult. time (ms)	Two Rot. key size (MB)	Two Rot. time (ms)
[1]	I	1.18	46.19	2.36	77.41
	II	4.46	86.94	8.92	144.86
	III	17.3	170.18	34.6	296.55
The proposed	I	0.59	49.28	1.77	82.04
	II	2.23	88.39	6.68	155.55
	III	8.65	178.21	25.95	316.58

Chapter 7

CONCLUSION

In this dissertation, the various reduction from MSIS to RSIS under some norm constraint of RSIS, and the ReCMK-HE scheme based on Re-RLWE are studied.

First, we showed that the $\mathsf{RSIS}_{q,m,\beta}$ problem is more difficult than the $\mathsf{MSIS}_{q^k,m^k,\beta'}$ problem, where $\beta' = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$. To show the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta'}$ to $\mathsf{RSIS}_{q,m,\beta}$, we derived two reductions:

(i) the reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$,

(ii) the reduction from $\mathsf{MSIS}_{q^k,m,\beta'}$ to $\mathsf{RSIS}_{q^k,m,\beta}$.

To prove (i), we used the property that the solution of $\mathsf{RSIS}_{q,m,\beta}$ is relatively prime to q. By finding m distinct solutions for $\mathsf{RSIS}_{q^k,m,\beta}$, we showed (ii). In (ii), we imposed the upper bound β on the norm of the solution of $\mathsf{RSIS}_{q^k,m,\beta}$. Combining the two reductions, we showed that it is possible to reduce $\mathsf{MSIS}_{q^k,m^k,\beta'}$ to $\mathsf{RSIS}_{q,m,\beta}$. Since β was imposed, we obtained the range of possible ranks of module d, which depends on the value of parameter q.

Second, we derived the reduction from $\mathsf{MSIS}_{c,m^k,\gamma}$ to $\mathsf{RSIS}_{q,m,\beta}$, where $\gamma = \frac{c}{q^k}(t\sqrt{m})^{k(d-1)}\beta^{kd}$ and c is a composite integer that has a factor q^k for some $k \geq 1$. To show this reduction, we proposed the three reductions:

(i) the reduction from $\mathsf{MSIS}_{q,m,\beta_1}$ to $\mathsf{RSIS}_{q,m,\beta}$,

(ii) the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ to $\mathsf{RSIS}_{q,m,\beta}$,

(iii) the reduction from $\mathsf{MSIS}_{q^k,m^k,\gamma}$ to $\mathsf{MSIS}_{c,m^k,\beta_2}$,

where $\beta_1 = (t\sqrt{m})^{d-1}\beta^d$, $\beta_2 = \beta_1^k = (t\sqrt{m})^{k(d-1)}\beta^{kd}$, c is a composite integer with a factor q^k, and $\gamma = \frac{c}{q^k}\beta_2 = \frac{c}{q^k}(t\sqrt{m})^{k(d-1)}\beta^{kd}$.

To show (i), we devised the new method to find m distinct solutions of $\mathsf{RSIS}_{q,m,\beta}$. This new method is to add randomness to the algorithm for solving $\mathsf{RSIS}_{q,m,\beta}$. Thus, we can devise an algorithm that gives m distinct solutions to the same instances of RSIS. Compared to the previous work [40], this reduction is preserved the same modulus and ring dimension. Also, the possible range of module rank for reduction from $\mathsf{MSIS}_{q,m,\beta}$ to $\mathsf{RSIS}_{q,m,\beta}$ could be doubled compared to that of Section 3.3.

To show (ii), we derived the method extending the reduction from $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ to $\mathsf{RSIS}_{q,m,\beta}$ shown in Theorem 3.1 to the reduction from $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ to $\mathsf{MSIS}_{q,m,\beta_1}$, where $\beta_2 = \beta_1^k = (t\sqrt{m})^{k(d-1)}\beta^{kd}$. Also, we showed that $\mathsf{MSIS}_{q^k,m^k,\beta_2}$ is more difficult than $\mathsf{MSIS}_{q^k,m^k,\beta_3}$ defined in Chapter 3, where $\beta_3 = m^{\frac{k}{2}(d-1)}\beta^{k(2d-1)}$ for $k \geq 1$ using the fact that MSIS becomes more difficult when the upper bound of MSIS is tighter. This means that RSIS is more difficult than MSIS, which is tighter than the MSIS in Chapter 3. In Chapter 3, all reductions depend on the prime modulus q. However, in Chapter 4, we proposed the reductions between the MSIS problems with the different modulus. Combining three reductions, we obtained the reduction from $\mathsf{MSIS}_{c,m^k,\gamma}$ to $\mathsf{RSIS}_{q,m,\beta}$.

Third, we proposed a variant of RLWE by reusing error, called Re-RLWE, where we can reduce the sizes of the multiplication key and the rotation keys. To define this problem, we defined the Re-RLWE distribution $\bar{A}_{s,\psi}$ over $R_q \times R_q \times R_q$, which is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $x, e \leftarrow \psi$, and output (a, b, c), where $b = a \cdot s + x \mod qR$, and $c = a \cdot x + e \mod qR$. Next, we defined the Re-RLWE problem, which is to distinguish with non-negligible advantage between the sample from $\bar{A}_{s,\psi}$ and the uniformly at random from $R_q \times R_q \times R_q$. And we proved that the RE-RLWE problem is more difficult than the RLWE problem.

Lastly, we suggested the variant of CMK-HE, called ReCMK-HE, which has the modified multiplication keys and the modified evaluation keys with a reduced key size. Due to the modified multiplication key and the modified rotation key, the multiplication and rotation operation times increased slightly. However, the multiplication key was reduced by about half, and the rotation keys were reduced to 3/4 compared to the original scheme [1].

Bibliography

[1] J. Park, Homomorphic encryption for multiple users with less communications, IEEE Access 9 (2021) 135915–135926.

[2] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM review 41 (2) (1999) 303–332.

[3] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, et al., Status report on the first round of the NIST post-quantum cryptography standardization process, US Department of Commerce, National Institute of Standards and Technology, 2019.

[4] M. Ajtai, Generating hard instances of lattice problems, in: Proceedings of the 28th annual ACM symposium on Theory of computing, 1996, pp. 99–108.

[5] P. Bert, P.-A. Fouque, A. Roux-Langlois, M. Sabt, Practical implementation of ring-SIS/LWE based signature and IBE, in: International Conference on Post-Quantum Cryptography, Springer, 2018, pp. 271–291.

[6] L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky, Lattice signatures and bimodal Gaussians, in: Annual Cryptology Conference, Springer, 2013, pp. 40–56.

[7] V. Lyubashevsky, Lattice signatures without trapdoors, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2012, pp. 738–755.

[8] D. Micciancio, O. Regev, Lattice-based cryptography, in: Post-quantum Cryptography, Springer, 2009, pp. 147–191.

[9] I. Chillotti, N. Gama, M. Georgieva, M. Izabachene, Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds, in: International Conference on the Theory and Application of Cryptology and Iinformation Security, Springer, 2016, pp. 3–33.

[10] C. Peikert, Public-key cryptosystems from the worst-case shortest vector problem, in: Proceedings of the 41th Annual ACM Symposium on Theory of Computing, 2009, pp. 333–342.

[11] V. Lyubashevsky, C. Peikert, O. Regev, On ideal lattices and learning with errors over rings, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2010, pp. 1–23.

[12] A. Langlois, D. Stehlé, Worst-case to average-case reductions for module lattices, Designs, Codes and Cryptography 75 (3) (2015) 565–599.

[13] M. R. Albrecht, A. Deo, Large modulus ring-LWE $\geq$ module-LWE, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2017, pp. 267–296.

[14] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the 41th Annual ACM Symposium on Theory of Computing, 2009, pp. 169–178.

[15] Z. Brakerski, V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) LWE, SIAM Journal on Computing 43 (2) (2014) 831–871.

[16] Z. Brakerski, C. Gentry, V. Vaikuntanathan, (leveled) fully homomorphic encryption without bootstrapping, ACM Transactions on Computation Theory (TOCT) 6 (3) (2014) 1–36.

[17] J. H. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2017, pp. 409–437.

[18] J. H. Cheon, K. Han, A. Kim, M. Kim, Y. Song, A full RNS variant of approximate homomorphic encryption, in: International Conference on Selected Areas in Cryptography, Springer, 2018, pp. 347–368.

[19] J.-W. Lee, E. Lee, Y. Lee, Y.-S. Kim, J.-S. No, High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2021, pp. 618–647.

[20] W. Jung, S. Kim, J. H. Ahn, J. H. Cheon, Y. Lee, Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs, IACR Transactions on Cryptographic Hardware and Embedded Systems (2021) 114–148.

[21] W. Xu, B. Wang, J. Liu, Y. Chen, P. Duan, Z. Hong, Toward practical privacy-preserving linear regression, Information Sciences (2022).

[22] Y. Liu, Y. Luo, Y. Zhu, Y. Liu, X. Li, Secure multi-label data classification in cloud by additionally homomorphic encryption, Information Sciences 468 (2018) 89–102.

[23] W. Ding, Z. Yan, R. H. Deng, Encrypted data processing with homomorphic re-encryption, Information Sciences 409 (2017) 35–55.

[24] H. Chen, W. Dai, M. Kim, Y. Song, Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 395–412.

[25] H. Chen, I. Chillotti, Y. Song, Multi-key homomorphic encryption from TFHE, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2019, pp. 446–472.

[26] J. Zhang, Z. L. Jiang, P. Li, S. M. Yiu, Privacy-preserving multikey computing framework for encrypted data in the cloud, Information Sciences 575 (2021) 217–230.

[27] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, K. Chen, Multi-key privacy-preserving deep learning in cloud computing, Future Generation Computer Systems 74 (2017) 76–85.

[28] Z. Brakerski, R. Perlman, Lattice-based fully dynamic multi-key FHE with short ciphertexts, in: Annual International Cryptology Conference, Springer, 2016, pp. 190–213.

[29] A. Aloufi, P. Hu, Collaborative homomorphic computation on data encrypted under multiple keys, arXiv preprint arXiv:1911.04101 (2019).

[30] C. Mouchet, J. Troncoso-Pastoriza, J.-P. Bossuat, J.-P. Hubaux, Multiparty homomorphic encryption from ring-learning-with-errors, Cryptology ePrint Archive (2020).

[31] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, D. Wichs, Multiparty computation with low communication, computation and interaction via threshold fhe, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2012, pp. 483–501.

[32] J. Blömer, J.-P. Seifert, On the complexity of computing short linearly independent vectors and short bases in a lattice, in: Proceedings of the 31th annual ACM symposium on Theory of Computing, 1999, pp. 711–720.

[33] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in: Proceedings of the 40th annual ACM Symposium on Theory of Computing, 2008, pp. 197–206.

[34] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM) 56 (6) (2009) 1–40.

[35] B. Applebaum, D. Cash, C. Peikert, A. Sahai, Fast cryptographic primitives and circular-secure encryption based on hard learning problems, in: Annual International Cryptology Conference, Springer, 2009, pp. 595–618.

[36] D. Micciancio, C. Peikert, Hardness of SIS and LWE with small parameters, in: Annual Cryptology Conference, Springer, 2013, pp. 21–39.

[37] C. Peikert, A. Rosen, Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices, in: Theory of Cryptography Conference, Springer, 2006, pp. 145–166.

[38] V. Lyubashevsky, D. Micciancio, Generalized compact knapsacks are collision resistant, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 144–155.

[39] D. Micciancio, O. Regev, Worst-case to average-case reductions based on gaussian measures, SIAM Journal on Computing 37 (1) (2007) 267–302.

[40] Z. Koo, J.-S. No, Y.-S. Kim, Reduction from module-sis to ring-sis under norm constraint of ring-sis, IEEE Access 8 (2020) 140998–141006.

[41] C. E. Shannon, Communication theory of secrecy systems, The Bell System Technical Journal 28 (4) (1949) 656–715.

[42] B. Shoshany, A C++17 thread pool for high-performance scientific computing, arXiv e-prints (May 2021). arXiv:2105.00613, doi:10.5281/zenodo.4742687.

[43] S. Sapir, W. Sagi, Memorypool for c++, `https://github.com/DevShiftTeam/AppShift-MemoryPool`, accessed: 2021-12-28 (2021).

초록

이 학위 논문에서는 i) 모듈-짧은 정수해 문제 (MSIS) 에서 환-짧은 정수해 문제 (RSIS) 로의 환원방법, ii) 모듈-짧은 정수해 문제 (MSIS)에서 환-짧은 정수해 문제 (RSIS)로의 향상된 환원방법, iii) 변형된 RLWE의 도입과 그 어려움, iv) 변형된 RLWE를 기반으로 한 변형된 compact-다중 키 동형 암호 (CMK-HE)에 대한 방법들이 연구되었다.

첫 번째로 RSIS의 특정 조건 하에서 MSIS에서 RSIS로의 환원을 제안한다. 이 환원을 보이기 위해, 두가지의 환원 방법을 보인다. 먼저 $\mathsf{RSIS}_{q^k,m^k,\beta^k}$ 문제에서 $\mathsf{RSIS}_{q,m,\beta}$로의 환원을 보인다. 그리고 RSIS의 특정 조건 하에서 $\mathsf{MSIS}_{q^k,m^k,\beta'}$ 문제에서 $\mathsf{RSIS}_{q^k,m^k,\beta^k}$문제로의 환원을 보인다. 두 결과를 통해 RSIS 문제가 MSIS의 문제보다 특정 조건 하에서 더욱 어렵다고 할 수 있고, 환원된 MSIS가 가능한 Rank의 범위를 제공한다.

두 번째로, 기본의 MSIS 문제에서 RSIS로의 환원보다 더욱 향상된 방법을 제시한다. RSIS문제와 같은 modulus와 같은 환-차원을 가지는 MSIS문제보다 RSIS의 문제가 더 어려움을 제안한다. 이 방법을 통해 기존의 MSIS가 가능한 Rank의 범위를 2배가량 증가시킬 수 있다. 그리고 첫 번째 방법에서 사용된 MSIS보다 두 번째 방법에서 사용된 MSIS가 더욱 어려움을 보였다. 또한, 소수 modulus를 갖는 MSIS이 합성수 modulus를 갖는 MSIS보다 더 어렵다는 것을 제안한다. 이 방법들을 통해 소수 modulus를 갖는 RSIS가 합성수 modulus를 갖는 MSIS보다 더 어렵다고 말할 수 있다.

세 번째로, RLWE 샘플에서 사용된 에러를 재사용한 새로운 문제인 Re-RLWE를 제안한다. 이 문제를 정의하기 위해 Re-RLWE분포를 정의하고, 그 어려움을 증

명한다.

마지막으로, Re-RLWE 기반의 변형된 compact 다중 키 동형암호를 제안한다. 이 암호시스템은 변형된 곱셈키와 변형된 회전키들을 가지며, 기존의 compact 다중 키 동형암호와 비교하여 키의 크기가 감소함을 보일 수 있다.

주요어: Learning with errors, 모듈-lerarning with errors, 모듈-짧은 정수해 문제, 다중 키 동형 암호, 환-learning with errors, 환-짧은 정수해 문제, 짧은 정수해 문제

학번: 2018-38214