Multilinear maps via secret ring

Chunsheng Gu

School of Computer Engineering, Jiangsu University of Technology, China
{chunsheng_gu}@163.com

Abstract. Garg, Gentry and Halevi (GGH13) described the first candidate multilinear maps using ideal lattices. However, Hu and Jia recently presented an efficient attack on the GGH13 map, which breaks the multipartite key exchange (MPKE) and witness encryption (WE) based on GGH13. In this work, we describe a new variant of GGH13 using secret ring, which preserves the origin functionality of GGH13. The security of our variant depends upon the following new hardness problem. Given the determinant of the circular matrix of some element in a secret ring, the problem is to find this secret ring and reconstruct this element.

Keywords: Multilinear maps, ideal lattices, multipartite key exchange, witness encryption, zeroizing attack

1 Introduction

Garg, Gentry, and Halevi (GGH13) described the first candidate construction of multilinear maps from ideal lattices [16]. Following the framework of GGH13, Coron, Lepoint, and Tibouchi (CLT13) presented another candidate over the integers using Chinese remainder theorem [11]. Gentry, Gorbunov and Halevi (GGH15) [19] constructed a graph-induced multilinear maps from lattices. To improve efficiency, Langlois, Stehlé, and Steinfeld [34] also proposed a variant GGHLite of GGH13.

While cryptographic multilinear maps have found extensive applications, including witness encryption [23], general program obfuscation [18,38], function encryption [18], and other applications [3,17,5,18], all known constructions of multilinear maps exist security problems [7,21,28,2,1,9,10].

For the GGH13 map, Hu and Jia [28] recently described an efficient attack, which breaks the GGH13-based applications on multipartite key exchange (MPKE) and witness encryption (WE) based on the hardness of 3-exact cover problem. Cheon and Lee [8] proposed an attack for the GGH13 map by computing a basis of secret ideal lattice. To fix the GGH13 map, Gentry, Halevi and Lepoint [26] first described a variant of the GGH13 scheme [16], in which the linear zero-testing procedure from [16] is replaced by a quadratic (or higher-degree) procedure. However, Brakerski et al. [2] have showed that this variant of the GGH13 map fails to thwart zeroizing attacks. Then, Halevi [27] described a variant of GGH13 based on graph-induced constraints. But, Coron et al [9] proved that this variant is also insecure.

On the other hand, Gu (Gu map-1) [24] constructed a new multilinear map without encodings of zero, which is a variant of GGH13. Since no encodings of zero are given in the public parameters, MPKE based on Gu map-1 [30] successfully avoids the attack in [28]. However, Gu map-1 cannot be used for the instance of witness encryption based on the hardness of 3-exact cover problem [29]. This is because there is no randomizer in Gu map-1. But the instance of WE based on the hardness of 3-exact cover problem is a strong application of multilinear map. To support witness encryption, Gu (Gu map-2) [25] also suggested another variant of GGH13 with encodings of zero by introducing random matrices.

For the CLT13 map, Cheon, Han, Lee, Ryu, and Stehlé [7] broke CLT13 using zeroizing attack introduced by Garg, Gentry, and Halevi. To fix the CLT13 scheme, Garg, Gentry, Halevi and Zhandry [20], and Boneh, Wu and Zimmerman [4] suggested two candidate fixes of multilinear maps over the integers. However, Coron, Lepoint, and Tibouchi showed that two candidate fixes of CLT13 can also be defeated using extensions of the Cheon et al.'s Attack [7]. By modifying zero-testing parameter, Coron, Lepoint and Tibouchi [13] proposed a new variant of CLT13. Unfortunately, this new version CLT15 is also broken by Cheon, Lee, and Ryu [10], and Minaud and Fouque [35].

For the GGH15 map, Coron et al [9] recently [9] described an efficient attack of GGH15 by extending the Cheon et al.'s attack [7], which breaks the GGH15-based MPKE by generating an equivalent user private key.

Although Gu map-2 has included encodings of zero, it uses the public ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$. In fact, all computations in Gu map-2 are operated over $\mathbb{Z}_q$, and no longer use R. In particular, The disclosure of this ring R may have security weaknesses. In this work, we will focus on the GGH13 variant based on secret rings.

Concurrently, Halevi [27] have observed that this kind of secret ring may improve the security of the scheme. However, Halevi wrote in [27] "It is not clear if there are very many different rings that have such 'nice geometry', and in particular it is not clear if hiding the ring is really possible." Note that 'nice geometry' means to be able to encode arbitrary cosets as small matrices.

Therefore, it can be said that this paper continues the work of [25] and [27]. On one hand, we describe a new variant of the scheme in [25] by using secret rings. On the other hand, we also partially answer the problems presented by Halevi in [27].

1.1 Our results

The GGH13 map. The GGH13 map works in a polynomial ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, where n is a positive integer. A random large integer q , a secret short ring element $g \in R$, and a secret random invertible element $z \in R_q = R/qR$ are chosen during generating the public parameters of the GGH13 construction. A plaintext element e in R/gR is encoded at level-k as $[c/z^k]_q$, where c is a small element in the coset $e_I = e + rg$ for some $r \in R$. Encodings at the same level can be added and subtracted as long as the numerator of the sum of the encodings is not reduced modulo q . Similarly, encodings at levels i, j can be multiplied so long as the numerator of the product of the encodings is not reduced modulo q . A zero-testing parameter $p_{zt} = [hz^{\kappa}/g]_q$ is presented to test for zero at a level-κ encoding. Given a level-κ encoding $u = [c/z^{\kappa}]_q$, one computes $v = [u \cdot\ p_{zt}]_q$ and checks if the norm of v is smaller than q to determine whether u is zero or not.

Our construction. The basic idea of our construction is to use some secret ring $R = \mathbb{Z}[x]/\langle f\rangle$ with monic irreducible polynomial f, and transform GGH13 over R into its matrix version. Concretely speaking, our construction works in a polynomial ring $R = \mathbb{Z}[x]/\langle f\rangle$, where $f = x^n + \sum_{i=0}^{n/2} f_i x^i$ such that $|f_i|$ are small integers. Similar to GGH13, we can generate the public parameter as follows:

First, we generate the level-1 encodings $y_i = [\frac{a_i g + e_i}{z}]_q$ of some non-zero elements $e_i \in R$ and transform them into its matrix form $\mathbf{Y}_i = [\mathbf{T} Rot(y_i)\mathbf{T}^{-1}]_q$ over $\mathbb{Z}_q^{n\times n}$, where $\mathbf{T} \in \mathbb{Z}_q^{n\times n}$, and $Rot(y_i)$ is a matrix whose j-th column is the vector of the coefficients of $x^{j-1}y_i \mod f$.

Then, we generate the zero-testing parameters corresponding to y_i in matrix form $\mathbf{P}_{zt,i} = [\mathbf{T} Rot(\frac{z^{\kappa}(b_i g+e_i)}{g})\mathbf{S}]_q$, where $\mathbf{T}, \mathbf{S} \in \mathbb{Z}_q^{n\times n}$.

Next, we generate a list of level-1 encodings $x_i = [\frac{c_i g}{z}]_q$ of "0" and transform them into its matrix form $\mathbf{X}_i = [\mathbf{T} Rot(x_i)\mathbf{T}^{-1}]_q$ over $\mathbb{Z}_q^{n\times n}$.

Finally, we sample another two matrics $\mathbf{T}_1 \leftarrow D_{\mathbb{Z}^{k_1\times n},\sigma}$, $\mathbf{S}_1 \leftarrow D_{\mathbb{Z}^{n\times k_2},\sigma}$ with $k_1, k_2 \in [n]$, and set $\mathbf{T}^* = \mathbf{T}_1\mathbf{T}^{-1}$, $\mathbf{S}^* = \mathbf{S}^{-1}\mathbf{S}_1$.

It is not difficult to verify that this construction supports addition and multiplication operations of encodings as long as encodings in the operation satisfies the level constraints corresponding to operations. Moreover, given a level-κ encoding $\mathbf{U}$, we can compute $\mathbf{V} = [\mathbf{T}^*\mathbf{U}\mathbf{P}_{zt}\mathbf{S}^*]_q$ and checks if the norm of $\mathbf{V}$ is smaller than q to determine if $\mathbf{U}$ encodes zero, where $\mathbf{P}_{zt} = \sum_i d_i \mathbf{P}_{zt,i}$ with $d_i \in \mathbb{Z}$. Therefore, we have obtained a new variant of GGH13.

To improve the efficiency of our scheme, we can replace $\mathbb{Z}$ in R with $R_\theta = \mathbb{Z}[\theta]/\langle\theta^{\lambda} + 1\rangle$, and set λ as the security parameter. In this case, we can take a constant n and a monic irreducible polynomial f with coefficients $f_i \in R_\theta$ such that $\|f_i\| \leq O(\lambda)$.

Furthermore, our construction supports all applications using GGH13 as public tools of encoding since it includes encodings of zero in the public parameters. As a consequence, our scheme can adaptively support the GGH13-based MPKE and WE that have been broken by Hu and Jia in [28].

1.2 Organization

We briefly recall some background in Section 2. We describe our construction using secret ring in Section 3. We describe new hardness assumption and analyze the security of our scheme in Section 4. We present MPKE and WE based on our construction in Section 5. Finally we draw some conclusions in Section 6.

2 Preliminarie

2.1 Notations

We denote $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ the ring of integers, the field of rational numbers, and the field of real numbers. We take n as a positive integer, and let $[n]$ be the set $\{1, 2, ..., n\}$. We use the absolute minimum residual system of modulo q, namely $[a]_q \in (-q/2, q/2]$. We denote vectors by lowercase bold letters, and matrices by uppercase bold letters, such as $\mathbf{a}, \mathbf{A}$. We write $\mathbf{I}$ as the identity matrix. We denote by a_j the j-th element of $\mathbf{a}$, and $A_{i,j}$ the element of the i-th row and j-th colomn of $\mathbf{A}$. Notation $\|\mathbf{a}\|$ (resp. $\|\mathbf{a}\|_\infty$) denotes the Euclidean norm (resp. the infinity norm) of $\mathbf{a}$.

2.2 Algebra

We denote by $\mathbb{Z}[x]$ and $\mathbb{Q}[x]$ the sets of polynomials with integer and rational coefficients respectively. A polynomial is monic if the coefficient of its highest degree is one. A polynomial is irreducible if it cannot be represented as a product of lower degree polynomials. Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial with degree n. Let R be the polynomial ring $\mathbb{Z}[x]/\langle f \rangle$, and $R_q = R/qR$. For $a \in R_q$, $[a]_q$ denotes each coefficient $[a_i]_q \in (-q/2, q/2]$ of a. For simplicity, we identify degree-$(n-1)$ polynomials with the corresponding n-dimensional vectors whose coordinates are corresponding to the coefficients of the polynomial. For example, we define the p-norm $\|a\|_p$ of a polynomial $a \in \mathbb{Z}[x]$ as the norm of the corresponding vector. When there is no ambiguity, we sometimes abuse $a \in R$ and its corresponding vector $\mathbf{a}$.

In this paper, we need some definitions and lemma introduced by Lyubashevsky and Micciancio [33].

Lemma 2.1 (Lemma 3.2, [33]). Every ideal I of $\mathbb{Z}[x]/\langle f \rangle$, where f is a monic, irreducible polynomial of degree n, is isomorphic to a full-rank lattice in $\mathbb{Z}^n$.

We first recall the definition of the expansion factor of polynomials [33].

Definition 2.2. Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial of degree n. The expansion factor of f is defined as

$$\phi_f = \max_{a \in \mathbb{Z}[x], \deg(a) \leq 2(\deg(f)-1)} \frac{\|a \mod f\|_\infty}{\|a\|_\infty}.$$

Lemma 2.3 (Theorem 3.5, [33]). Given a monic polynomial $f \in \mathbb{Z}[x]$:

(1) If $f = x^n + \sum_{i=0}^{n/2} f_i x^i$, then $\phi_f \leq (\|2f\|_1)^3$.

(2) If $f = x^n + \sum_{i=0}^{n-1} f_i x^i$, then $\phi_f \leq (\|2f\|_1)^{n-1}$.

2.3 Lattices

An n-dimensional full-rank lattice $L \subset \mathbb{R}^n$ is the set of all integer linear combinations $\sum_{i=1}^{n} x_i \mathbf{b}_i$ of n linearly independent vectors $\mathbf{b}_i \in \mathbb{R}^n$. If we arrange the vectors $\mathbf{b}_i$ as the columns of matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, then $L = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$. We say that $\mathbf{B}$ spans L if $\mathbf{B}$ is a basis for L. Given a basis $\mathbf{B}$ of L , we define $P(\mathbf{B}) = \{\mathbf{Bx}|\mathbf{x} \in \mathbb{R}^n \text{ and } x_i \in [-1/2, 1/2)\}$ as the parallelization corresponding to $\mathbf{B}$. Let $\det(\mathbf{B})$ denote the determinant of $\mathbf{B}$.

Given $g \in R$, let $I = \langle g\rangle$ be the principal ideal in R generated by g. We denote the $\mathbb{Z}$-basis of I by $Rot(g) = (g \mod f, xg \mod f, ..., x^{n-1}g \mod f)$. Namely, the i-th column of $Rot(g)$ is the vector of the coefficients of $x^{i-1}g \mod f$.

Given $\mathbf{c} \in \mathbb{R}^n$, $\sigma > 0$, the Gaussian distribution of a lattice L is defined as $D_{L,\sigma,\mathbf{c}} = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(L)$ for $\mathbf{x} \in L$, where $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/\sigma^2)$, $\rho_{\sigma,\mathbf{c}}(L) = \sum_{\mathbf{x}\in L} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. As shorthand, we will write $D_{L,\sigma}$ instead of $D_{L,\sigma,\mathbf{0}}$. We denote a Gaussian sample as $\mathbf{x} \leftarrow D_{L,\sigma}$ (or $d \leftarrow D_{I,\sigma}$) over the lattice L (or I).

Definition 2.4 (Smoothing parameter [36]). For an n-dimensional lattice L, and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(L)$ is defined to be the smallest s such that $\rho_{1/s}(L^*\backslash\{0\}) \le \epsilon$.

Lemma 2.5 (Lemma 3.3 [36]). For any n-dimensional lattice L and a negligible function $\epsilon(n)$, $\eta_\epsilon(L) \le \sqrt{\omega(\log n)} \cdot \lambda_n(L)$, where $\omega(\log n)$ is any superlogarithmic function.

Lemma 2.6 (Lemma 4.4 [36]). For any n-dimensional lattice L, vector $\mathbf{c} \in \mathbb{R}^n$, and reals $0 < \epsilon < 1$, $s \ge \eta_\epsilon(L)$, we have

$$\Pr_{\mathbf{x}\leftarrow D_{L,s,\mathbf{c}}} \{\|x - c\| > s\sqrt{n}\} \le \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}.$$

2.4 Multilinear Maps

Definition 2.7 (Multilinear Map [3]). For $\kappa+1$ cyclic groups $G_1, ..., G_\kappa, G_T$ of the same order q , a κ-multilinear map $e : G_1 \times ... \times G_\kappa \to G_T$ has the following properties:

(1) Elements $\{g_j \in G_j\}_{j=1,...,\kappa}$, index $j \in [\kappa]$, and integer $a \in \mathbb{Z}_q$ hold that

$$e(g_1, ..., a \cdot g_j, ..., g_\kappa) = a \cdot e(g_1, ..., g_\kappa)$$

(2) Map e is non-degenerate in the following sense: if elements $\{g_j \in G_j\}_{j=1,...,\kappa}$ are generators of their respective groups, then $e(g_1, ..., g_\kappa)$ is a generator of G_T .

Definition 2.8 (κ-Graded Encoding System [16]). A κ-graded encoding system over R is a set system of $S = \{S_j^{(a)} \in R : a \in R, j \in [\kappa]\}$ with the following properties:

(1) For every index $j \in [\kappa]$, the sets $S = \{S_j^{(a)} \in R : a \in R\}$ are disjoint.

(2) Binary operations'+' and'−' exist, such that every a_1, a_2, every index $j \in [\kappa]$, and every $u_1 \in S_j^{(a_1)}$ and $u_2 \in S_j^{(a_2)}$ hold that $u_1 + u_2 \in S_j^{(a_1+a_2)}$ and

$u_1 - u_2 \in S_j^{(a_1 - a_2)}$, where $a_1 + a_2$ and $a_1 - a_2$ are the addition and subtraction operations in R respectively.

(3) Binary operation '$\times$' exists, such that every a_1, a_2, every index $j_1, j_2 \in [\kappa]$ with $j_1 + j_2 \le \kappa$, and every $u_1 \in S_{j_1}^{(a_1)}$ and $u_2 \in S_{j_2}^{(a_2)}$ hold that $u_1 \times u_2 \in S_{j_1+j_2}^{(a_1 \times a_2)}$, where $a_1 \times a_2$ is the multiplication operation in R and $j_1 + j_2$ is the integer addition.

3 Multilinear map via secret ring

Setting the parameters. Let λ be the security parameter, κ the multilinearity level, n the dimension of elements of R. Concrete parameters are set as $\sigma = \lambda$, $\alpha = \lambda^2$, $n = \lambda^2$, $\tau = 2n$, $\beta = \sqrt{q}$, $q \ge 2^{12\kappa+1}\lambda^{64\kappa+124}$, $k_1, k_2 \in [n]$ such that $k_1 k_2 < n$.

3.1 Construction

Instance generation $\text{Par} \leftarrow \text{InstGen}(1^\lambda, 1^\kappa)$:

(1) Choose a prime $q \ge 2^{16\kappa\lambda} n^{O(\kappa)}$.

(2) Choose a monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$ such that $f = x^n + \sum_{i=0}^{n/2} f_i x^i$ with $|f_i| < \sigma$. Let $R = \mathbb{Z}[x]/\langle f \rangle$, $R_q = R/qR$, and $\mathbb{K} = \mathbb{Q}[x]/\langle f \rangle$.

(3) Sample $g \leftarrow D_{R,\sigma}$ so that g is a prime element in R and $\|g^{-1}\| \le n$, and a random element $z \leftarrow R_q$ so that $z^{-1} \in R_q$.

(4) Sample $\mathbf{T}, \mathbf{S} \in \mathbb{Z}_q^{n \times n}$ so that $\mathbf{T}^{-1}, \mathbf{S}^{-1} \in \mathbb{Z}_q^{n \times n}$.

(5) Sample $\mathbf{T}_1 \leftarrow D_{\mathbb{Z}^{k_1 \times n}, \sigma}$, $\mathbf{S}_1 \leftarrow D_{\mathbb{Z}^{n \times k_2}, \sigma}$, and set $\mathbf{T}^* = \mathbf{T}_1 \mathbf{T}^{-1}$, $\mathbf{S}^* = \mathbf{S}^{-1}\mathbf{S}_1$.

(6) For $i \in [\tau]$,

(6.1) sample elements $a_i, e_i, c_i \leftarrow D_{R,\alpha}$, $b_i \leftarrow D_{R,\beta}$;

(6.2) set $\mathbf{Y}_i = [\mathbf{T}Rot(\frac{a_i g + e_i}{z})\mathbf{T}^{-1}]_q$, $\mathbf{X}_i = [\mathbf{T}Rot(\frac{c_i g}{z})\mathbf{T}^{-1}]_q$;

(6.3) set $\mathbf{P}_{zt,i} = [\mathbf{T}Rot(\frac{z^\kappa(b_i g + e_i)}{g})\mathbf{S}]_q$.

(7) Output the public parameters $\text{Par} = \{q, \{\mathbf{Y}_i, \mathbf{X}_i, \mathbf{P}_{zt,i}\}_{i \in [\tau]}, \mathbf{T}^*, \mathbf{S}^*\}$.

Generating level-t encoding $\mathbf{U} \leftarrow \text{Enc}(\text{Par}, t, \mathbf{d}, \mathbf{r})$:

Given $\mathbf{d} \leftarrow D_{\mathbb{Z}^\tau, \alpha}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}^\tau, \alpha}$, generate a level-$t$ encoding

$$\mathbf{U} = \left[\sum_{i=1}^{\tau} d_i \cdot (\mathbf{Y}_i)^t + \sum_{i=1}^{\tau} r_i \cdot (\mathbf{X}_i)^t\right]_q.$$

Adding encodings $\mathbf{U} \leftarrow \text{Add}(\text{Par}, t, \mathbf{U}_i, \cdots, \mathbf{U}_k)$:

Given k level-t encodings $\mathbf{U}_i, i \in [k]$, their sum $\mathbf{U} = \left[\sum_{i=1}^{k} \mathbf{U}_i\right]_q$ is a level-t encoding.

Multiplying encodings $\mathbf{U} \leftarrow \text{Mul}(\text{Par}, 1, \mathbf{U}_1, \cdots, \mathbf{U}_k)$:

Given k level-1 encodings $\mathbf{U}_i, i \in [k]$, their product $\mathbf{U} = \left[\prod_{i=1}^{k} \mathbf{U}_i\right]_q$ is a level-k encoding.

Zero testing $\text{IsZero}(\text{Par}, \mathbf{U}, \mathbf{d})$:

Given a level-κ encoding $\mathbf{U} = \left[\mathbf{T}Rot(\frac{rg+e}{z^{\kappa}})\mathbf{T}^{-1}\right]_q$ and a vector $\mathbf{d} \leftarrow D_{\mathbb{Z}^{\tau},\alpha}$,

(1) we compute $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_i \mathbf{P}_{zt,i}$, $\mathbf{V} = \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q$;

(2) we check whether $\|\mathbf{V}\|$ is short. That is,

$$\text{IsZero}(\text{Par}, \mathbf{U}, \mathbf{d}) = \begin{cases} 1, & \text{if } \|\mathbf{V}\| < q^{3/4}; \\ 0, & \text{otherwise.} \end{cases}$$

Extraction $sk \leftarrow \text{Ext}(\text{Par}, \mathbf{U}, \mathbf{d})$:

Given a level-κ encoding $\mathbf{U}$ and a vector $\mathbf{d} \leftarrow D_{\mathbb{Z}^{\tau},\alpha}$,

(1) we compute $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_i \mathbf{P}_{zt,i}$, $\mathbf{V} = \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q$;

(2) we collect $(\log q)/4 - \lambda$ most-significant bits of each element of the $k_1 \times k_2$-matrix $\mathbf{V}$. That is,

$$\text{Ext}(\text{Par}, \mathbf{U}, \mathbf{d}) = \text{MSB}(\mathbf{V}).$$

Remark 3.1. (1) Note that the above construction is similar to that in [25]. The main difference is that this scheme uses some secret ring $R = \mathbb{Z}[x]/\langle f \rangle$, whereas the scheme in [25] uses the public ring $R = \mathbb{Z}[x]/\langle x^n + 1 \rangle$. As a result, this difference can lead to significant differences in their security. Concrete details are analyzed later. (2) It is easy to transform the above symmetric construction into an asymmetric variant by using different elements z_j and random matrices $\mathbf{T}_j$.

3.2 Correctness

To prove the correctness, we first describe two simple lemmas.

Lemma 3.2. If $a, b \in R$, then $Rot(a)Rot(b) = Rot(ab \mod f)$.

Proof. We only need to show that the first column of $Rot(a)Rot(b)$ is equal to the vector corresponding to the polynomial $ab \mod f$.

It is easy to verify that the first column of $Rot(a)Rot(b)$ is corresponding to $\sum_{i=0}^{n-1} b_i \cdot (ax^i \mod f) = (a \cdot \sum_{i=0}^{n-1} b_i x^i) \mod f = ab \mod f$. ∎

Lemma 3.3. If $a, b \in R_q$, then $[Rot(a)Rot(b)]_q = [Rot(ab \mod f)]_q$.

Proof. The proof is completely similar to that of Lemma 3.2. ∎

Lemma 3.4. The algorithm $\text{InstGen}(1^{\lambda}, 1^{\kappa})$ runs in PPT.

Proof. A prime element $g \in R$ can be generated in PPT according to Landau's prime number theorem (Theorem 5.17, [15]). It is easy to verify that all other operations can run in PPT. ∎

Lemma 3.5. The encoding $\mathbf{U} \leftarrow \text{Enc}(\text{Par}, t, \mathbf{d}, \mathbf{r})$ is a level-t encoding.

Proof. By $\mathbf{Y}_i^t = \left[\mathbf{T}\ Rot(\frac{a_i g+e_i}{z})^t\mathbf{T}^{-1}\right]_q$ and $\mathbf{X}_i^t = \left[\mathbf{T}\ Rot(\frac{c_i g}{z})^t\mathbf{T}^{-1}\right]_q$, we have

$$\begin{aligned}\mathbf{U} &= \left[\sum\nolimits_{i=1}^{\tau} d_i\cdot\mathbf{Y}_i^t + \sum\nolimits_{i=1}^{\tau} r_i\cdot\mathbf{X}_i^t\right]_q\\ &= \left[\mathbf{T}\ Rot\left(\frac{\sum\nolimits_{i=1}^{\tau} d_i a_i^{'} g + \sum\nolimits_{i=1}^{\tau} r_i c_i^{'} g + \sum\nolimits_{i=1}^{\tau} d_i e_i^t}{z^t}\right)\mathbf{T}^{-1}\right]_q\\ &= \left[\mathbf{T}\ Rot(\frac{ag+e}{z^t})\mathbf{T}^{-1}\right]_q\end{aligned}$$

where $a_i^{'} = ((a_i g+e_i)^t - e_i^t)/g$, $c_i^{'} = (c_i g)^t/g$, $a = \sum\nolimits_{i=1}^{\tau} d_i\cdot a_i^{'} + \sum\nolimits_{i=1}^{\tau} r_i c_i^{'}$, and $e = \sum\nolimits_{i=1}^{\tau} d_i\cdot e_i^t$. ∎

Lemma 3.6. $\mathbf{U} = \mathrm{Add}(\mathrm{Par}, t, \mathbf{U}_i, \cdots, \mathbf{U}_k)$ is a level-t encoding.

Proof. Given level-t encodings $\mathbf{U}_i, i\in[k]$. Let $\mathbf{U}_i = \left[\mathbf{T}\ Rot(\frac{r_i^{'} g+e_i^{'}}{z^t})\mathbf{T}^{-1}\right]_q$. Then

$$\mathbf{U} = \left[\sum\nolimits_{i=1}^{k}\mathbf{U}_i\right]_q = \left[\mathbf{T}\ Rot(\frac{rg+e}{z^t})\mathbf{T}^{-1}\right]_q,$$

where $r = \sum\nolimits_{i=1}^{k} r_i^{'}$ and $e = \sum\nolimits_{i=1}^{k} e_i^{'}$. ∎

Lemma 3.7. $\mathbf{U} \leftarrow \mathrm{Mul}(\mathrm{Par}, 1, \mathbf{U}_1, \cdots, \mathbf{U}_k)$ is a level-k encoding.

Proof. Given level-1 encodings $\mathbf{U}_i, i\in[k]$, let $\mathbf{U}_i = \left[\mathbf{T}\ Rot(\frac{r_i^{'} g+e_i^{'}}{z})\mathbf{T}^{-1})\right]_q$. Then, by Lemma 3.3 we have

$$\mathbf{U} = \left[\mathbf{T}\ Rot(\frac{rg+e}{z^k})\mathbf{T}^{-1}\right]_q,$$

where $e = \prod\nolimits_{i=1}^{k} e_i^{'}$, $r = (\prod\nolimits_{i=1}^{k}(r_i^{'} g+e_i^{'}) - e)/g$.

Lemma 3.8. $\mathrm{IsZero}(\mathrm{Par}, \mathbf{U}, \mathbf{d})$ correctly determines whether $\mathbf{U}$ is a level-κ encoding of zero or not.

Proof. By Lemma 2.3 and $f = x^n + \sum\nolimits_{i=0}^{n/2} f_i x^i$, we have

$$\phi_f = \|2f\|_1^3 = (n\sigma)^3 = \lambda^9.$$

Given $\mathbf{P}_{zt,i} = \left[\mathbf{T}Rot(\frac{z^\kappa(b_i g+e_i)}{g})\mathbf{S}\right]_q$ and $\mathbf{d}\leftarrow D_{\mathbb{Z}^\tau,\alpha}$, we have

$$\mathbf{P}_{zt} = \sum\nolimits_{i=1}^{\tau} d_i\mathbf{P}_{zt,i} = \left[\mathbf{T}Rot(\frac{z^\kappa(hg+c)}{g})\mathbf{S}\right]_q,$$

where $h = \sum\nolimits_{i=1}^{\tau} d_i b_i$, and $c = \sum\nolimits_{i=1}^{\tau} d_i e_i$.

By $b_i\leftarrow D_{R,\beta}$, $e_i\leftarrow D_{R,\alpha}$, and $\mathbf{d}\leftarrow D_{\mathbb{Z}^\tau,\alpha}$, we obtain

$$\begin{aligned}\|h\| = \|\sum\nolimits_{i=1}^{\tau} d_i b_i\| \le \tau|d_i|\|b_i\| \le 2n^2\alpha\beta \le 2\lambda^7\sqrt{q},\\ \|c\| = \|\sum\nolimits_{i=1}^{\tau} d_i e_i\| \le \tau|d_i|\|e_i\| \le 2n^2\alpha^2 \le 2\lambda^8,\\ \|hg+c\| \le 2\phi_f\|h\|\|g\| \le \lambda^9\cdot 2\lambda^7\sqrt{q}\cdot\sqrt{n}\sigma \le 2\lambda^{18}\sqrt{q}.\end{aligned}$$

Without loss of generality, we let $\mathbf{U} = \left[\prod_{j=1}^{\kappa} \mathbf{U}_j\right]_q$ since $\mathbf{U}$ is a level-κ encoding.

According to $\mathbf{U}_j = \mathrm{Enc}(\mathrm{Par}, 1, \mathbf{d}_j, \mathbf{r}_j)$, we have

$$\mathbf{U}_j = \left[\sum_{i=1}^{\tau} d_{j,i} \cdot \mathbf{Y}_i + \sum_{i=1}^{\tau} r_{j,i} \cdot \mathbf{X}_i\right]_q = \left[\mathbf{T}\ Rot\big(\frac{a_{(j)}g + e_{(j)}}{z}\big)\mathbf{T}^{-1}\right]_q,$$

where $a_{(j)} = \sum_{i=1}^{\tau} d_{j,i}a_i + \sum_{i=1}^{\tau} r_{j,i}c_i$, $e_{(j)} = \sum_{i=1}^{\tau} d_{j,i}e_i$.

Similarly, by $a_i, e_i, c_i \leftarrow D_{R,\alpha}$, and $\mathbf{d}_j \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, we have

$$\|a_{(j)}\| = \|\sum_{i=1}^{\tau} d_{j,i}a_i + \sum_{i=1}^{\tau} r_{j,i}c_i\| \leq 2\tau|d_{j,i}|\|a_i\| \leq 4n^2\alpha^2.$$
$$\|e_{(j)}\| = \|\sum_{i=1}^{\tau} d_{j,i}e_i\| \leq \tau|d_{j,i}|\|e_i\| \leq 2n^2\alpha^2.$$

So, we get

$$\begin{aligned}\|\prod_{j=1}^{\kappa} \big(a_{(j)}g + e_{(j)}\big)\| &\leq \phi_f^{\kappa-1} \prod_{j=1}^{\kappa} \|\big(a_{(j)}g + e_{(j)}\big)\| \\ &\leq \phi_f^{\kappa-1}\big(\max_{j\in[\kappa]} 2\|a_{(j)}\|\|g\|\phi_f\big)^{\kappa}\| \\ &\leq \phi_f^{\kappa-1}\big(2 \cdot 4n^2\alpha^2 \cdot \sqrt{n}\sigma \cdot \phi_f\big)^{\kappa} \\ &\leq \phi_f^{2\kappa-1}\big(8n^{2.5}\alpha^2\sigma\big)^{\kappa} \\ &\leq 8^{\kappa}\lambda^{16\kappa-4}.\end{aligned}$$

We now analyze the encoding $\mathbf{U}$ encodes 0 or non-zero, respectively.

For simplicity, we let

$$\mathbf{U} = \left[\mathbf{T}\ Rot\big(\frac{ag + e}{z^{\kappa}}\big)\mathbf{T}^{-1}\right]_q,$$

where $e = \prod_{j=1}^{\kappa} e_{(j)}$, and $a = \prod_{j=1}^{\kappa} \big(a_{(j)}g + e_{(j)}\big) - e$.

If $\mathbf{U}$ is a level-κ encoding of zero (i.e. $e = 0$), then,

$$\begin{aligned}\mathbf{V} &= \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q \\ &= \left[\mathbf{T}^* \cdot \mathbf{T}Rot\big(\frac{ag}{z^{\kappa}}\big)\mathbf{T}^{-1} \cdot \mathbf{T}\ Rot\big(\frac{z^{\kappa}(hg + c)}{g}\big)\mathbf{S} \cdot \mathbf{S}^*\right]_q \\ &= \left[\mathbf{T}_1 Rot\big(a(hg + c)\big)\mathbf{S}_1\right]_q\end{aligned}$$

Since $\|\mathbf{T}_1\| = \|\mathbf{S}_1\| \leq n\sqrt{n}\sigma \leq \lambda^4$, we obtain

$$\begin{aligned}\|\mathbf{T}_1 Rot\big(a(hg + c)\big)\mathbf{S}_1\| &\leq \|\mathbf{T}_1\| \cdot \|Rot\big(a(hg + c)\big)\| \cdot \|\mathbf{S}_1\| \\ &\leq \lambda^8 \cdot \|a(hg + c)\| \\ &\leq \lambda^8 \cdot \|a\| \cdot \|hg + c\| \cdot \phi_f \\ &\leq \lambda^8 \cdot 8^{\kappa}\lambda^{16\kappa-4} \cdot 2\lambda^{18}\sqrt{q} \cdot \lambda^9 \\ &< q^{3/4}\end{aligned}$$

Thus, $\mathbf{V}$ is not reduced modulo q. That is, $[\mathbf{V}]_q = \mathbf{V}$.

If $\mathbf{U}$ is a level-κ encoding of non-zero, i.e. $e \neq 0 \mod g$. Thus,

$$\begin{aligned}\mathbf{V} &= \Big[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\Big]_q \\ &= \Big[\mathbf{T}_1 Rot(\frac{ag+e}{z^\kappa})\ Rot(\frac{z^\kappa(hg+c)}{g})\mathbf{S}_1\Big]_q \\ &= \Big[\mathbf{T}_1 Rot(\frac{(ag+e)(hg+c)}{g})\mathbf{S}_1\Big]_q \\ &= \Big[\mathbf{T}_1 Rot\big(a(hg+c)\big)\mathbf{S}_1 + \mathbf{T}_1 Rot\big(\frac{e(hg+c)}{g}\big)\mathbf{S}_1\Big]_q\end{aligned}$$

By Lemma 4 in [16], we have

$$\|\mathbf{T}_1 Rot(\frac{e(hg+c)}{g})\mathbf{S}_1\| \approx q.$$

Again using $\|\mathbf{T}_1 Rot\big(a(hg+c)\big)\mathbf{S}_1\| \leq q^{3/4}$, we have $\|\mathbf{V}\| \approx q$. ∎

Lemma 3.9 Suppose that two level-κ encodings $\mathbf{U}_1$, $\mathbf{U}_2$ encode same plaintext, then

$$\text{Ext}(\text{Par}, \mathbf{U}_1, \mathbf{d}) = \text{Ext}(\text{Par}, \mathbf{U}_2, \mathbf{d}).$$

Proof. Assume that $\mathbf{U}_i = \Big[\mathbf{T}\ Rot(\frac{u_i g+e}{z^\kappa})\mathbf{T}^{-1}\Big]_q$, $i \in [2]$. So, we have

$$\begin{aligned}\mathbf{V}_i &= \Big[\mathbf{T}^* \cdot \mathbf{U}_i \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\Big]_q \\ &= \Big[\mathbf{T}_1 Rot(\frac{u_i g+e}{z^\kappa})\ Rot(\frac{z^\kappa(hg+c)}{g})\mathbf{S}_1\Big]_q \\ &= \Big[\mathbf{T}_1 Rot(\frac{(u_i g+e)(hg+c)}{g})\mathbf{S}_1\Big]_q \\ &= \Big[\mathbf{T}_1 Rot\big(u_i(hg+c)\big)\mathbf{S}_1 + \mathbf{T}_1 Rot\big(\frac{e(hg+c)}{g}\big)\mathbf{S}_1\Big]_q\end{aligned}$$

For our parameters setting, $\|\mathbf{T}_1 Rot\big(u_i(hg+c)\big)\mathbf{S}_1\| \leq q^{3/4}$. When $e \neq 0 \mod g$, by Lemma 4 in [16], we have

$$\|\mathbf{T}_1 Rot(\frac{e(hg+c)}{g})\mathbf{S}_1\| \approx q.$$

Thus, $\text{Ext}(\text{Par}, \mathbf{U}_1, \mathbf{d}) = \text{Ext}(\text{Par}, \mathbf{U}_2, \mathbf{d})$ with overwhelming probability. ∎

3.3 Variant

To improve the efficiency of our construction, we use polynomial ring instead of integer ring. Concretely speaking, we let $R_\theta = \mathbb{Z}[\theta]/\langle\theta^\lambda + 1\rangle$, and $R = R_\theta[x]/f$, $R_q = R/qR$. In this case, we can set n as a constant and $f = x^n + \sum_{i=0}^{n-1} f_i x^i$ with coefficients $f_i \in R_\theta$ such that $\|f_i\| \leq O(\lambda)$.

It is easy to verify that this variant of our construction is still correct.

4 Security

4.1 Hardness assumption

Consider the following security experiment:

(1) Par $\leftarrow$ InstGen$(1^\lambda, 1^\kappa)$

(2) For $j = 0$ to κ:

Sample $\mathbf{d}_j \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, $\mathbf{r}_j \leftarrow D_{\mathbb{Z}^\tau,\alpha}$;

Generate level-1 encoding $\mathbf{U}_j = \left[\sum_{i=1}^{\tau} d_{j,i}\mathbf{Y}_i + \sum_{j=1}^{\tau} r_{j,i}\mathbf{X}_i\right]_q$.

(3) Set $\mathbf{U} = \left[\prod_{j=1}^{k} \mathbf{U}_j\right]_q$ and $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_{0,i}\mathbf{P}_{zt,i}$.

(4) Set $\mathbf{V}_c = \mathbf{V}_d = \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q$.

(5) Sample $\mathbf{s} \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, and set $\mathbf{P}_{zt,r} = \sum_{i=1}^{\tau} s_i\mathbf{P}_{zt,i}$.

(6) Set $\mathbf{V}_r = \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt,r} \cdot \mathbf{S}^*\right]_q$.

Definition 4.1. (ext-GCDH/ext-GDDH). According to the security experiment, the ext-GCDH and ext-GDDH are defined as follows:

Level-κ extraction CDH (ext-GCDH): Given $\{\text{Par}, \mathbf{U}_0, \cdots, \mathbf{U}_\kappa\}$, output a level-$\kappa$ extraction encoding $\mathbf{W} \in \mathbb{Z}_q^{k_1 \times k_2}$ such that $\left\|[\mathbf{V}_c - \mathbf{W}]_q\right\| \leq q^{3/4}$.

Level-κ extraction DDH (ext-GDDH): Given $\{\text{Par}, \mathbf{U}_0, \cdots, \mathbf{U}_\kappa, \mathbf{V}\}$, distinguish between

$$D_{ext-GDDH} = \{\text{Par}, \mathbf{U}_0, \cdots, \mathbf{U}_\kappa, \mathbf{V}_d\},$$
$$D_{ext-RAND} = \{\text{Par}, \mathbf{U}_0, \cdots, \mathbf{U}_\kappa, \mathbf{V}_r\}.$$

4.2 Cryptanalysis

In this section, we first generate easily computable quantities in our construction, then analyze possible attacks using these quantities.

4.2.1 Easily computable quantities.

Since $\mathbf{T}, \mathbf{S} \in \mathbb{Z}_q^{n\times n}$, we must compute $\mathbf{V} = \left[\mathbf{T}^* \cdot \mathbf{U} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q$ to eliminate $\mathbf{T}, \mathbf{S}$, where $\mathbf{U}$ is a level-κ encoding, $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_i\mathbf{P}_{zt,i}$ with $\mathbf{d} \in \mathbb{Z}^\tau$.

For simplicity, we write

$$\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_i\mathbf{P}_{zt,i} = \left[\mathbf{T}Rot(\frac{z^\kappa(hg+c)}{g})\mathbf{S}\right]_q = \left[\mathbf{T}Rot(\frac{z^\kappa h_1}{g})\mathbf{S}\right]_q,$$

where $h_1 = hg + c$.

Let integer $t > 0$, $\mathbf{U}_{i,j,t} = \mathbf{X}_i^t \mathbf{Y}_j^{\kappa-t}$ be a level-κ encoding of zero. So, we have

$$\begin{aligned}\mathbf{V}_{i,j,t} &= \Big[\mathbf{T}^* \cdot \mathbf{U}_{i,j,t} \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\Big]_q \\ &= \Big[\mathbf{T}_1 \cdot \mathit{Rot}(c_i g)^t \cdot \mathit{Rot}(y_j)^{\kappa-t} \cdot \mathit{Rot}(\frac{h_1}{g}) \cdot \mathbf{S}_1\Big]_q \\ &= \Big[\mathbf{T}_1 \cdot \mathit{Rot}\big(c_i^t \cdot g^{t-1} \cdot y_j^{\kappa-t} \cdot h_1\big) \cdot \mathbf{S}_1\Big]_q \\ &= \mathbf{T}_1 \cdot \mathit{Rot}\big(c_i^t \cdot g^{t-1} \cdot y_j^{\kappa-t} \cdot h_1\big) \cdot \mathbf{S}_1,\end{aligned}$$

where $y_j = a_j g + e_j$.

The final equality is to use the fact that $\mathbf{V}_{i,j,t}$ are not reduced modulo q.

4.2.2 Attacks on secret ring

Using different level-κ encodings $\mathbf{U}_{i,j,t}$, we can generate a list of $\mathbf{V}_{i,j,t}$ that are not reduced modulo q. Furthermore, we can keep some encoding in $\mathbf{U}_{i,j,t}$ unchanged, and change other encodings to produce $(n \times n)$-dimensional matrices in combination. In this way, we can compute the determinants of $\mathit{Rot}(c_i)$, $\mathit{Rot}(g)$, $\mathit{Rot}(y_j)$, $\mathit{Rot}(h_1)$ by applying GCD algorithm.

1. Attack of known ring

We choose cyclotomic rings as known rings. Let $f = x^n + 1$ be the public polynomial that is used to generated the ring R, where n is power of 2. Let $p = \det(\mathit{Rot}(g))$. Since p is a prime, $\mathrm{GCD}(f, g) = x - \mu \mod p$. Using the quantum algorithm [6], we may recover short generator of $\langle g \rangle$.

Without loss of generality, let $\mathbf{V}_i = \mathbf{T}_1 \mathit{Rot}(r_i) \mathbf{S}_1$ be a list of $(n \times n)$-dimensional matrices generated by the public parameters. Assume that we obtain r_i by the quantum algorithm [6], then we can compute $\mathbf{V}_1 \mathbf{V}_2^{-1} = \mathbf{T}_1 \mathit{Rot}(r_1/r_2) \mathbf{T}_1^{-1}$. Since we have obtained r_1/r_2, we can compute $\mathbf{T}_1$ and $\mathbf{S}_1$. As a consequence, we can further find other secret parameters in our construction. Therefore, there may be security weaknesses in the constructions on public rings.

Note that in fact, the quantum algorithm in [6] can only recover short generator of r_i, cannot guarantee this short generator must be r_i. In this case, the above attack may not be successful.

2. Attack of secret ring

If we keep f secret and choose $f = x^n + \sum_{i=0}^{n/2} f_i x^i$ with $|f_i| < \sigma$, as far as we know, our construction did not find security weaknesses.

According to Landau's prime number theorem (Theorem 5.17, [15]), the number of the irreducible polynomials f that satisfy constraints is exponential in n. For any irreducible polynomial $f = x^n + \sum_{i=0}^{n/2} f_i x^i \in \mathbb{Z}[x]$, if there exists linear factor $x - \mu$ in f over modulo p, then (p, μ) can generate a principal ideal that has small generator. However, given $p = \det(\mathit{Rot}(g))$, we do not know how to effectively find g or f. We conjecture that this problem is difficult.

3. Lattice reduction attack

Given $\mathbf{V}_i = \mathbf{T}_1 Rot(r_i)\mathbf{S}_1$, we can find $\mathbf{T}_1$, $\mathbf{S}_1$ by lattice reduction algorithms [14,32], and further obtain $Rot(r_i)$ and f. However, when n is large enough, all known lattice reduction algorithms run in exponential time in λ.

4.2.3 Hu-Jia Attack.

In this section, we show that the Hu-Jia attack [28] does not work for our construction.

1. Hu-Jia Attack

Hu-Jia attack includes three steps. That is, Step 1 generates an equivalent level-0 encoding for a level-1 encoding; Step 2 computes an equivalent level-0 encoding for the product of several level-0 encodings; Step 3 obtains the shared secret key of an equivalent product level-0 encoding by the modified encoding/decoding. The concrete details are as follows:

Step 1: Generate an equivalent level-0 encoding for a level-1 encoding

(1) Let par be the public parameters of the GGH13 map, i.e.

$$\text{par} = \{q, y = [\frac{1+ag}{z}]_q, x_1 = [\frac{c_1 g}{z}]_q, x_2 = [\frac{c_2 g}{z}]_q, p_{zt} = [\frac{hz^\kappa}{g}]_q\}.$$

We generate special decodings $\{y_{(1)}, x_{(i)}, i = 1, 2\}$, where

$$\begin{aligned} y_{(1)} &= \left[p_{zt} y^{\kappa-1} x_1\right]_q = h(1+ag)^{\kappa-1} c_1, \\ x_{(i)} &= \left[p_{zt} y^{\kappa-2} x_i x_1\right]_q = h(1+ag)^{\kappa-2}(c_i g)c_1, i = 1, 2, \end{aligned}$$

where $y_{(1)}$, $x_{(i)}$ are not reduced modulo q.

(2) Given a level-1 encoding u, we have $u = \left[dy + r_1 x_1 + r_2 x_2\right]_q$, where d is secret level-0 encoding, and r_1, r_2 random noise elements.

Compute special decoding

$$v = \left[p_{zt} u y^{\kappa-2} x_1\right]_q = dy_{(1)} + r_1 x_{(1)} + r_2 x_{(2)}.$$

Since v is not reduced modulo q, we compute

$$v \mod y_{(1)} = r_1 x_{(1)} \mod y_{(1)} + r_2 x_{(2)} \mod y_{(1)}.$$

(3) Given $v \mod y_{(1)}$ and $\{x_{(1)} \mod y_{(1)}, x_{(2)} \mod y_{(1)}\}$, we get $v' = v \mod y_{(1)} \in \langle x_{(1)}, x_{(2)}\rangle$ such that $(v - v') \mod y_{(1)} = 0$. Let $v' = r_1' x_{(1)} + r_2' x_{(2)}$.

(4) Compute $d_{(0)} = (v - v')/y_{(1)}$ over $\mathbb{K} = \mathbb{R}[X]/\langle x^n + 1\rangle$ such that the quotient $d_{(0)} \in R$. By arranging, we obtain

$$d_{(0)} = \frac{v - v'}{y_{(1)}} = d + \frac{(r_1 - r_1')c_1 g + (r_1 - r_2')c_2 g}{1 + rg}$$

Again since g and $1+rg$ are co-prime, we get $d - d_{(0)} \in \langle g \rangle$. Thus, $d_{(0)}$ is an equivalent level-0 encoding of d. Although $\|d_{(0)}\|$ is not small, Hu and Jia [28] controlled the size of $d_{(0)}$ by using $x_{(i)} \in \langle g \rangle$.

Step 2: Compute an equivalent product of several level-0 encodings

Let $d_{(k)}$ be the level-0 encoding included by level-1 encoding u_k, $d_{(k,0)}$ an equivalent encoding of $d_{(k)}$. Then $\prod_{k=1}^{\kappa+1} d_{(k,0)}$ is an equivalent secret of $\prod_{k=1}^{\kappa+1} d_{(k)}$ such that $\prod_{k=1}^{\kappa+1} d_{(k,0)} - \prod_{k=1}^{\kappa+1} d_{(k)} \in \langle g \rangle$.

Step 3: Find the shared secret key of an equivalent product level-0 encoding

Let $\eta = \prod_{k=1}^{\kappa+1} d_{(k,0)}$. We compute $\eta' = y_{(1)}\eta \mod x_{(1)}$, and $\eta'' = [yx_1\eta']_q$.

2. Non-applicabiltiy of Hu-Jia Attack

(1) Let Par be the public parameters of our construction, i.e.

$$\text{Par} = \{q, \{\mathbf{Y}_i, \mathbf{X}_i, \mathbf{P}_{zt,i}\}_{i\in[\tau]}, \mathbf{T}^*, \mathbf{S}^*\}$$

For convenience, we let $\mathbf{P}_{zt} = \left[\mathbf{T}Rot\left(\frac{z^\kappa h_1}{g}\right)\mathbf{S}\right]_q$, $y_i = a_i g + e_i$, $x_j = c_j g$.

Similarly, we generate special decodings $\text{S} = \{\{\mathbf{Y}_{(i)}\}_{i\in[\tau]}, \{\mathbf{X}_{(j)}\}_{j\in[\tau]}\}$ as follows:

$$\mathbf{Y}_{(i)} = \left[\mathbf{T}^* \cdot (\mathbf{Y}_1^{\kappa-2}\mathbf{Y}_i\mathbf{X}_1) \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q = \mathbf{T}_1 Rot(y_1^{\kappa-2} y_i c_1 h_1)\mathbf{S}_1,$$
$$\mathbf{X}_{(j)} = \left[\mathbf{T}^* \cdot (\mathbf{Y}_1^{\kappa-2}\mathbf{X}_j\mathbf{X}_1) \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q = \mathbf{T}_1 Rot(y_1^{\kappa-2} x_j c_1 h_1)\mathbf{S}_1,$$

where $\mathbf{Y}_{(i)}$, $\mathbf{X}_{(j)}$ are not reduced modulo q.

(2) Given a level-1 encoding $\mathbf{U} = \left[\sum_{i=1}^{\tau} d_i \cdot \mathbf{Y}_i + \sum_{j=1}^{\tau} r_j \cdot \mathbf{X}_j\right]_q$, we compute special decoding

$$\mathbf{V} = \left[\mathbf{T}^* \cdot (\mathbf{U}\mathbf{Y}_1^{\kappa-2}\mathbf{X}_1) \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*\right]_q = \sum_{i=1}^{\tau} d_i \cdot \mathbf{Y}_{(i)} + \sum_{j=1}^{\tau} r_j \cdot \mathbf{X}_{(j)}.$$

Since $\mathbf{V}$ is not reduced modulo q and $\mathbf{V} \in \mathbb{Z}^{k_1 \times k_2}$, $\mathbf{V}$ belongs to the space spanned by $k = k_1 \times k_2$ elements in S.

On the one hand, we cannot efficiently find k elements in S such that $\mathbf{V} = \sum_{t=1}^{\tau_1} d_{i_t} \cdot \mathbf{Y}_{(i_t)} + \sum_{t=1}^{k-\tau_1} r_{j_t} \cdot \mathbf{X}_{(j_t)}$, and d_{i_t}, r_{j_t} are small integers. In fact, there sometimes does not exist k elements in S satisfying to the condition that d_{i_t}, r_{j_t} are small integers.

On the other hand, we cannot efficiently solve $\mathbf{V} = \sum_{t=1}^{\tau_1} d_{i_t} \cdot \mathbf{Y}_{(i_t)} + \sum_{t=1}^{k-\tau_1} r_{j_t} \cdot \mathbf{X}_{(j_t)}$ such that d_{i_t}, r_{j_t} are small integers if we choose $k = 2\tau$ elements in S to guarantee that there are small integers d_{i_t}, r_{j_t}.

The integers d_{i_t}, r_{j_t}, are required to be small since $\mathbf{V}_i = \mathbf{T}_1 Rot(r_i)\mathbf{S}_1$ cannot be directly multiplied to derive the product of some equivalent secret keys as that in [28]. We need to use d_{i_t}, r_{j_t} to generate the zero-testing parameter $\mathbf{P}_{zt}$ corresponding to the level-1 encoding $\mathbf{U}$.

In short, we cannot find an equivalent level-0 encoding encoded by $\mathbf{U}$. Thus, the Hu-Jia attack is prevented in our construction.

5 Applications

In this section, we describe two applications using our construction, the MPKE protocol and the instance of witness encryption.

5.1 MPKE protocol

The MPKE protocol consists of Setup, Publish, and KeyGen as follows:

Setup$(1^\lambda, 1^N)$:

Output Par $\leftarrow$ InstGen$(1^\lambda, 1^\kappa)$ as the public parameters.

Publish(Par, j):

The j-th party samples $\mathbf{d}_j \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, $\mathbf{r}_j \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, publishes the public key $\mathbf{U}_j = \left[\sum_{i=1}^{\tau}(d_{j,i} \cdot \mathbf{Y}_i) + \sum_{i=1}^{\tau}(r_{j,i} \cdot \mathbf{X}_i)\right]_q$ and remains $\mathbf{d}_j$ as the secret key.

KenGen$(\text{Par}, j, \mathbf{d}_j, \{\mathbf{U}_k\}_{k\neq j})$:

The j-th party computes $\mathbf{C}_j = \prod_{k\neq j} \mathbf{U}_k$, $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} d_{j,i}\mathbf{P}_{zt,i}$, and extracts the common secret key $sk_j = \text{Ext}(\text{Par}, \mathbf{C}_j, \mathbf{d}_j) = \text{MSB}\big([\mathbf{T}^* \cdot \mathbf{C}_j \cdot \mathbf{P}_{zt} \cdot \mathbf{S}^*]_q\big)$.

Theorem 5.1. Suppose the ext-GCDH/ext-GDDH defined in Section 4.1 is hard, then our construction is one round multipartite Diffie-Hellman key exchange protocol.

5.2 Witness Encryption

5.2.1 Construction

Garg, Gentry, Sahai, and Waters [23] constructed an instance of witness encryption based on the NP-complete 3-exact cover problem and the GGH map. However, Hu and Jia [28] have broken the GGH-based WE. In this section, we present a new construction of WE based our new multilinear map.

3-Exact Cover Problem [22]. Given a collection Set of subsets $T_1, T_2, \cdots, T_\pi$ of $[K] = \{1, 2, \cdots, K\}$ such that $K = 3\xi$ and $|T_i| = 3$, find a 3-exact cover of $[K]$. For an instance of witness encryption, the public key is a collection Set and the public parameters Par in our construction, the secret key is a hidden 3-exact cover of $[K]$.

Encrypt$(1^\lambda, \text{Par}, M)$:

(1) For $k \in [K]$, sample $\mathbf{d}_k \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, $\mathbf{r}_k \leftarrow D_{\mathbb{Z}^\tau,\alpha}$ and generate level-1 encodings $\mathbf{U}_k = \left[\sum_{i=1}^{\tau}\left(d_{k,i}\mathbf{Y}_i + r_{k,i} \cdot \mathbf{X}_i\right)\right]_q$.

(2) Compute $\mathbf{U} = \left[\prod_{k=1}^{K} \mathbf{U}_k\right]_q$ and $sk = \text{Ext}(\text{Par}, \mathbf{U}, \{1, 0, \cdots, 0\})$, and encrypt a message M into ciphertext C.

(3) For each subset $T_j = \{j_1, j_2, j_3\}$, sample $r_{T_j} \leftarrow D_{\mathbb{Z}^\tau,\alpha}$, and generate a level-3 encoding $\mathbf{U}_{T_j} = \left[\mathbf{U}_{j_1}\mathbf{U}_{j_2}\mathbf{U}_{j_3} + \sum_{i=1}^{\tau} r_{T_j,i}\mathbf{X}_i^3\right]_q$.

(4) Output the ciphertext C and all level-3 encodings $E = (\mathbf{U}_{T_j}, T_j \in \text{Set})$.

Decrypt(C, E, W):

(1) Given C, E and a witness set W, compute $\mathbf{U} = \left[\prod_{T_j \in W} \mathbf{U}_{T_j}\right]_q$.

(2) Generate $sk = \mathrm{Ext}(\mathrm{Par}, \mathbf{U}, \{1, 0, \cdots, 0\})$, and decrypt C to a message M.

Similar to [23], the security of our construction depends on the hardness assumption of the Decision Graded Encoding No-Exact-Cover.

Theorem 5.2. Suppose that the Decision Graded Encoding No-Exact-Cover is hard. Then our construction is a witness encryption scheme.

5.2.2 Hu-Jia Attacks.

(1) The Hu-Jia attack [28] is prevented in our new construction. One cannot obtain an equivalent secret key according to the analysis in 4.2.2. As a result, one cannot get an equivalent secret key using a combined 3-exact cover.

(2) The Hu-Jia attack [29] is thwarted in our new construction. Since Gu map-1 [24] uses hidden randomizers, in some sense one merely can generate a deterministically level-3 encoding. As a result, one can compute $\mathbf{U}_{T_t} = \left[\mathbf{U}_{T_j}\mathbf{U}_{T_k}(\mathbf{U}_{T_l})^{-1}\right]_q$ if $T_t = T_j \cup T_k - T_l$. Thus, one can generate a combined 3-exact cover, and correctly compute a secret level-K encoding. However, since $\mathbf{U}_{T_j} = \left[\mathbf{U}_{j_1}\mathbf{U}_{j_2}\mathbf{U}_{j_3} + \sum_{i=1}^{\tau} r_{T_j,i}\mathbf{X}_i^3\right]_q$ is a level-3 encoding in our new construction, one cannot obtain $\mathbf{U}_{T_t} = \left[\mathbf{U}_{T_j}\mathbf{U}_{T_k}(\mathbf{U}_{T_l})^{-1}\right]_q$ when $T_t = T_j \cup T_k - T_l$. This is because our construction contains the level-1 encodings $\mathbf{X}_i$ of zero.

6 Conclusions

In this paper, we describe a new variant of GGH13 via secret ring, which supports all applications for public tools of encoding in GGH13, such MPKE and WE. However, the security of our construction depends upon new hardness assumption, which cannot be reduced to classical hardness problem, such as LWE or SVP.

Acknowledgments. This work was supported in part by the Natural Science Foundation of China under Grant 61672270.

References

1. M. Albrecht, S. Bai, and L. Ducas. A subfield lattice attack on overstretched NTRU assumptions Cryptanalysis of some FHE and Graded Encoding Schemes. CRYPTO 2016, LNCS 9814, pp.153-178. http://eprint.iacr.org/2016/127.
2. Z. Brakerskiy, C. Gentry, S. Halevi, T. Lepoint, A. Sahai, M. Tibouchi. Cryptanalysis of the Quadratic Zero-Testing of GGH. http://eprint.iacr.org/2015/845.

3. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 324:71-90, 2003.
4. D. Boneh, D. J. Wu, and J. Zimmerman. Immunizing multilinear maps against zeroizing attacks. http://eprint.iacr.org/2014/930.
5. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. CRYPTO 2014, LNCS 8616, pp. 480-499.
6. R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings, EUROCRYPT 2016, LNCS 9666, pp. 559-585.
7. J. H. Cheon, K. Han, C. Lee, H. Ryu, D. Stehle. Cryptanalysis of the Multilinear Map over the Integers. http://eprint.iacr.org/2014/906.
8. J. H. Cheon, C. Lee. Cryptanalysis of the multilinear map on the ideal lattices. http://eprint.iacr.org/2015/461.
9. J. S. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi. Cryptanalysis of GGH15 Multilinear Maps. CRYPTO 2016, LNCS 9815, pp. 607-628.
10. J. H. Cheon, C. Lee, H. Ryu. Cryptanalysis of the New CLT Multilinear Maps. http://eprint.iacr.org/2015/934.
11. J. S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. CRYPTO 2013, LNCS 8042, pp. 476-493.
12. J. S. Coron, T. Lepoint, and M. Tibouchi. Cryptanalysis of two candidate fixes of multilinear maps over the integers. http://eprint.iacr.org/2014/975.
13. J. S. Coron, T. Lepoint, and M. Tibouchi. New Multilinear Maps over the Integers. http://eprint.iacr.org/2015/162.
14. Y. Chen and P. Q. Nguyen. BKZ 2.0 Better Lattice Security Estimates, ASIACRYPT 2011, LNCS 7073, pp. 1-20.
15. S. Garg. Candidate Multilinear Maps [PhD thesis]. University of California, Los Angeles, 2013.
16. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. EUROCRYPT 2013, LNCS 7881, pp. 1-17.
17. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. FOCS 2013, pp.40-49.
18. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps, CRYPTO (2) 2013, LNCS 8043, 479-499.
19. C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. TCC 2015, Part II, LNCS 9015, pp. 498-527.
20. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. http://eprint.iacr.org/2014/666.
21. C. Gentry, S. Halevi, H. K. Majiy, A. Sahaiz. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. http://eprint.iacr.org/2014/929.
22. O. Goldreich. Computational Complexity: a Conceptual Perspective. Cambridge University Press, New York, NY, USA, 1 edition, 2008.
23. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. STOC 2013, pp. 467-476.
24. Chunsheng Gu. Multilinear Maps Using Ideal Lattices without Encodings of Zero. http://eprint.iacr.org/2015/023.
25. Chunsheng Gu. An Improved Multilinear Map and its Applications. International Journal of Information Technology and Web Engineering, 2015, 10(3), pp. 64-81.
26. S. Halevi. The state of cryptographic multilinear maps, 2015. Invited Talk of CRYPTO 2015.

27. S. Halevi. Graded Encoding, Variations on a Scheme, http://eprint.iacr.org/2015/866.
28. Yupu Hu and Huiwen Jia. Cryptanalysis of GGH Map. http://eprint.iacr.org/2015/301.
29. Yupu Hu and Huiwen Jia. A Comment on Gu Map-1. http://eprint.iacr.org/2015/448.
30. Yupu Hu and Huiwen Jia. An Optimization of Gu Map-1. http://eprint.iacr.org/2015/453.
31. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. ANTS 1998, LNCS 1423, pp. 267-288.
32. H.W. Lenstra, A.K. Lenstra and L. Lovasz. Factoring polynomials with rational coefficients. Mathematische Annalen, 1982, 261(4): 515-534.
33. V. Lyubashevsky, D. Micciancio. Generalized Compact Knapsacks Are Collision Resistant, ICALP 2006: Automata, Languages and Programming, LNCS 4052, pp. 144-155. The full version of this extended abstract appears in ECCC TR05-142.
34. A. Langlois, D. Stehl, and R. Steinfeld, GGHLite: More Efficient Multilinear Maps from Ideal Lattices, EUROCRYPT 2014, LNCS 8441, 2014, pp. 239-256.
35. B. Minaud and P. Fouque. Cryptanalysis of the New Multilinear Map Over the integers, http://eprint.iacr.org/2015/941.
36. Micciancio D and Regev O. Worst-case to average-case reductions based on Gaussian measures. SIAM Journal on Computing, 2007, 37(1): 267-302
37. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices, EUROCRYPT 2011, LNCS 6632, pp. 27-47.
38. J. Zimmerman. How to obfuscate programs directly. EUROCRYPT 2015, Part II, LNCS 9057, pp. 439-467.