이학박사 학위논문

Mathematical Analysis of Multilinear Maps over the Integers

(정수에서 정의된 다중 선형 함수의 수학적 분석)

2016년 8월

서울대학교 대학원

수리과학부

류한솔

Mathematical Analysis of Multilinear Maps over the Integers

(정수에서 정의된 다중 선형 함수의 수학적 분석)

지도교수 천 정 희

이 논문을 이학박사 학위논문으로 제출함

2016년 4월

서울대학교 대학원

수리과학부

류한솔

류한솔의 이학박사 학위논문을 인준함

2016년 6월

위 원 장	김 명 환	(인)
부 위 원 장	천 정 희	(인)
위 원	이 향 숙	(인)
위 원	Damien Stehlé	(인)
위 원	David Donghoon Hyeon	(인)

Mathematical Analysis of Multilinear Maps over the Integers

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Hansol Ryu

Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

August 2016

Abstract

Mathematical Analysis of Multilinear Maps over the Integers

Hansol Ryu

Department of Mathematical Sciences
The Graduate School
Seoul National University

Multilinear maps have lots of cryptographic applications. Until now, there are three types of multilinear maps: the first is constructed using ideal lattices, the second is defined over the integers, and the last is graph-induced one. However none of them have reduction to well-known hard problems. More serious matter is that they are all proven insecure when low-level encodings of zero are provided .

Especially, for multilinear maps over the integers, construction and analysis are being repeated. At Crypto 2013, Coron, Lepoint, and Tibouchi proposed a multilinear map using CRT (CLT13). However, it was revealed to be insecure so-called CHLRS attack (CHL$^+$15). After then, several attempts have been made to repair the scheme, but quickly proven insecure by extended CHLRS attack. The same authors revised their scheme at Crypto 2015 again.

In this thesis, we describe attacks against CLT15. Our attacks share the essence of the cryptanalysis of CLT13 and exploits low level encodings of zero, provided by a ladder, as well as other public parameters. As in CHL$^+$15, this

leads to finding all the secret parameters of κ-multilinear maps in polynomial time of the security parameter. As a result, CLT15 is fully broken for all possible applications, while the security of CLT13 is not known when low-level encodings are not provided.

Key words: Multilinear maps, graded encoding schemes
Student Number: 2009-20267

Contents

List of Figures

Chapter 1

Introduction

The cryptographic bilinear map has many applications, including tripartite Diffie-Hellman [Jou00], and identity-based encryption [BF01]. Boneh and Silverberg formalized the concept of multilinear map and described interesting applications, including multipartite Diffie-Hellman and very efficient broadcast encryption [BS03]. After that, there were many researches to apply multilinear maps [RS09, PTT10] despite of absence of multilinear maps.

In 2013, Garg, Gentry, and Halevi proposed a multilinear maps from ideal lattices(GGH13, for short) [GGH13]. It has similar features with somewhat homomorphic encryption scheme, it is a "noisy" map. So it is little bit differ from ideal multilinear map considered in [BS03], but it enables multipartite Diffie-Hellman key exchange and many other applications. The security of GGH13 is based on a new problem so the authors gave diverse analysis of it. Nonetheless, now GGH13 suffers from the attacks [HJ15, CJL16, MSZ16].

We briefly describe the GGH13 multilinear map. It uses a polynomial ring and has secrets $g \in R$ and $z \in R/qR$ for some integer parameter q. The space of encoded values is a coset space $R/\mathcal{I}$, where $\mathcal{I} = \langle g \rangle$. Then the encoding of $e + \mathcal{I}$ is $c/z \bmod q$, where $c \in e + \mathcal{I}$ and short. Since it is a kind

of noisy map, only restricted number of multiplications are allowed, we say κ. The addition and multiplication are defined well as long as the numerator remains short.

GGH13 publishes a zero-testing parameter $\boldsymbol{p}_{zt} = h\frac{z^\kappa}{g} \bmod q$, where h is not too large. Then multiplying with level-κ encoding gives small value only when the encoded value is zero. Hence one can decide whether an encoding is zero, publicly. Using this property, one can solve some problems on GGH13, such as Graded Decisional Diffie-Helman assumption (GDDH), subgroup membership (SubM), and decisional linear (DLIN) problems. Actually, all the other attacks on GGH13 also probe for weak spots by using this zero-testing parameter.

Shortly afterwards, Coron, Lepoint, and Tibouchi proposed another candidate of multilinear maps (CLT13, for short) [CLT13]. It is constructed over the integers and gives the first implementation of multilinear maps [CLT13]. The most recent candidate, called GGH15, was suggested by Gentry, Gorbunov, and Halevi using a directed acyclic graph [GGH15].

In [CLT13], it was claimed that CLT13 is robust against a zeroizing attack. Hence, CLT13 supports the GDDH, SubM, and DLIN problems are hard in it, while GGH13 supports only the GDDH.

However, Cheon, Han, Lee, Ryu, and Stehlé proposed an attack, called CHLRS, on the scheme [CHL+15], which runs in polynomial time and recovers all secrets. As in the zeroizing attack of GGH13, the attack utilizes public low level encodings of zero, which allows an encoding to be generated without the secret values being known. The core of the attack is to compute several zero-testing values related to one another. Then, one can construct a matrix, the eigenvalues of which consist of the CRT component of e, which is $e \bmod p_i$ for some encoding e, where $p_1, \cdots, p_n$ are secret values of the scheme. Then, it reveals all the secrets of the scheme.

In response, two attempts have been made to make CLT13 secure against

the CHLRS attack [GGHZ14, BWZ14]. However, both are shown to be insecure in [CGH$^+$15]. At the same time, another fix of CLT13 was proposed at Crypto15 by Coron, Lepoint, and Tibouch (CLT15, for short) [CLT15]. CLT15 is almost the same as the original scheme, except in the zero-testing parameter and procedure. To prevent zero-testing values from being obtained in CLT13, the authors did not publish the modulus x_0 and performed zero-testing in independent modulus N. They claimed that it is secure against a CHLRS attack, because a zero-testing value of an encoding e depends on the CRT components of e non-linearly.

We briefly introduce the CLT15 scheme. It is a graded encoding scheme and its level-t encoding e is an integer satisfying $e \equiv \frac{r_{it}g_i+m_i}{z^t} \mod p_i$ for $1 \leq i \leq n$, where $p_1, \cdots, p_n$ are secret primes, z is a random invertible integer in $\mod \prod p_i$, $(m_1, \cdots, m_n) \in \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ is a plaintext for secret moduli $g_1, \cdots, g_n$, and $r_{1t}, \cdots, r_{nt}$ are random noises. Then, it can be written as $\sum_{i=1}^{n}[r_{it}+m_i/g_i]_{p_i}u_{it}+a_tx_0$ for some integer a_t, where $u_{it} = \left[\frac{g_i}{z^t}\left(\frac{x_0}{p_i}\right)^{-1}\right]_{p_i}\frac{x_0}{p_i}$ for $1 \leq i \leq n$.

The zero-testing of level-κ encoding operates as follows. For a zero-testing parameter $\boldsymbol{p}_{zt}$ and a level-κ encoding $e = \sum_{i=1}^{n}[r_i + m_i/g_i]_{p_i}u_{i\kappa} + ax_0$, which is smaller than x_0,

$$\boldsymbol{p}_{zt} \cdot e \equiv \sum_{i=1}^{n}[r_i + m_i/g_i]_{p_i} \cdot v_i + av_0 \mod N,$$

where $v_i = [\boldsymbol{p}_{zt} \cdot u_{i\kappa}]_N$ and $v_0 = [\boldsymbol{p}_{zt} \cdot x_0]_N$. Note that v_i's are small as compared to N for all $0 \leq i \leq n$ and the size of a depends on that of e. Hence, the right hand side is small when all m_i's are zero. Therefore, it is used to determine whether it constitutes an encoding of zero or not.

Since av_0 exceeds N for a large e, the zero-testing is effective only when the size of e is small. However, the size of the encodings is almost doubled through multiplication and is too large to allow one to obtain a correct zero-testing value. Accordingly, CLT15 publishes encodings of zero of various sizes

(called ladders) to reduce the size of the encodings. The ladders are of the form $X_j = \sum_{i=1}^{n} s_{ij} u_{i\kappa} + q_j x_0$, where $0 \leq j \leq M$ for some integers q_j, and for small integers $s_{ij}, 1 \leq i \leq n, 0 \leq j \leq M$, and the size of X_j is about $2^j x_0$. For an encoding e larger than x_0,one can obtain e', an encoding of the same plaintext, the size of which is reduced using a ladder. Then, it can be written as $e' = e - \sum_{j=0}^{M} b_j X_j$, for some $b_0, \cdots, b_M \in \{0, 1\}$.

The points of a CHLRS attack can be divided into two parts. The first is that, for a level-κ encoding of zero $e = \sum_{i=1}^{n} [\frac{r_i g_i}{z^\kappa} (\frac{x_0}{p_i})^{-1}]_{p_i} \frac{x_0}{p_i} + a x_0$,

$$\boldsymbol{p}_{zt} \cdot e \mod x_0 = \sum_{i=1}^{n} r_i \hat{v_i},$$

where $\hat{v_i}$ is common to all the encodings in CLT13, holds over the integers. The second point is that the zero-testing value of a product of two encodings is a quadratic form of some values related to each encoding. More precisely, for two encodings $e_1 = \sum_{i=1}^{n} [\frac{r_{i1} g_i}{z^t} (\frac{x_0}{p_i})^{-1}]_{p_i} \frac{x_0}{p_i} + a_1 x_0$ and $e_2 = \sum_{i=1}^{n} [\frac{r_{i2}}{z^{\kappa - t}} (\frac{x_0}{p_i})^{-1}]_{p_i} \frac{x_0}{p_i} + a_2 x_0$, the product is $e_1 e_2 \equiv \sum_{i=1}^{n} [\frac{r_{i1} r_{i2} g_i}{z^\kappa} (\frac{x_0}{p_i})^{-1}]_{p_i} \frac{x_0}{p_i}$ mod x_0. Therefore, the zero-testing value of $e_1 e_2$ is

$$\boldsymbol{p}_{zt} \cdot e_1 e_2 \mod x_0 = \sum_{i=1}^{n} r_{i1} r_{i2} \hat{v_i}.$$

Let us look at CLT15 in these aspects. For a level-κ encoding of zero $e = \sum_{i=1}^{n} r_i u_{i\kappa} + a x_0$, the zero-testing value of e is written as

$$\boldsymbol{p}_{zt} \cdot e \mod N = \sum_{i=1}^{n} r_i v_i + a v_0,$$

for common v_i's, similar to CLT13. Let e_1 be a level-t encoding of zero, e_2 be a level-$(\kappa - t)$ encoding, and e be a product of e_1 and e_2. Then, these can be written as $e_1 = \sum_{i=1}^{n} r_{i1} u_{it} + a_1 x_0$, $e_2 = \sum_{i=1}^{n} r_{i2} u_{i\kappa - t} + a_2 x_0$, and $e = \sum_{i=1}^{n} r_{i1} r_{i2} u_{i\kappa} + a x_0$, for some integers $a, a_1, a_2, r_{i1}, r_{i2}, 1 \leq i \leq n$, where a is a quadratic form of $a_1, a_2, r_{i1}, r_{i2}, 1 \leq i \leq n$. Since the size of e is larger than that of x_0, we need to reduce the size of e to performzero-testing. Let e'

be a size-reduced encoding of e; then, it is of the form $e' = e - \sum_{j=0}^{M} b_j X_j = \sum_{i=1}^{n}(r_{i1}r_{i2} - \sum_{j=0}^{M} b_j s_{ij})u_{i\kappa} + (a - \sum_{j=0}^{M} b_j q_j)x_0$, for some $b_0, \cdots, b_M \in \{0,1\}$. In this case, the zero-testing value gives

$$\begin{aligned}
[\boldsymbol{p}_{zt} \cdot e']_N &= \Big[\boldsymbol{p}_{zt} \cdot \Big(e - \sum_{j=0}^{M} b_j X_j\Big)\Big]_N \\
&= \sum_{i=1}^{n}\Big(r_{i1}r_{i2} - \sum_{j=0}^{M} b_j s_{ij}\Big)v_i + \Big(a - \sum_{j=0}^{M} b_j q_j\Big)v_0 \\
&= \sum_{i=1}^{n}\Big(r_{i1}r_{i2}\Big)v_i + av_0 - \sum_{j=0}^{M} b_j\Big(\sum_{i=1}^{n} s_{ij}v_i + q_j v_0\Big).
\end{aligned}$$

Therefore, if one has $\sum_{i=1}^{n} s_{ij}v_i + q_j v_0$ for all j, one can compute $\sum_{i=1}^{n}(r_{i1}r_{i2})v_i + av_0$ and follow a CHLRS attack strategy. We define a function ϕ such that the above equation is written as

$$\boldsymbol{p}_{zt} \cdot e' \mod N = \phi(e) - \sum_{j=0}^{M} b_j \cdot \phi(X_j). \tag{1.0.1}$$

Note that $\phi(e) = [\boldsymbol{p}_{zt} \cdot e]_N$, when e is a level-κ encoding of zero smaller than x_0. Since X_j's are level-κ encodings of zero and the size of X_0 is small, one can obtain$\phi(X_0)$ by the zero-testing procedure. $\phi(X_j)$ can be obtained inductively, because the size-reduced X_j is a linear summation of $X_0, \cdots, X_{j-1}, X_j$. When one has $\phi(X_j)$ in hand, it is easy to calculate $\phi(e)$ for a level-κ encoding of 0 of arbitrary size using Equation (1.0.1).

We look into the exact expression of the ϕ-value over $\mathbb{Q}$. By using $(n+1)$ level-t encodings of zero and $(n+1)$ level-$(\kappa - t)$ encodings, we constitute matrix equations that consist only of a product of matrices. As in [CHL+15], we have a matrix, the eigenvalues of which consist of the CRT components of an encoding.

It needs only ladders and two level-0 encodings, and runs in polynomial time. Therefore it is totally broken all possible applications in contrast with CLT13 is still secure when no low-level encodings of zero are provided.

Currently, the construction and analysis of multilinear maps are being repeated. As seen in the case of multilinear maps over the integers, simple technical modification can not makes scheme secure without fundamental understanding of attack. The construction of secure multilinear map only can be started from deep perception of attacks and so it is meaningful to examine cryptanlaysis of multilinear maps.

Organization of the Paper

In section 2, we give a description of multilinear map and graded encodings schemes. We also introduce a multilinear map procedures and additional problems related to multiinear maps. In section 3, we recall CLT13 multilinear map and explain zero-testing detailed. After that CHLRS attack will be introduced. CLT15 multilinear map is also explained concentrated on the difference with CLT13. In the last section, we introduce a cryptanalysis of CLT15 multilinear map which takes polynomial time in security parameter.

Contributions

The thesis contains a joint work with Jung Hee Cheon and Changmin Lee [CLR15] which appears in Eurocrypt 2016 as a merged paper [CFL+16].

List of Papers

- [CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the New CLT Multilinear Maps. IACR Cryptology ePrint Archive, 2015:934.
- [CFL+16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the New CLT Multilinear Map over the Integers. EUROCRYPT 2016, 509-536.

Chapter 2

Introduction to Multilinear Maps

At first, we define some notations to be used. After then, we give a brief introduction to multilinear maps.

2.1 Notation

For a finite set S, we use $s \leftarrow S$ to denote the operation of uniformly choosing an element s from S. For an integer p, $\mathbb{Z}_p$ is a ring of integers modulo p, and $x \mod p$ and $[x]_p$ denotes a number in $\mathbb{Z} \cap \left(-\frac{p}{2}, \frac{p}{2}\right]$, which is congruent to x modulo p. For $x, y, p \in \mathbb{Z}$, $x \equiv y \mod p$ or $x \equiv_p y$ means that x is congruent to y modulo p.

We use lower-case bold letters to denote vectors whereas upper-case bold letters are used to denote matrices. For an $n \times n$ square matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$, we use (m_{ij}) to represent a matrix $\boldsymbol{M}$, where m_{ij} is the (i, j)-th component of $\boldsymbol{M}$. Let $\boldsymbol{M}^T$ be the transpose of $\boldsymbol{M}$ and $\|\boldsymbol{M}\|_\infty$ be the $\max_i \sum_{j=1}^{n} |m_{ij}|$ which is the maximum of 1-norm of row vectors. We denote by $\mathsf{diag}(d_1, \cdots, d_n)$ the

diagonal matrix with diagonal coefficients equal to $d_1, \cdots, d_n$.

Chinese Remainder Theorem. Given n prime numbers $p_1, \cdots, p_n$, define p_i^* as in [Hal15]:

$$p_i^* = \prod_{j \neq i} p_j = \frac{x_0}{p_i},$$

where $x_0 = \prod_{1 \leq j \leq n} p_i$. For $(x_1, \ldots, x_n) \in \mathbb{Z}^n$, let $\mathsf{CRT}_{(p_i)}(x_i)$ denote the unique integer in $\mathbb{Z} \cap [0, \prod p_i)$ such that $\mathsf{CRT}_{(p_i)}(x_i) \bmod p_i = x_i \bmod p_i$, as per the Chinese Remainder Theorem.

It is useful to observe that for any $(x_1, \ldots, x_n) \in \mathbb{Z}^n$:

$$\mathsf{CRT}_{(p_i)}(x_i p_i^*) = \sum_i x_i p_i^* \bmod \prod_i p_i. \tag{2.1.1}$$

2.2 Multilinear Maps and Graded Encoding Schemes

Boneh and Silverberg introdeced cryptographic multilinear map [BS03], as a natural generalization of bilinear maps. A multilinear map is defined as follows.

Definition 2.2.1 (Multilinear Map [BS03])**.** Given κ+1 cyclic groups $G_1, \cdots, G_\kappa$, G_T of the same prime order p, a map $e : G_1 \times \cdots \times G_\kappa \to G_T$ is a *κ-multilinear map* iff it satisfies the following two properties:

1. For all $a_1, \ldots, a_\kappa \in \mathbb{Z}_p$ and $\{g_i \in G_i\}_{i=1,\cdots,\kappa}$,

$$e(g_1^{a_1}, \cdots, g_\kappa^{a_\kappa}) = e(g_1, \cdots, g_\kappa)^{a_1 \cdots a_\kappa}.$$

2. If $\{g_i \in G_i\}_{i=1,\cdots,\kappa}$ are all generators of their respective groups, then $e(g_1, \cdots, g_\kappa)$ is a generator of G_T.

This definition is slightly differ from [BS03], it gave the symmetric multilinear map which is the case $G_1 = \cdots = G_\kappa$. We follow the definition of assymmetric multilinear map as in [Rot13, GGH13]. In their paper, Boneh and Silverberg suggested applications which are one-round multipartite key exchange and efficient broadcast encryption. Multilinear map has lots of applications, however the construction of multilinear map has not been made after 10 years when it was suggested.

In 2013, Garg, Gentry and Halevi proposed a candidate multilinear maps from ideal lattices [GGH13]. It has similar features as homomorphic encryption, it is a "noisy" multilinear map. The noise of an encoding increases after operations, especially multiplicaion rapidly increases it. After GGH13, several multilinear maps are suggested and all of them share the concept of [GGH13]. It does not fit the exact definition of cryptographic multilinear

map, it is a somewhat relaxed version of multilinear map. So the authors suggested a slightly different definition. whick called graded encoding system. It has a graded structure and supports two kinds of operation, addition and multiplication. Hence it is defined using a ring not a group.

Now we give a precise definition of κ-graded encoding system suggested in [GGH13].

Definition 2.2.2 (Graded Encoding System [GGH13])**.** A *κ-graded encoding system* consists of a ring R and a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subset \{0,1\}^* | \alpha \in R, 0 \leq i \leq \kappa\}$, with the following properties:

1. For every fixed i, the sets $S_i^{(\alpha)}$ are disjoint

2. There is an associative binary operation '+' and a self-inverse unary operation '$-$' on $\{0,1\}^*$ such that for every $\alpha_1, \alpha_2 \in R$, every index $i \leq \kappa$, and every $u_1 \in S_i^{(\alpha_1)}, u_2 \in S_i^{(\alpha_2)}$, it holds that:

$$u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)} \quad \text{and} \quad -u_1 \in S_i^{(-\alpha_1)}$$

   where $\alpha_1 + \alpha_2$ and $-\alpha_1$ are addition and negation in R.

3. There is an associative binary operation '$\times$' on $\{0,1\}^*$ such that for every $\alpha_1, \alpha_2 \in R$, every i_1, i_2 such that $i_1 + i_2 \leq \kappa$, and every $u_1 \in S_{i_1}^{(\alpha_1)}, u_2 \in S_{i_2}^{(\alpha_2)}$, it holds that $u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$. Here $\alpha_1 \cdot \alpha_2$ is the multiplication in R, and $i_1 + i_2$ is the integer addition.

The main difference between the cryptographic multilinear map defined in Def.2.2.1 and graded encoding system (Def.2.2.2) is that the encodings in graded encoding system is randomized. For the same message $\alpha \in R$, it can be encoded in many ways. The set $S_i^{(\alpha)}$ is a set of encodings of a ring elemnet α at level i.

Since the randomness of encodings, it is not trivial to check two encodings encoded the same ring element or not. From this reason, it needs additional procedure to decide equailty of encodings.

Another differenece is that intermediate level of encodings can be multiplied in contrast with only κ encodings can be coputed at the same time in cryptographic multilinear map.

2.3 Multilinear Map Procedures

In this sectios, we introduce multilinear map procedure defined in [GGH13] and [CLT13]. The goal of multilinear map is not to obtain a plaintext of encoding. It only needs to obtain the same values in the end. The zero-testing procedure enables to determine whether a top-level encoding is zero or not. This means that we can decide two top-level encodings are in the same set $S_\kappa^{(\alpha)}$ for some $\alpha \in R$ or not, by additive homomorphic property. Additional extraction procedure gives the same random string to encodings of the same plaintext.

Similar in public key homomorphic encryption scheme, it needs re-randomization procedure. However the users cannot encode plaintext $\alpha \in R$ directly, they just do several operations on random level-0 encodings without knowing underlying plaintexts. The number of multiplications can not exceed level-κ which is fixed in the beginning.

Instance Generation: $(\mathsf{params}, \boldsymbol{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. The randomized instance generation procedure takes as input the security parameter λ, the multilinearity level κ, and outputs the public parameters $(\mathsf{params}, \boldsymbol{p}_{zt})$, where params is a description of a κ-graded encoding system as above, and $\boldsymbol{p}_{zt}$ is a zero-test parameter.

Ring Sampler: $u \leftarrow \mathsf{samp}(\mathsf{params})$. The randomized sampling procedure takes as input the public parameters parmas and outputs a level-0 encoding $u \in S_0^{(\alpha)}$ for a nearly uniform $\alpha \in R$. Note that u does not need to be uniform in $S_0^{(\alpha)}$.

Encoding: $u' \leftarrow \mathsf{enc}(\mathsf{params}, u)$. The possibly randomized encoding procedure takes as input the public parameters params, and a level-0 encoding $u \in S_0^\alpha$ for some $\alpha \in R$, and outputs a level-1 encoding $u' \in S_1^{(\alpha)}$.

Re-Randomization: $\mathsf{reRand}(\mathsf{params}, i, u)$. The randomized rerandomization

procedure takes as input the public parameters params, a level $i \leq \kappa$, and a level-i encoding $u \in S_i^{\alpha}$ for some $\alpha \in R$, and outputs another level-i encoding $u' \in S_i^{(\alpha)}$ of the same α. It satisfies that for any $u_1, u_2 \in S_i^{(\alpha)}$, the output distributions of $\mathsf{reRand}(\mathsf{params}, i, u_1)$ and $\mathsf{reRand}(\mathsf{parmas}, i, u_2)$ are nearly the same.

Negation: $-u \leftarrow \mathsf{neg}(\mathsf{params}, u)$. the negation procedure takes as input the public parameters params, and a level-i encoding $u \in S_i^{(\alpha)}$ for some $\alpha \in R$, and outputs a level-i encoding $u' \in S_i^{(-\alpha)}$. We write u' as $-u$ as a shorthand for applying these procedure.

Addition: $u_1 + u_2 \leftarrow \mathsf{add}(\mathsf{params}, u_1, u_2)$. The addition procedure takes as input the public parameters parmas, two level-i encodings $u_1 \in S_i^{(\alpha_1)}, u_2 \in S_i^{(\alpha_2)}$ for some $\alpha_1, \alpha_2 \in R$, and outputs a level-i encoding $u' \in S_i^{(\alpha_1+\alpha_2)}$. We denote u' as $u_1 + u_2$.

Multiplication: $u' \leftarrow \mathsf{mul}(\mathsf{params}, u_1, u_2)$. The multiplication procedure takes as input the public parameters params, two encodings $u_1 \in S_i^{(\alpha_1)}, u_2 \in S_j^{(\alpha_2)}$ of some $\alpha_1, \alpha_2 \in R$ at levels i and j such that $i + j \leq \kappa$, and outputs a level-$(i + j)$ encoding $u' \in S_{i+j}^{(\alpha_1 \cdot \alpha_2)}$.

Zero-test: $\mathsf{isZero}(\mathsf{params}, u)$. The zero-testing procedure takes as input the public parameters params, and an encoding $u \in S_{\kappa}^{(\alpha)}$ of some $\alpha \in R$ at the maximum level κ, and outputs 1 if $\alpha = 0$, 0 otherwise, with negligible probability of error (over the choice of $u \in S_{\kappa}^{(\alpha)}$).

Extraction: $\mathsf{ext}(\mathsf{params}, \boldsymbol{p}_{zt}, u)$. The extraction procedure takes as input the public parameters params, the zero-test parameter $\boldsymbol{p}_{zt}$, and an encoding $u \in S_{\kappa}^{(\alpha)}$ of some $\alpha \in R$ at the maximum level κ, and outputs $s \in \{0,1\}^{\lambda}$ such that:

1. For an $\alpha \in R$ and $u_1, u_2 \in S_{\kappa}^{(\alpha)}$, $\mathsf{ext}(\mathsf{params}, \boldsymbol{p}_{zt}, u_1) = \mathsf{ext}(\mathsf{pp}, \boldsymbol{p}_{zt}, u_2)$.

2. The distribution $\{\mathsf{ext}(\mathsf{params}, \boldsymbol{p}_{zt}, u) : \alpha \leftarrow R, u \in S_{\kappa}^{(\alpha)}\}$ is nearly uniform over $\{0,1\}^{\lambda}$.

Note that only the same level encodings can be added and the multiplication outputs encoding of increased level. The zero-testing can be done only for top-level encodings.

2.4 Related Problems

We introduce some additional problems related to multilinear maps. When GGH13 was suggested, it looks hard that Subgroup Membership Problem (SubM) and Decisional Linear Problem in GGH13. However after a whilw, it has proven that they are not secure. When CLT13 and CLT15 was proposed, they also look forward to support that problems so they have more applications. Until now, there are no known reduction from the hardness of these problems to the MDDH problem.

Let $G = \mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$ and G_i be the subgroup of order g_i obtained by forcing the components of the other $\mathbb{Z}_{g_j}$'s to be zero. For index set $I \subseteq [n]$, we denote $G_I = \prod_{i \in I} G_i$. We let $\mathsf{enc}_1(\mathfrak{t})$ denote a properly generated level-1 encoding of $\mathfrak{t} \in G$. For integers $L, N > 0$, we let $\mathsf{Rk}_i(\mathbb{Z}_N^{L \times L})$ denote the set of $L \times L$ matrices over $\mathbb{Z}_N$ of rank i. If N is a product of primes, we define the rank of a matrix as the maximum of the ranks of the matrices obtained by reduction modulo all the prime divisors of N.

Definition 2.4.1. (The Subgroup Membership Problem) SubM is as follows. Given λ and κ, generate params and $\boldsymbol{p}_{zt}$ using $\mathsf{InstGen}$ and $\{\mathsf{enc}_1(\mathfrak{g}_i) : i \in [\ell]\}$ where the $\mathfrak{g}_i$'s are uniformly and independently sampled in a strict subgroup G_I of G, with ℓ sufficiently large so that the $\mathfrak{g}_i$'s generate G_I with overwhelming probability. Given params, $\boldsymbol{p}_{zt}$, $\{\mathsf{enc}_1(\mathfrak{g}_i) : i \in [\ell]\}$ and $u = \mathsf{enc}_1(\mathfrak{m})$, determine whether $\mathfrak{m}$ is sampled uniformly in G_I or in G.

Definition 2.4.2. (L-Decisional Linear Problem) L-DLIN is as follows. Given λ and κ, generate params and $\boldsymbol{p}_{zt}$ using $\mathsf{InstGen}$. Define $N = \prod_i g_i$. Given params and $\boldsymbol{p}_{zt}$, the goal is to distinguish between the distributions

$$\{(\mathsf{enc}_1(m_{ij}))_{i,j}\}_{(m_{ij})_{i,j} \leftarrow \mathsf{Rk}_{L-1}(\mathbb{Z}_N^{L \times L})} \quad \text{and} \quad \{(\mathsf{enc}_1(m'_{ij}))_{i,j}\}_{(m'_{ij})_{i,j} \leftarrow \mathsf{Rk}_L(\mathbb{Z}_N^{L \times L})}.$$

Graded External DDH Problem (GHDH) is defined on asymmetric multilinear maps.

Definition 2.4.3. (Graded External DDH Problem) GXDH is as follows. Given λ and κ, generate params and $\boldsymbol{p}_{zt}$ using $\mathsf{InstGen}$. Given params, $\boldsymbol{p}_{zt}$ and $\mathsf{enc_t(a)}, \mathsf{enc_t(b)}$ and $\mathsf{enc_t(c)}$ with $a, b \leftarrow G$ and for a given $t \in [\kappa]$, the goal is to decide whether $c = a \cdot b$ or c is uniformly and independently sampled in G.

Chapter 3

Break and Repair: Two Multilinear Maps over the Integers

In this section, we introduce two multilinear maps over the integers. The first one is suggested at CRYPTO 2013 by Coron, Lepoint and Tibouchi. It has similar features with homomorphic encryptions over the integers such as [vDGHV10] and [CCK+13]. Especially, the ciphertext of message $(m_1, \cdots, m_n) \in \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ is of the form

$$\mathsf{CRT}_{(q_0,p_1,\cdots,p_n)}(q_0, r_1 g_1 + m_1, \cdots, r_n g_n + m_n)$$

in [CCK+13], and a level-k encoding of $(m_1, \cdots, m_n) \in \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ is of the form

$$\mathsf{CRT}_{(p_1,\cdots,p_n)}((r_1 g_1 + m_1)/z^k, \cdots, (r_n g_n + m_n)/z^k)$$

in [CLT13]. The correctness of these two schemes hold as long as each $r_i g_i + m_i$ does not go over p_i, so both allow restricted number of multiplications ([CCK+13] can support unrestricted number of multiplications with

bootstrapping technique). The structure of encodings and addition, multiplication process are quite similar.

The difference is here. Homomorphic encryption allows to decrypt only with secret key. But the goal of multilinear map is not to recover a message, it needs to decide whether encoded values are the same or not without secret information. The zero-testing parameter is constructed to achieve this and the additional structure of z enables to decide it only when the level is κ.

However, this causes a weakness of the scheme. Cheon, Han, Lee, Ryu and Stehlé proposed a polynomial time attack, so called CHLRS attack [CHL$^+$15]. Using the structural feature of zero-testing, one can find all the secrets of CLT13 in polynomial time of security parameter.

After CHLRS attack, there are several attempts to make a scheme to be secure [GGHZ14, BWZ14]. However they were proven insecure soon [CGH$^+$15]. Around the same time, Coron, Lepoint and Tibouch proposed another fix of the scheme [CLT15] which is called CLT15. To thwart CHLRS approach, they modified the zero-testing to become a non-linear equation in secret parameters. Since CHLRS used the fact that the zero-testing value of encoding of zero is written as a linear combination of secrets, CLT15 looks secure when it was suggested.

We first introduce CLT13 multilinear map, and then explain how to analyze it. After that, we give a description of CLT15 construction and how to compensate the defect CLT13.

3.1 The CLT13 Multilinear Map and CHLRS Attack

3.1.1 The CLT13 Multilinear Map

We introduce Coron *et al.*'s first construction, CLT13 multilinear map. The scheme relies on the following parameters.

λ: the security parameter

κ: the multilinearity parameter

ρ: the bit length of the randomness used for encodings

α: the bit length of the message slots

η: the bit length of the secret primes p_i

n: the number of distinct secret primes

τ: the number of level-1 encodings of zero in public parameters

ℓ: the number of level-0 encodings in public parameters

ν: the bit length of the image of the multilinear map

β: the bit length of the entries of the zero-test matrix H

Coron *et al.* suggested to set the parameters so that the following conditions are met:

- $\rho = \Omega(\lambda)$: to avoid brute force attack (see also [LS14] for a constant factor improvement).
- $\alpha = \lambda$: so that the ring of messages $\mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$ does not contain a small subring $\mathbb{Z}_{g_i}$.

- $n = \Omega(\eta \cdot \lambda)$: to thwart lattice reduction attacks.
- $\ell \geq n \cdot \alpha + 2\lambda$: to be able to apply the leftover hash lemma from [CLT13, Le. 1].
- $\tau \geq n \cdot (\rho + \log_2(2n)) + 2\lambda$: to apply leftover hash lemma from [CLT13, Se. 4].
- $\beta = \Omega(\lambda)$: to avoid the so-called gcd attack.
- $\eta \geq \rho_\kappa + \alpha + 2\beta + \lambda + 8$, where ρ_κ is the maximum bit size of the random r_i's a level-κ encoding. When computing the product of κ level-1 encodings and an additional level-0 encoding, one obtains $\rho_\kappa = \kappa \cdot (2\alpha + 2\rho + \lambda + 2\log_2 n + 2) + \rho + \log_2 \ell + 1$.
- $\nu = \eta - \beta - \rho_f - \lambda - 3$: to ensure zero-test correctness.

Instance generation: $(\mathsf{params}, \boldsymbol{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. Set the scheme parameters as explained above. For $i \in [n]$, generate η-bit primes p_i, α-bit primes g_i, and compute $x_0 = \prod_{i \in [n]} p_i$. Sample $z \leftarrow \mathbb{Z}_{x_0}$. Let $\Pi = (\pi_{ij}) \in \mathbb{Z}^{n \times n}$ with $\pi_{ij} \leftarrow (n2^\rho, (n+1)2^\rho) \cap \mathbb{Z}$ if $i = j$, otherwise $\pi_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$. For $i \in [n]$, generate $\vec{r}_i \in \mathbb{Z}^n$ by choosing randomly and independently in the half-open parallelepiped spanned by the columns of the matrix Π and denote by r_{ij} the j-th component of $\vec{r}_i$. Generate $\boldsymbol{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$, $\boldsymbol{A} = (a_{ij}) \in \mathbb{Z}^{n \times \ell}$ such that $\boldsymbol{H}$ is invertible and $\|\boldsymbol{H}^T\|_\infty \leq 2^\beta$, $\|(\boldsymbol{H}^{-1})^T\|_\infty \leq 2^\beta$ and for $i \in [n]$,

$j \in [\ell]$, $a_{ij} \leftarrow [0, g_i)$. Then define:

$$\begin{aligned}
y &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i + 1}{z}\right), \text{ where } r_i \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z} \text{ for } i \in [n], \\
x_j &= \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij} g_i}{z}\right) \text{ for } j \in [\tau], \\
x'_j &= \mathsf{CRT}_{(p_i)}(x'_{ij}), \text{ where } x'_{ij} = r'_{ij} g_i + a_{ij} \\
&\quad \text{and } r'_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z} \text{ for } i \in [n], j \in [\ell], \\
(\boldsymbol{p}_{zt})_j &= \left[\sum_{i=1}^{n} \left[h_{ij} \cdot z^\kappa \cdot g_i^{-1}\right]_{p_i} \cdot \prod_{i' \neq i} p_{i'}\right]_{x_0} \text{ for } j \in [n].
\end{aligned}$$

Output $\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \nu, y, \{x_j\}, \{x'_j\}, \{\Pi_j\}, s)$ and $\boldsymbol{p}_{zt}$. Here s is a seed for a strong randomness extractor, which is used for an "**Extraction**" procedure.

Sampling level-zero encodings: $e \leftarrow \mathsf{samp}(\mathsf{params})$. For $1 \le j \le \ell$, sample $b_j \leftarrow \{0, 1\}$ and compute $e = [\sum_{j=1}^{\ell} b_j \cdot x'_j]_{x_0}$ Then e is an encoding of message $\boldsymbol{m}$ with the distribution of $\vec{m}$ is statistically close to uniform over the ring $\mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$.

Lemma 3.1.1. *[CLT13] Let $e \leftarrow \mathsf{samp}(\mathsf{params})$ and write $e \equiv r_i g_i + m \mod p_i$. Assume $\ell \ge n\alpha + 2\lambda$. The distribution of $(\mathsf{params}, \boldsymbol{m})$ is statistically close to the distribution of $(\mathsf{params}, \boldsymbol{m}')$ where $\boldsymbol{m}' \leftarrow \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$.*

Encodings at higher levels: $e_k \leftarrow \mathsf{enc}(\mathsf{params}, k, e)$. Note that y is a level-1 encoding of $\mathbf{1}$. Hence by multiplying with y, we can raise the level of encoding. Compute $e_k = [e \cdot y^k]_{x_0}$.

However, it is not secure when used in multipartie Diffie-Hellman key-exchange, since the private encoding e can be recovered directly by computing $e_k \cdot y^{-k} \mod x_0$. Therefore we need re-randomization procedure to make an encoding e_k which is not depend on e.

Re-randomizing level-1 encodings: $e' \leftarrow \mathsf{reRand}(\mathsf{params}, e)$. Let the matrix $\Pi = (\varpi_{i,j}) \in \mathbb{Z}^{n\times n}$ be a diagonally dominant matrix such that:

$$\begin{aligned} \varpi_{ij} \leftarrow &\quad (n2^\rho, (n+1)2^\rho) \cap \mathbb{Z}, \quad \text{if } i = j, \\ \leftarrow &\quad (-2^\rho, 2^\rho) \cap \mathbb{Z}, \quad \text{otherwise.} \end{aligned}$$

Using the matrix, we define level-one encoding of zero x_j's:

$$1 \le j \le \tau, x_j = \mathsf{CRT}_{(p_i)} \left(\frac{r_{ij} g_i}{z} \right),$$

where $(r_{1j}, \cdots, r_{nj})$ is randomly and independently generated in the half-open parallelipiped spanned by tge columns of Π.

For $j \in [\tau], i \in [n]$, sample $b_j \leftarrow \{0,1\}$, $b'_i \leftarrow [0, 2^\mu) \cap \mathbb{Z}$, with $\mu = \rho + \alpha + \lambda$. Return $e' = [e + \sum_{j\in[\tau]} b_j \cdot x_j + \sum_{i\in[n]} b'_i \cdot \Pi_i]_{x_0}$. Then by the leftover hash lemma over lattices, the distribution of e' is independent of e.

Lemma 3.1.2. *[CLT13] Let* $e \leftarrow \mathsf{samp}(\mathsf{params})$, $e_1 \leftarrow \mathsf{enc}(\mathsf{params}, 1, e)$, *and* $e'_1 \leftarrow \mathsf{reRand}(\mathsf{params}, 1, e_1)$. *Write* $e'_1 = \mathsf{CRT}_{(p_i)} \left(\frac{r_i g_i + m_i}{z} \right)$ *and* $\boldsymbol{r} = (r_1, \cdots, r_n)$. *Let the parameters be in as before, then the distribution of* $(\mathsf{params}, \boldsymbol{r})$ *is statistically close to the distribution of* $(\mathsf{params}, \boldsymbol{r}')$ *where* $\boldsymbol{r}'$ *is randomly generated in the half-open parallelepiped spanned by the column vectors of* $2^\mu \Pi$.

Adding and multiplying encodings: $\mathsf{Add}(e_1, e_2) = [e_1 + e_2]_{x_0}$ and $\mathsf{Mul}(e_1, e_2) = [e_1 \cdot e_2]_{x_0}$. The correctness holds as long as the numerator does not go over p_i.

Zero-testing: $\mathsf{isZero}(\mathsf{params}, \boldsymbol{p}_{zt}, e_\kappa) =^? 0/1$. Given a level-$\kappa$ encoding e, return 1 if $\|[\boldsymbol{p}_{zt} \cdot e]_{x_0}\|_\infty < x_0 \cdot 2^{-\nu}$, and return 0 otherwise.

Extraction: $sk \leftarrow \mathsf{ext}(\mathsf{params}, \boldsymbol{p}_{zt}, e_\kappa)$. To extract a random $(\boldsymbol{p}_{zt}, c)$. Given a level-κ encoding e, Compute $MSB_\nu([\boldsymbol{p}_{zt} \cdot e]_{x_0})$.

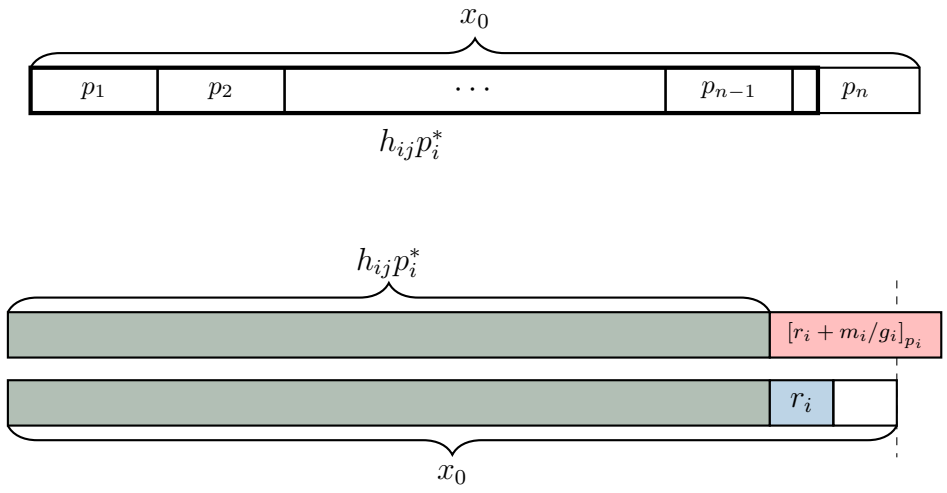

Figure 3.1: Zero-testing of CLT13

Now we examine the zero-testing procedure. Since the multiplication of the zero-testing vector and encoding is defined modulo x_0, the CRT representation is convenient. Using the equation 2.1.1, we can rewrite zero-testing vector as follow:

$$\begin{aligned}(\boldsymbol{p}_{zt})_j &= \sum_{1\le i\le n}\left[\frac{h_{ij}z^\kappa}{g_i}\right]_{p_i}\cdot p_i^* \mod x_0 \\ &= \mathsf{CRT}\left(\frac{h_{ij}z^\kappa}{g_i}p_i^*\right)\end{aligned}$$

Then, for a top-level encoding $e = \mathsf{CRT}_{(p_i)}((r_i + m_i \cdot g_i^{-1}) \cdot g_i/z^\kappa)$,

$$\begin{aligned}(\boldsymbol{p}_{zt}\cdot e)_j \mod x_0 &= \mathsf{CRT}_{(p_i)}(h_{ij}(r_i + m_i\cdot g_i^{-1})p_i^*) \\ &= \sum_{1\le i\le n} h_{ij}[r_i + m_i\cdot g_i^{-1}]_{p_i}p_i^* \mod x_0\end{aligned}$$

Hence, if $m_i = 0, 1 \leq i \leq n$, then the following holds over the integers:

$$\begin{aligned}(\boldsymbol{p}_{zt} \cdot e)_j \mod x_0 &= \mathsf{CRT}_{(p_i)}(h_{ij} r_i p_i^*) \\ &= \sum_{1 \leq i \leq n} h_{ij} r_i p_i^* \qquad\qquad < x_0 \cdot 2^{-\nu},\end{aligned}$$

so $\|\boldsymbol{p}_{zt} \cdot e \bmod x_0\|_\infty < x_0 \cdot 2^{-\nu}$.When there is an i with $m_i \neq 0$, by the construction of matrix $\boldsymbol{H}$, one can show that $\|\boldsymbol{p}_{zt} \cdot e \bmod x_0\|_\infty \geq x_0 \cdot 2^{-\nu+2}$, and so the correctness of zero-testing is obtained.

Optimizations. Coron *et al.* noticed that the size of public parameters is too large. For example, the size of public parameter is larger than 1 TB when $\lambda = 80$. So they suggested three heuristic optimizations.

1. Non-uniform sampling: take $\ell = 2\lambda$ and publish small number of x'_j, level-0 encodings of random messages.

2. Quadratic re-randomization: store $\lfloor \sqrt{n} \rfloor$ random encodings of level-0 and $\lfloor \sqrt{n} \rfloor$ level-1 encodings of zero. Combine pairwise to generate n level-1 encodings of zero.

3. Only 1 zero-testing parameter: use single integer $\boldsymbol{p}_{zt}$.

These optimizations make sampling is not uniform, and randomization becomes heuristic only. In the case of zero-testing, an encoding of zero gives a small value when multiplied with zero-testing integer, but the converse does not hold anymore.

3.1.2 Zeroizing Attacks on CLT13

In this section, we introduce how to cryptanalyze CLT13 multilinear map. In [CLT13], it was claimed that CLT13 is robust against a zeroizing attack in [GGH13]. The idea of zeroizing attack in [GGH13] is that to compute

many zero-testing values of encodings of zero and make a non-trivial ideal generated by a secret of zero-testing parameter. However, it becomes a trivial ideal $\mathbb{Z}$ when the attack is employed to CLT13. Hence, CLT13 supports the Graded Decisional Diffie-Helman assumption (GDDH), subgroup membership (SubM), and decisional linear (DLIN) problems are hard in it, while GGH13 supports onlythe GDDH.

The attack is proposed by Cheon, Han, Lee, Ryu, and Stehlé, called CHLRS attack [CHL$^+$15]. It uses many low-level encodings of zero, and finds all secrets in polynomial time of security parameter. As in the zeroizing attack of GGH13, the attack utilizes public low level encodings of zero, which allows an encoding to be generated without the secret values being known. The core of the attack is to compute several zero-testing values related to one another. Then, one can construct a matrix, the eigenvalues of which consist of the CRT component of x, which is $x \mod p_i$ for some encoding x, where $p_1, \cdots, p_n$ are secret values of the scheme. Then, it reveals all the secrets of the scheme. However, it is not adapted when no low-level encodings of zero are provided such as in iO application.

Shortly after that, Coron *et al.* suggested an extension of CHLRS attack [CGH$^+$15]. It is applied when orthogonal encodings are provided, which are well-separated sets of encodings which can make top-level encodings of zero. It can be applied to a matrix variant of GGH13, and some variants of CLT13 [BWZ14, GGHZ14].

Now, we explain the attack procedure. Let a be a level-s encoding, b be a level-t encoding and$c$ be a level-$(\kappa - s - t)$ encoding of zero. Then they can

be written as follows:

$$\begin{aligned} a &= \mathsf{CRT}_{(p_i)}\left(\frac{a_i}{z^s}\right), \\ b &= \mathsf{CRT}_{(p_i)}\left(\frac{b_i}{z^t}\right), \\ c &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i}{z^{\kappa-s-t}}\right), \end{aligned}$$

and the product of these three encoding is

$$abc \bmod x_0 = \mathsf{CRT}_{(p_i)}\left(a_i b_i r_i \frac{g_i}{z^\kappa}\right).$$

Hence the zero-testing value of this product gives the below equation which holds over the integers,

$$\boldsymbol{p}_{zt} \cdot abc \bmod x_0 = \sum_{1 \le i \le n} a_i b_i r_i h_i p_i^*,$$

and it can be expressed using the product of matrices.

$$\begin{pmatrix} a_1 & \dots & a_n \end{pmatrix} \begin{pmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{pmatrix} \begin{pmatrix} h_1 p_1^* & & \\ & \ddots & \\ & & h_n p_n^* \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

The point is that $h_i p_i^*$'s are independent of encodings a, b, and c. Hence by fixing c and varying a and b, we can construct an n-dimensional square matrix which can be expressed as a product of matrices.

Suppose we have three sets of encodings, denoted by $A = \{a_1, \cdots, a_n\}, B = \{b_1, \cdots, b_n\}$, and $C = \{c_0, c_1\}$. Each set has encodings of the same level such that the product of any $a_i, b_j (i, j \in [n])$, and $c_\sigma (\sigma \in \{0, 1\})$ is a level-κ encoding. We also assume that c_σ are encodings of zero. Then we can write

each encodings as follows:

$$\begin{aligned} a_j = &\quad \mathsf{CRT}_{(p_i)}\left(\frac{a_{ij}}{z^s}\right), \quad 1 \le j \le n, \\ b_j = &\quad \mathsf{CRT}_{(p_i)}\left(\frac{b_{ij}}{z^t}\right), \quad 1 \le j \le n, \\ c_\sigma = &\quad \mathsf{CRT}_{(p_i)}\left(\frac{r_{i\sigma} g_i}{z^{\kappa-s-t}}\right), \quad \sigma \in \{0,1\}, \end{aligned}$$

and the zero-testing value of $a_i b_j c_0, 1 \le i, j \le n$ gives the following equation:

$$\begin{aligned} \boldsymbol{W}_0 &= \begin{pmatrix} & a_{i,j} & \\ & & \end{pmatrix}^T \begin{pmatrix} r_{10} & & \\ & \ddots & \\ & & r_{n0} \end{pmatrix} \begin{pmatrix} h_1 p_1^* & & \\ & \ddots & \\ & & h_n p_n^* \end{pmatrix} \begin{pmatrix} & b_{i,j} & \\ & & \end{pmatrix} \\ &= \quad \boldsymbol{A}^T \qquad\qquad \boldsymbol{R_0} \qquad\qquad \boldsymbol{P} \qquad\qquad \boldsymbol{B} \end{aligned} \tag{3.1.1}$$

Then by repeating the same procedure using c_1 instead of c_0, we may have $\boldsymbol{W}_1 = \boldsymbol{A}^T \cdot \boldsymbol{R}_1 \cdot \boldsymbol{P} \cdot \boldsymbol{B}$. Then we may invert the matrix $\boldsymbol{W}_0$ over $\mathbb{Q}$ (whp) and compute below

$$\begin{aligned} \boldsymbol{W}_0^{-1} \cdot \boldsymbol{W}_1 &= \boldsymbol{B}^{-1} \cdot \boldsymbol{R}_0^{-1} \cdot \boldsymbol{R}_1 \cdot \boldsymbol{B} \\ &= \boldsymbol{B}^{-1} \cdot \mathsf{diag}(r_{11}/r_{10}, \cdots, r_{n1}/r_{n0}) \cdot \boldsymbol{B}, \end{aligned}$$

since diagonal matrices are commutative. By computing eigenvalues of $\boldsymbol{W}_0^{-1} \cdot \boldsymbol{W}_1$, we obtain relatively prime integers u_1 and u_0 satisfying $u_1/u_0 = r_{i1}/r_{i0}$ for some i. Note that $u_1 c_0 - u_0 c_1 \equiv_{p_i} (u_1 r_{i0} g_i - u_0 r_{i1} g_i)/z^{\kappa-s-t} \equiv_{p_i} 0$. Therefore computing $\gcd(x_0, u_1 c_0 - u_0 c_1)$ gives p_i and this reveals all the secrets immediately.

It needs sufficiently many low-level random encodings and only two low-level encodings of zero. As pointed in [CHL+15], the conditions are achieved when it is used in one round multi-party Diffie-Hellman. Note that one of the set A and B can be substituted as $\{(\boldsymbol{p}_{zt})_j\}$.

This attack heavily relies on the accessibility of low-level encodings of zero. What if we use a non-zero encoding c_σ instead of encoding of zero? As in previous, we compute the zero-testing values of $a_i b_j c_\sigma, 1 \leq i, j \leq n$. Since it is not an encoding of zero, the equation (3.1.1) holds in $\bmod x_0$ not over the integers. So it can be rewritten as

$$\boldsymbol{W}_0 \;=\; c_0 \boldsymbol{p}_{zt} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} b_1 & \dots & b_n \end{pmatrix} \mod x_0.$$

Therefore it is not a full rank matrix in $\bmod x_0$ and it looks hard to obtain any information about secrets.

3.2 The CLT15 Multilinear Map

In this section we introduce the CLT15 multilinear map. As seen 3.1.2, a weak point of CLT13 is that the zero-testing value of an encoding of zero, can be decomposed as a product of matrices over the integers. To overcome this, CLT15 added a new type noise, in order to the zero-testing value can be expressed an affine matrix equation only.

CLT15 retained the structure of encodings from CLT13. The main difference is zero-testing procedure, it is conducted in some independent modulus N. By using the independent modulus N and keeping x_0 secret, two kinds of new noises are added when adapt CHLRS attack. So it looks hard to apply the attack and secure. The optimized implementation of CLT15 is comparable to CLT13 in the aspect of timing and the size of public parameters.

The scheme relies on the following parameters.

λ: the security parameter

κ: the multilinearity parameter, i.e., the proposed map is κ- linear

ρ: the bit length of the initial noise usedfor encodings

α: the bit length of the primes g_i

η: the bit length of the secret primes p_i

n: the number of distinct secret primes

γ: the bit length of encodings $(= n\eta)$

τ:the number of level-1 encodings of zero in public parameters

ℓ: the number of level-0 encodings in public parameters

ν: the bit length of the image of the multilinear map

β: the bit length of the entries of the zero-test matrix H

Coron *et al.* suggested setting the parameters according to the following conditions.

- $\rho = \Omega(\lambda)$: to avoid a brute force attack on the noise
- $\alpha = \lambda$: to prevent a situation wherethe order of message ring $\mathbb{Z}_{g_1} \times \ldots \times \mathbb{Z}_{g_n}$ has a small prime factor.
- $n = \Omega(\eta\lambda)$: to thwart lattice reduction attacks
- $\ell \geq n\alpha + 2\lambda$: to apply the leftover hash lemma from [CLT15]
- $\tau \geq n(\rho + \log_2(2n)) + 2\lambda$: to apply the leftover hash lemma from [CLT15]
- $\beta = 3\lambda$: as a conservative security precaution
- $\eta \geq \rho_\kappa + 2\alpha + 2\beta + \lambda + 8$, where ρ_κ is the maximum bit size of the noise r_i of a level-κ encoding. When computing the product of κ level-1 encodings and an additional level-0 encoding, one obtains $\rho_\kappa = \kappa(2\alpha + 2\rho + \lambda + 2\log_2 n + 3) + \rho + \log_2 \ell + 1$
- $\nu = \eta - \beta - \rho_f - \lambda - 3$: to ensure correctness of zero-testing.

The constraints are the same as in [CLT13]; the condition that differs is β.

Instance generation: $(\mathsf{params}, \boldsymbol{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$. Set the scheme parameters as explained above. For $1 \leq i \leq n$, generate η-bit odd primes p_i and α-bit primes g_i, and compute $x_0 = \prod_{i=1}^{n} p_i$. Generate a random prime integer N of size $\gamma + 2\eta + 1$ bits. Using LLL algorithms in dimension 2, special pairs of nonzero integers $(\alpha_i, \beta_i)_{i=1}^n$ are chosen to satisfy $|\alpha_i| < 2^{\eta-1}$, $|\beta_i| < 2^{2-\eta} \cdot N$, $\beta_i \equiv \alpha_i u_i' p_i^{-1} \pmod N$, where $u_i' = \left[\frac{g_i}{z^\kappa}(p_i{}^*)^{-1}\right]_{p_i} \cdot p_i{}^*$. Finally, generate $\vec{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$ such that $\vec{H}$ is invertible and $\|\vec{H}^T\|_\infty \leq 2^\beta$,

$\|(\vec{H}^{-1})^T\|_\infty \leq 2^\beta$ and for $1 \leq i \leq n$, $1 \leq j \leq \ell$, $m_{ij} \leftarrow [0, g_i) \cap \mathbb{Z}$. Then, define

$$\begin{aligned}
y &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i + 1}{z}\right), \\
x_j &= \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij} g_i}{z}\right), \text{ for } 1 \leq j \leq \tau, \\
x'_j &= \mathsf{CRT}_{(p_i)}(r'_{ij} g_i + m_{ij}) \text{ for } 1 \leq j \leq \ell, \\
X_j^{(t)} &= \mathsf{CRT}_{(p_i)}\left(\frac{r_{ij}^{(t)} g_i}{z^t}\right) + q_j^{(t)} x_0 \text{ for } 0 \leq j \leq \gamma + \lfloor \log_2 \ell \rfloor, 1 \leq t \leq \kappa, \\
\Pi_j &= \sum_{i=1}^{n} \varpi_{ij} g_i \left[z^{-1} \left(\frac{x_0}{p_i}\right)^{-1} \right]_{p_i} \frac{x_0}{p_i} + \varpi_{n+1,j} x_0 \text{ for } 1 \leq j \leq n+1, \text{and} \\
(\boldsymbol{p}_{zt})_j &= \sum_{i=1}^{n} h_{ij} \alpha_i p_i^{-1} \pmod N \text{ for } 1 \leq j \leq n,
\end{aligned}$$

where $r_i, r'_{ij}, r_{ij}^{(t)} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$, $q_j^{(t)} \leftarrow [2^{\gamma+j-1}/x_0, 2^{\gamma+j}/x_0) \cap \mathbb{Z}$, and $\varpi_{ij} \leftarrow (-2^\rho, 2^\rho) \cap \mathbb{Z}$ if $i \neq j$, $\varpi_{ii} \leftarrow ((n+1)2^\rho, (n+2)2^\rho) \cap \mathbb{Z}$. Then, output

$$\mathsf{params} = (n, \eta, \alpha, \rho, \beta, \tau, \ell, \mu, y, \{x_j\}_{j=1}^{\tau}, \{x'_j\}_{j=1}^{\ell}, \{X_i^{(j)}\}, \{\Pi_j\}_{j=1}^{n+1}, s) \text{ and } \boldsymbol{p}_{zt}.$$

In this paper, we used only one zero-testing parameter. Hence, hereafter, we use a notation $\boldsymbol{p}_{zt} = \sum_{i=1}^{n} h_i \alpha_i p_i^{-1} \pmod N$ instead of a vector $(\boldsymbol{p}_{zt})_j$, if no confusion results.

Sampling level-0 encodings: $c \leftarrow \mathsf{samp}(\mathsf{params})$. Since the user does not know p_i, one cannot encode a vector $\vec{m} \in \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$.Hence, CLT15 provides level zero encodings $\{x'_j\}$ for sampling. A level zero encoding c is computed as a random subset sum of $\{x'_j\}$. Namely, for $1 \leq j \leq \ell$, sample $b_j \leftarrow \{0, 1\}$ and compute $c = \sum_{j=1}^{\ell} b_j \cdot x'_j$.

Encodings at higher levels: $c_k \leftarrow \mathsf{enc}(\mathsf{params}, k, c)$. Given a level-0 encoding c, to obtain alevel-1 encoding c_1 with the same plaintext as c, compute$c_1 = c \cdot y$. Since x_0 is not given, a ladder of level-1 encodings of zero $X_j^{(1)}$ is provided. Then, iteratively reduce the size of c_1 to that of $X_0^{(1)}$.

In general, to obtain a level-k encoding, compute $c_k = c \cdot y^k$ and reduce the size of c_k after each multiplication by y using ladders $\{X_j^{(i)}\}_{j=0}^{\gamma+\lfloor \log_2 \ell \rfloor}$ for levels $i = 1, \cdots, k$.

Re-randomizing level-1 encodings: $c' \leftarrow \mathsf{reRand}(\mathsf{params}, c)$. For $1 \leq j \leq \tau, 1 \leq i \leq n+1$, sample $b_j \leftarrow \{0,1\}$, $b'_i \leftarrow [0, 2^\mu) \cap \mathbb{Z}$, with $\mu = \rho + \alpha + \lambda$. Return $c' = c + \sum_{j=1}^{\tau} b_j \cdot x_j + \sum_{i=1}^{n+1} b'_i \cdot \Pi_i$.This procedure can be adapted to higher levels $1 < k \leq \kappa$ by publishing appropriate quantities in params.

Adding and multiplying encodings: For two encodings, the addition and multiplication are performed in $\mathbb{Z}$. After the arithmetic, reduce the size to that of $2x_0$ using the ladder.

Zero-testing: $\mathsf{isZero}(\mathsf{params}, \boldsymbol{p}_{zt}, e) \stackrel{?}{=} 0/1$. Given a level-$\kappa$ encoding e, return 1 if $\|\boldsymbol{p}_{zt} \cdot e \pmod N\|_\infty < N \cdot 2^{-\nu}$, and 0 otherwise.

Extraction: $sk \leftarrow \mathsf{ext}(\mathsf{params}, \boldsymbol{p}_{zt}, e)$. Given a level-$\kappa$ encoding e, compute the most significant ν bits of $[\boldsymbol{p}_{zt} \cdot e]_N$.

Now we take a closer look at zero-testing procedure. For the sake of simplicity, we denote $\boldsymbol{p}_{zt}$ as a specific $(\boldsymbol{p}_{zt})_j$ for some j and omit the index j. First, we remind that the zero-testing procedure of CLT13. For a level-κ encoding $e = \mathsf{CRT}_{(p_i)}([r_i + m_i/g_i]_{p_i} \frac{g_i}{z^\kappa})$, the zero-testing value is

$$\boldsymbol{p}_{zt} \cdot e \bmod x_0 = \sum_{1 \leq i \leq n} h_i [r_i + m_i/g_i]_{p_i} p_i^* \bmod x_0.$$

Since p_i^* is small as p_i than x_0, $[r_i + m_i/g_i]_{p_i} p_i^* \approx x_0$. Therefore $h_i[r_i + m_i/g_i]_{p_i} p_i^*$ goes over x_0 if $m_i \neq 0$, and $h_i r_i p_i^* \ll x_0$. Hence one can decide whether it is an encoding of zero.

As we describe before, zero-testing of CLT15 executed in another modulus N, however the principle is almost same. Let e be a level-κ encoding, then $e = \mathsf{CRT}_{(p_i)}([r_i + m_i/g_i]_{p_i} \frac{g_i}{z^\kappa}) + a' x_0$. To compare with encoding of CLT13,

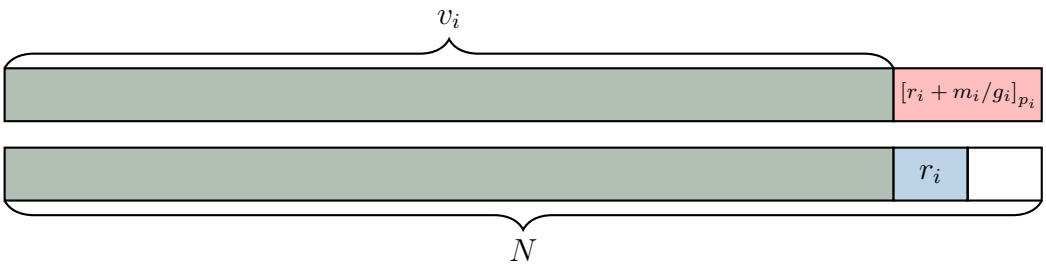

Figure 3.2: Zero-testing of CLT15

it has additional term $a'x_0$ since we can not reduce it using x_0. We can rewrite it as $e = \sum[r_i + m_i/g_i]_{p_i}u_i + ax_0$, where $u_i = [g_i/z^\kappa(p_i^*)^{-1}]_{p_i}p_i^*$ which is independent of encoding. Then the zero-testing value can be written

$$\boldsymbol{p}_{zt} \cdot e \bmod N = \sum_{1 \le i \le n} [r_i + m_i/g_i]_{p_i} v_i + av_0 \bmod N,$$

where $v_i = \boldsymbol{p}_{zt} \cdot u_i \bmod N$ and $v_0 = \boldsymbol{p}_{zt} \cdot x_0 \bmod N$. The zero-testing vector is constructed to satisfy v_i is small as p_i than N. Hence as similar in CLT13, $[r_i + m_i/g_i]_{p_i}v_i \approx N$ if $m_i \neq 0$, and $r_i v_i \ll N$.

However, in this case, we need to consider the additional term, av_0. To identify zero, the size of av_0 must be much smaller than N, hence the size of a must be controlled. The following lemma gives more detailed explanation on this.

Lemma 3.2.1 (Zero testing lemma). *Let e be a level-κ encoding of zero with $e = \sum_{i=1}^n r_i u_i' + ax_0, (r_1, \cdots, r_n, a \in \mathbb{Z})$. Then,*

$$\boldsymbol{p}_{zt} \cdot e \ mod\ N = \sum_{i=1}^{n} r_i v_i + av_0,$$

holds over the integers, if $|a| < 2^{2\eta-\beta-\log_2 n-1}$ and $|r_i| < 2^{\eta-\beta-\log_2 n-6}$ for $1 \le i \le n$.

Proof 1. *By the construction of the zero-testing element, we have* $\boldsymbol{p}_{zt} \cdot e \equiv \sum_{i=1}^{n} r_i v_i + a v_0 \pmod{N}$. *It is sufficient to show that the right hand side is smaller than* $N/2$. *For* $1 \leq i \leq n$,

$$v_i \equiv \sum_{j=1}^{n} h_j \alpha_j p_j^{-1} u_i' \equiv h_i \beta_i + \sum_{j \neq i} h_j \alpha_j \left[\frac{g_i}{z^\kappa} (p_i^*)^{-1} \right]_{p_i} \frac{x_0}{p_i p_j} \mod N,$$

and therefore, $|v_i| < 2^{\gamma+\eta+\beta+4}$ *for* $1 \leq i \leq n$. *Moreover,* $v_0 = \sum_{j=1}^{n} h_j \alpha_j \frac{x_0}{p_j}$ *and* $|v_0| < n 2^{\gamma+\beta-1}$. □

The size of a is deeply related to the size of e. When e is large, the size of a is close to e/x_0. So we can control the size of a by reduce the size of encoding e. From this reason, size-reduction must be done before performing zero-testing. To reduce the size of encoding while hiding x_0, CLT15 publishes encodings of zero of increasing size. It is inspired from [vDGHV10] and called a ladder.

More precisely, a ladder $(X_j^{(k)})_{0 \leq j \leq \gamma'}$ are encodings of zero of each level $k \leq \kappa$,

$$X_0^{(k)} < X_1^{(k)} < \cdots < X_{\gamma'}^{(k)}, \qquad X_j^{(k)} \approx 2^j x_0,$$

where $\gamma' = \gamma + \lfloor \log_2 \ell \rfloor$. Using this ladder, we can reduce the size of encoding e down to the size of $X_0^{(k)}$ without altering the encoded value.

CHLRS attack to CLT15 multilinear map

When CLT15 was suggested, it looks hard to apply CHLRS attack. We first explain direct adaptations of CHLRS attack to CLT15 and why they did not succeed. As in 3.1.2, suppose we have three encodings a, b and c where abc is a level-κ encoding. We also assume that c is an encoding of zero. Then we

can write each encodings as follows:

$$\begin{aligned} a &= \mathsf{CRT}_{(p_i)}\left(\frac{a_i}{z^s}\right), \\ b &= \mathsf{CRT}_{(p_i)}\left(\frac{b_i}{z^t}\right), \\ c &= \mathsf{CRT}_{(p_i)}\left(\frac{r_i g_i}{z^{\kappa-s-t}}\right). \end{aligned}$$

In CLT15, it is neccessary to reduce the size of encoding before conducting a zero-testing. We define e' be a size-reduced encoding of abc using ladders, then

$$\begin{aligned} e' &= abc - \sum_{1\le j\le \gamma'} b_j X_j^{(\kappa)} \\ &= \sum_{1\le i\le n} (a_i b_i r_i + s_i)\cdot u_i' + a x_0, \end{aligned}$$

where $b_j \in \{0,1\}$ and s_i, a are some integers. Hence the zero-testing value of e' can be written as

$$\boldsymbol{p}_{zt}\cdot e' \mod N = \sum_i (a_i b_i r_i + s_i) v_i + a v_0. \tag{3.2.1}$$

As similar in equation (3.1.1), we can construct a matrix equation using different choice of a and b, then

$$\boldsymbol{W_c} = \boldsymbol{X}\times\boldsymbol{C}\times\boldsymbol{Y} + \boldsymbol{S} + \boldsymbol{A}\cdot v_0.$$

Since it has two additional unknown matrices S and A, it looks hard to find secret informations as before. Moreover, additional problems are expected to being secure such as DLIN and SubM problems.

Chapter 4

Main Attack

In this section, we describe how to cryptanalyze CLT15 multilinear map. To thwart CHLRS approach, additional noises are added in CLT15 multilinear map by executing zero-testing in independent modulus. This causes two kinds of noises: one is related to a secret modulus x_0 and the other is caused from ladders. Hence it looks robust to CHLRS approach.

Ironically, the starting point of our attack is there. From the equation obtained from CHLRS approach to CLT15, we deduce integer equations from zero-testing values of encodings of zero. To obtain a matrix equation as in CLT13, we remove the effect of noises in two-steps. The first step is to eliminate noise which comes from ladders. From this step, we gain equations over the integers with one more variable to compared with CLT13. The second step is to build an $(n+1)$-dimension matrix equation. By enlarging the dimension of matrix equation, we obtain matrix equation consists of multiplication of matrices only.

4.1 Computing ϕ-values

Let us remind the zero-testing value of encoding in CLT15. If e is an encoding of zero, then it can be written as $e = \sum r_i u_i + a x_0$. Note that the size of r_i and u_i are smaller than p_i and x_0, respectively. Hence the former term of equation of e cannot be larger than $n2^{\eta+\gamma}$. Therefore the size of a is heavily depends on the size of the encoding. For exampls, the size of e is about $2^{2\gamma}$ and so the size of a is close to 2^{γ} when e is obtained from the multiplication of two low-level encodings.

As we explained in Lemma 3.2.1, the zero-testing equation holds over the integers only when the size of a is small. Suppose e is an encoding of zero of large size. To obtain a meaningful result from zero-testing, we need to reduce its size using ladders. Then the size-reduced encoding e' can be written as

$$\begin{aligned} e' &= e - \sum_{j=0}^{\gamma'} b_j X_j^{(\kappa)} \\ &= \sum_{i=1}^{n} \Big(r_i - \sum_{j=0}^{\gamma'} b_j r_{ij}^{(\kappa)} \Big) \cdot u_i + \Big(a - \sum_{j=0}^{\gamma'} b_j a_j^{(\kappa)} \Big) \cdot x_0, \end{aligned}$$

where $b_j \in \{0,1\}$ and $X_j^{(\kappa)} = \sum_i r_{ij}^{(\kappa)} u_i +_j^{(\kappa)} x_0$, a level-$\kappa$ ladder. Now e' satisfied the conditions in Lemma 3.2.1 and so we obtain the following equations:

$$\begin{aligned} \boldsymbol{p}_{zt} \cdot e' \mod N &= \boldsymbol{p}_{zt} \cdot \Big(\sum_{i=1}^{n} \Big(r_i - \sum_{j=0}^{\gamma'} b_j r_{ij}^{(\kappa)} \Big) \cdot u_i + \Big(a - \sum_{j=0}^{\gamma'} b_j a_j^{(\kappa)} \Big) \cdot x_0 \Big) \mod N \\ &= \sum_{i=1}^{n} \Big(r_i - \sum_{j=0}^{\gamma'} b_j r_{ij}^{(\kappa)} \Big) \cdot v_i + \Big(a - \sum_{j=0}^{\gamma'} b_j a_j^{(\kappa)} \Big) \cdot v_0 \\ &= \sum_{i=1}^{n} r_i v_i + a v_0 - \sum_{j=0}^{\gamma'} b_j \cdot \Big(\sum_{i=1}^{n} r_{ij}^{(\kappa)} v_i + a_j^{(\kappa)} v_0 \Big) \end{aligned}$$

Unlike the equation (3.2.1), we write the zero-testing value explicitly using publicly computable b_j's. If we have $\sum_{i=1}^{n} r_i v_i + a v_0$, a new noise s_i in the

equation (3.2.1) can be removed. To obtain the value, it needs to compute each $\sum_{i=1}^{n} r_{ij}^{(\kappa)} v_i + a_j^{(\kappa)} v_0, 0 \leq j \leq \gamma'$.

We define a function ϕ which maps an encoding $e = \sum_{i=1}^{n} r_i u_i + a x_0$ to an integer $\sum_{i=1}^{n} r_i v_i + a v_0$. Then the above equation can be rewritten as

$$\boldsymbol{p}_{zt} \cdot e' \mod N = \phi(e) - \sum b_j \cdot \phi(X_j^{(\kappa)}). \tag{4.1.1}$$

Hence if we have $\phi(X_j^{(\kappa)})$'s, then we can obtain $\phi(e)$ for large encoding.

At first, we give an exact description of ϕ. It is a map from the integer to integer such that

$$\begin{aligned} \phi: \ \mathbb{Z} &\to \mathbb{Z} \\ x &\mapsto \sum_{i=1}^{n} \left[x \cdot \frac{z^\kappa}{g_i} \right]_{p_i} v_i + \frac{x - \sum_{i=1}^{n} [x \cdot \frac{z^\kappa}{g_i}]_{p_i} u_i}{x_0} v_0, \end{aligned}$$

where $v_i = \boldsymbol{p}_{zt} \cdot u_i \mod N (1 \leq i \leq n)$ and $v_0 = \boldsymbol{p}_{zt} \cdot x_0 \mod N$ as before. Note that, $x \equiv \sum_{i=1}^{n} [x \cdot \frac{z^\kappa}{g_i}]_{p_i} u_i \pmod{p_j}$ for $1 \leq j \leq n$. Hence x_0 divides $x - \sum_{i=1}^{n} [x \cdot \frac{z^\kappa}{g_i}]_{p_i} u_i$ and the function is well-defined. One can easily check that $\phi(e) = \sum r_i v_i + a v_0$ for an encoding of zero $e = \sum r_i u_i + a x_0$.

When the encoding satisfies conditions in Lemma 3.2.1,ϕ-value equals the zero-testing value. However they only equal in modulo N when the size of encoding is large. So the first goal of our attack is to obtain ϕ-values of encoding xcy where it is a level-κ encoding of zero as in CHLRS attack. That means we remove one noise s_i in the equation (3.2.1).

Now we introduce a nice property of ϕ which says it is additive homomorphic.

Proposition 4.1.1. Let e_1 and e_2 be level-κ encodings of zero such that $e_j \equiv \frac{r_{ij} g_i}{z^\kappa} \pmod{p_i}$ and $|r_{ij}| < p_i/2$ for all $1 \leq i \leq n$, $j = 1, 2$. Suppose $|r_{i1} + r_{i2}| < p_i/2$ for all $1 \leq i \leq n$, then

$$\phi(e_1 + e_2) = \phi(e_1) + \phi(e_2).$$

Proof 2. *We note that* $e_j \cdot \frac{g_i}{z^\kappa} \mod p_i = r_{ij}$. *Hence* $\phi(e_j) = \sum_{i=1}^n r_{ij} v_i + a_j v_0$ *for some integer* a_j, *and* $(e_1 + e_2) \cdot \frac{g_i}{z^\kappa} \mod p_i = r_{i1} + r_{i2} \mod p_i$. *From the condition* $|r_{i1} + r_{i2}| < p_i/2$, *modulus reduction does not occur in the right hand side. Therefore the third equality holds in the following equations:*

$$\begin{aligned}
\phi(e_1) + \phi(e_2) &= \Big(\sum_{i=1}^n r_{i1} v_i + a_1 v_0\Big) + \Big(\sum_{i=1}^n r_{i2} v_i + a_2 v_0\Big) \\
&= \sum_{i=1}^n (r_{i1} + r_{i2}) \cdot v_i + (a_1 + a_2) \cdot v_0 \\
&= \phi\Big((r_{i1} + r_{i2}) \cdot u_i + (a_1 + a_2) \cdot x_0\Big) = \phi(e_1 + e_2).
\end{aligned}$$

□

Note that the condition on r_{ij} also needs to correctness of the multilinear map. Hence we may regard ϕ as an additive homomorphism.

Consider the zero-testing of encoding xcy. At first we multiply x with c and reduce its size using level-$(s+t)$ ladder. After that y is multiplied and size-reduction is done. So it is of the form

$$(xc - \sum_j b_j X_j^{(s+t)}) \cdot y - \sum_j b'_j X_j^{(\kappa)}, \tag{4.1.2}$$

and the zero-testing value can be written as follow using ϕ

$$\begin{aligned}
\boldsymbol{p}_{zt} &\cdot\Big((xc - \sum_j b_j X_j^{(s+t)}) \cdot y - \sum_j b'_j X_j^{(\kappa)}\Big) \mod N &(4.1.3)\\
&= \phi(xcy) - \sum_j b_j \phi(y X_j^{(s+t)}) - \sum_j b'_j \phi(X_j^{(\kappa)}). &(4.1.4)
\end{aligned}$$

By computing $\phi(X_j^{(\kappa)})$ and $\phi(y X_j^{(s+t)})$, we can obtain $\phi(xcy)$. To do so, we first compute ϕ-values of level-κ ladders. We already know ϕ-value of the smallest ladder $X_0^{(\kappa)}$ by Lemma 3.2.1 since it is small and encoding of zero. For the other ladders, the size condition on a, which means the size of encoding is only a problem to adopt the Lemma.

However this problem can be easily solved by size-reduction of ladder themselves. For a ladder $X_k^{(\kappa)}$, we may reduce its size using $X_j^{(\kappa)}, j < k$. More precisely,

$$X_k^{(\kappa)} - \sum_{j<k} b_j X_j^{(\kappa)},$$

satisfies conditions in Lemma 3.2.1 for properly chosen $b_j \in \{0, 1\}$. Then the following holds:

$$\begin{aligned} \boldsymbol{p}_{zt} \cdot (X_k^{(\kappa)} - \sum_{j<k} b_j X_j^{(\kappa)}) \mod N &= \phi\Big(X_k^{(\kappa)} - \sum_{j<k} b_j X_j^{(\kappa)}\Big) \\ &= \phi(X_k^{(\kappa)}) - \sum_{j<k} b_j \phi(X_j^{(\kappa)}). \end{aligned}$$

The first equality comes from Lemma 3.2.1 and the second equality holds by Proposition 4.1.1. Note that the zero-testing value and coefficients b_j's are known values. Therefore using $\phi(X_j^{(\kappa)}), j < k$, we obtain ϕ-value of $X_k^{(\kappa)}$. This procedure can be done from the known value $\phi(X_0^{(\kappa)}) = \boldsymbol{p}_{zt} \cdot X_0^{(\kappa)} \mod N$, inductively. So we can compute ϕ-values of level-κ encodings less than $X_{\gamma'}^{(\kappa)}$.

Note that the size of $yX_j^{(s+t)}$ can be larger than the size of $X_{\gamma'}^{(\kappa)}$. In this case, we can reduce the size of $X_j^{(s+t)}$ using $X_k^{(s+t)}$ for $k < j$ similarly. So it can be obtained by induction.

4.2 Computing Matrix Equation over $\mathbb{Q}$

Now we can compute ϕ-values of large encodings of zero. As in CHLRS attack, suppose we have three encodings x, y and c with y is an encoding of zero.

$$\begin{aligned} x &= \mathsf{CRT}_{(p_i)}\left(\frac{x_i}{z}\right), \\ y &= \mathsf{CRT}_{(p_i)}\left(\frac{y_i g_i}{z^{\kappa-1}}\right), \\ c &= \mathsf{CRT}_{(p_i)}\left(c_i\right). \end{aligned}$$

Then $xcy = \sum x_i r_i y_i u_i + a x_0$ for some integer a. Hence we can obtain $\phi(xcy) = \sum x_i c_i y_i v_i + a v_0$ using ϕ-values of ladders. However CHLRS attack cannot be applied yet since it has additional noise a. If we directly adopt CHLRS approach, it only gives a matrix equation as like

$$\boldsymbol{W_c} = \boldsymbol{X} \times \boldsymbol{C} \times \boldsymbol{Y} + \boldsymbol{A} \cdot v_0$$

with unknown matrix $\boldsymbol{A}$.

Our strategy is to chase the relation of a. Using this equation of a, we remove non-linearity of matrix equation by raising the dimension of matrix. For a notational convenience, we assume level of x, c, y be $1, 0$ and $\kappa - 1$. Then we may rewrite x and y as follow:

$$\begin{aligned} x &= \mathsf{CRT}_{(p_i)}\left(\frac{x_i}{z}\right) = x_i \left[z^{-1}\right]_{p_i} + q_i p_i \quad \text{for each } i, \\ y &= \mathsf{CRT}_{(p_i)}\left(\frac{y_i g_i}{z^{\kappa-1}}\right) = \sum_{i=1}^{n} y_i \left[\frac{g_i}{z^{\kappa-1}}\left(\frac{x_0}{p_i}\right)^{-1}\right]_{p_i} \cdot \frac{x_0}{p_i} + a x_0, \end{aligned}$$

for some $q_i, a \in \mathbb{Z}$. The multiplication of x and c can be written as

$$xc = x_i c_i [z^{-1}]_{p_i} + q_i' p_i,$$

where $q'_i = (xc - x_i c_i [z^{-1}]_{p_i})/p_i$. Then we express xcy as follow:

$$\begin{aligned} & xc \cdot y \\ = \; & (x_i c_i [z^{-1}]_{p_i} + q'_i p_i) \cdot \Big(\sum_{i=1}^{n} y_i \left[\frac{g_i}{z^{\kappa-1}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i} \cdot \frac{x_0}{p_i} + a x_0 \Big) \\ = \; & \sum_{i=1}^{n} \Big(x_i c_i y_i \left[z^{-1} \right]_{p_i} \left[\frac{g_i}{z^{\kappa-1}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i} \cdot \frac{x_0}{p_i} + y_i \left[\frac{g_i}{z^{\kappa-1}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i} q'_i x_0 \Big) + (xc)(a x_0). \end{aligned}$$

Note that $[z^{-1}]_{p_i} \left[\frac{g_i}{z^{\kappa-1}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i} = \left[\frac{g_i}{z^{\kappa}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i} + s_i p_i$ for some integer s_i. Let us define $\theta_i = \left[\frac{g_i}{z^{\kappa-1}} \Big(\frac{x_0}{p_i} \Big)^{-1} \right]_{p_i}$, then $[z^{-1}]_{p_i} \theta_i \frac{x_0}{p_i} = u_i + s_i x_0$. Hence the above can be written

$$\begin{aligned} & xc \cdot y \\ = \; & \sum_{i=1}^{n} (x_i c_i y_i u_i + x_i c_i y_i s_i x_0 + y_i \theta_i q'_i x_0) + xca \cdot x_0 \\ = \; & \sum_{i=1}^{n} x_i c_i y_i u_i + \sum_{i=1}^{n} (x_i c_i y_i s_i + y_i \theta_i q'_i) x_0 + xca \cdot x_0. \end{aligned}$$

Therefore

$$\phi(xcy) = \sum_{i=1}^{n} x_i c_i y_i \cdot v_i + \sum_{i=1}^{n} (x_i c_i y_i s_i + y_i \theta_i q'_i) \cdot v_0 + xca \cdot v_0.$$

By plugging $q'_i = (xc - x_i c_i [z^{-1}]_{p_i})/p_i$ into the equation, we obtain

$$\begin{aligned} \phi(xcy) \; &= \; \sum_{i=1}^{n} y_i (v_i + s_i v_0 - \frac{\theta_i v_0}{p_i} [z^{-1}]_{p_i}) c_i x_i + \sum_{i=1}^{n} y_i \frac{\theta_i v_0}{p_i} cx + a v_0 cx \\ &= \; \sum_{i=1}^{n} y_i w_i c_i x_i + \sum_{i=1}^{n} y_i w'_i cx + a v_0 cx, \end{aligned}$$

where $w_i = v_i + s_i v_0 - \frac{\theta_i}{p_i} [z^{-1}]_{p_i} v_0$ and $w'_i = \frac{\theta_i v_0}{p_i}$. It can be written (over $\mathbb{Q}$)

as follows:

$$\phi(xcy) = \begin{pmatrix} y_1 & y_2 & \cdots & y_n & a \end{pmatrix} \begin{pmatrix} w_1 & & & 0 & w'_1 \\ & w_2 & & & w'_2 \\ & & \ddots & & \vdots \\ & & & w_n & w'_n \\ 0 & & & & v_0 \end{pmatrix} \begin{pmatrix} c_1 x_1 \\ c_2 x_2 \\ \vdots \\ c_n x_n \\ cx \end{pmatrix} \quad (4.2.1)$$

Since $p_i w_i = p_i(v_i + s_i v_0) - \theta_i \left[z^{-1}\right]_{p_i} \cdot v_0 \equiv -\theta_i \left[z^{-1}\right]_{p_i} \cdot v_0 \not\equiv 0 \pmod{p_i}$ w_i is not a zero. Therefore $v_0 \prod_{i=1}^{n} w_i \neq 0$ and so the matrix in Equation (4.2.1) is non singular.

By applying Equation (4.2.1) to various x, y, we have linear matrix equation. Taking for $0 \leq j, k \leq n$,

$$\begin{aligned} x_j &= \mathsf{CRT}_{(p_i)} \left(\frac{x_{ij}}{z} \right), \\ y_k &= \sum_{i=1}^{n} y_{ik} \theta_i \frac{x_0}{p_i} + a_k x_0, \end{aligned}$$

so that the dimension of matrix be $n+1$, then

$$\begin{aligned} \boldsymbol{W}_c &= \begin{pmatrix} y_{10} & \cdots & y_{n0} & a_0 \\ & \ddots & & \vdots \\ & & & \\ y_{1n} & \cdots & y_{nn} & a_n \end{pmatrix} \begin{pmatrix} w_1 & & & 0 & w'_1 \\ & w_2 & & & w'_2 \\ & & \ddots & & \vdots \\ & & & w_n & w'_n \\ 0 & & & & v_0 \end{pmatrix} \begin{pmatrix} c_1 & & & & 0 \\ & c_2 & & & \\ & & \ddots & & \\ & & & c_n & \\ 0 & & & & c \end{pmatrix} \begin{pmatrix} x_{10} & \cdots & x_{1n} \\ & & \\ & \ddots & \vdots \\ x_{n0} & & x_{nn} \\ X_0 & \cdots & X_n \end{pmatrix} \\ &= \boldsymbol{Y} \qquad\qquad \boldsymbol{W} \qquad\qquad \mathsf{diag}(c_1, \cdots, c_n, c) \qquad\qquad \boldsymbol{X}. \end{aligned}$$

Note that w_i and w'_i are regardless of any encodings.

We do the same procedure for the same x_j, y_k and for $c = 1$which is a level-0 encoding of $(1, \cdots, 1)$. Then we have matrix $\boldsymbol{W}_1$ which splits into $(n+1)$-dimension matrices as follow:

$$\begin{aligned} \boldsymbol{W}_1 &= \boldsymbol{Y} \times \boldsymbol{W} \times \mathsf{diag}(1, \cdots, 1) \times \boldsymbol{X} \\ &= \boldsymbol{Y} \times \boldsymbol{W} \times \boldsymbol{X}. \end{aligned}$$

As we mentioned before, $\boldsymbol{W}$ is non-singular. Furthermore $\boldsymbol{Y}$ and $\boldsymbol{X}$ also non-singular matrices with high probability. So we can compute $\boldsymbol{W}_1^{-1}\boldsymbol{W}_c$ over $\mathbb{Q}$, then

$$\begin{aligned}\boldsymbol{W}_1^{-1}\boldsymbol{W}_c &= (\boldsymbol{Y}\boldsymbol{W}\boldsymbol{X})^{-1}\cdot\boldsymbol{Y}\boldsymbol{W}\mathsf{diag}(c_1,\cdots,c_n,c)\boldsymbol{X}\\ &= \boldsymbol{X}^{-1}\mathsf{diag}(c_1,\cdots,c_n,c)\boldsymbol{X}.\end{aligned}$$

It is a similar matrix with $\mathsf{diag}(c_1,\cdots,c_n,c)$, hence they have the same eigenvalues c_i and c. Hence we get c_i and then $c-c_i$ is a multiple of secret p_i.

We repeat the same procedure for another level-0 encoding $c' = \mathsf{CRT}_{(p_i)}(c'_i)$ and construct a matrix $\boldsymbol{W}_{c'}$. Similarly, compute $\boldsymbol{W}_1^{-1}\boldsymbol{W}_{c'}$ and get c'_i. Observe that

$$\gcd(c-c_i, c'-c'_i) = p_i,$$

and so we get secret modulus p_i and it reveals all the other secrets.

Attack Complexity

To compute $\phi(xcy)$, we first compute $\phi(X_j^{(\kappa)}), 0 \le j \le \gamma'$. It is conducted by $O(\gamma^2)$-times comparisons and subtractions of $(\gamma+\gamma')$-bit integers and $(\gamma'+1)$-times zero-testing. Hence its computational cost is $\widetilde{O}(\gamma^2)$ by using fast Fourier transform.

Inversion and multiplication of $(n+1)$-dimension matrices, and computation of eigenvalues and greatest common divisor takes $\widetilde{O}(n^\omega\gamma)$-bit computation with $\omega \le 2.38$. There is one more to consider. We want to $\boldsymbol{W}_1$ be a full rank matrix. Note that the rank of a matrix $\boldsymbol{M} \in \mathbb{Z}^{d\times d}$ can be coputed in time $\widetilde{\mathcal{O}}\left(d^\omega \log \|\vec{M}\|_\infty\right)$ [Sto09]. Hence the total complexity of our attack is $\widetilde{\mathcal{O}}(2(\gamma+\log\ell)(n^\omega \log N)) = \widetilde{\mathcal{O}}(\kappa^{\omega+4}\lambda^{2\omega+6})$ which is a polynomial time in security parameter.

Bibliography

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology–CRYPTO 2001*, pages 213–229. Springer, 2001.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

[BWZ14] Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. *IACR Cryptology ePrint Archive*, 2014.

[CCK+13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In *Eurocrypt*, pages 315–335. Springer, 2013.

[CFL+16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna,*

Austria, May 8-12, 2016, Proceedings, Part I, pages 509–536, 2016.

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New attacks on multilinear maps and their limitations. In *Advances in Cryptology–CRYPTO 2015*, pages 247–266. Springer, 2015.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology–EUROCRYPT 2015*, pages 3–12. Springer, 2015.

[CJL16] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. *IACR Cryptology ePrint Archive*, 2016:139, 2016.

[CLR15] Jung Hee Cheon, Changmin Lee, and Hansol Ryu. Cryptanalysis of the new CLT multilinear maps. *IACR Cryptology ePrint Archive*, 2015:934, 2015.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2013*, pages 476–493, 2013.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2015*, pages 267–286, 2015.

BIBLIOGRAPHY

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology - EUROCRYPT 2013*, pages 1–17, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography*, pages 498–527. Springer, 2015.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure functional encryption without obfuscation. *IACR Cryptology ePrint Archive*, 2014.

[Hal15] Shai Halevi. Cryptographic graded-encoding schemes: Recent developments. TCS+ online seminar, available at `https://sites.google.com/site/plustcs/past-talks/20150318shaihaleviibmtjwatson`, 2015.

[HJ15] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. Technical report, Cryptology ePrint Archive, Report 2015/301, 2015.

[Jou00] Antoine Joux. A one round protocol for tripartite Diffie–Hellman. In *Algorithmic number theory*, pages 385–393. Springer, 2000.

[LS14] Hyung Tae Lee and Jae Hong Seo. Security analysis of multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2014*, pages 224–240, 2014.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. *IACR Cryptology ePrint Archive*, 2016:147, 2016.

BIBLIOGRAPHY

[PTT10] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Optimal authenticated data structures with multilinear forms. In *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Proceedings*, pages 246–264, 2010.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In *TCC*, pages 579–598, 2013.

[RS09] Markus Rückert and Dominique Schröder. Aggregate and verifiably encrypted signatures from multilinear maps without random oracles. In *Advances in Information Security and Assurance, Third International Conference and Workshops, Proceedings*, pages 750–759, 2009.

[Sto09] Arne Storjohann. Integer matrix rank certification. In *Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation*, pages 333–340. ACM, 2009.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2010*, pages 24–43, 2010.

국문초록

다중 선형 함수는 다양한 암호학적 응용이 가능하다. 지금까지 제안 된 다중 선형 함수는 세 가지의 종류가 있다. 처음 제안 된 다중 선형 함수는 아이디얼 격자를 이용하여 설계되었고 두 번째는 정수 위에서 정의 되었다. 마지막으로 방향성이 있는 그래프를 이용하여 정의 된 다중 선형 함수가 있다. 그러나 이들은 모두 안전성 기반 문제가 암호학적으로 안전한 문제로 환원되지 못했을 뿐 아니라 낮은 레벨의 0의 인코딩이 공개되는 경우 안전하지 못함이 밝혀졌다.

그 중에서도 정수 위에서 정의된 다중 선형 함수의 경우 설계와 분석이 반복되고 있다. 코론, 르퐁, 티부시는 2013년 크립토에서 처음으로 중국인의 나머지 정리를 이용한 다중 선형 함수를 제안하였다. 그러나 2015년 유로크립트에서 이 함수가 안전하지 않음이 밝혀졌고 그 후 이를 안전하게 변형하기 위한 다양한 노력이 있었다. 하지만 대부분의 변형 된 함수 역시 안전하지 못하였다. 그러던 중 원 저자들은 공격을 피하기 위한 기법을 추가한 다중 선형 함수를 설계하였고 이는 안전할 것으로 기대되어 2015년 크립토에서 발표되었다.

우리는 이 학위 논문에서 코론 등이 2015년 크립토에서 제안한 다중 선형 함수의 분석 방법을 소개한다. 이 방법은 먼저 제안된 코론 등의 다중 선형 함수의 분석 방법과 핵심을 공유하며 우리는 공개 파라미터를 이용하여 함수의 비밀 정보를 다항식 시간 안에 찾을 수 있다. 코론 등이 처음 제안한 다중 선형 함수는 낮은 레벨의 0의 인코딩을 공개하지 않는 프로그램 난독화 등에 사용될 때의 안전성이 아직 알려지지 않았다. 하지만 새로 제안 된 함수의 경우 곱하기를 위해 추가된 사다리가 낮은 레벨의 0의 인코딩의 역할을 하므로 이 함수가 사용될 수 있는 모든 응용에서 안전하지 않다.

주요어휘: 다중 선형 함수, 그레이디드 인코딩 스킴
학번: 2009-20267