Multilinear Maps Using Ideal Lattices without Encodings of Zero

Gu Chunsheng
School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China
E-mail: chunsheng_gu@163.com

January 11, 2015

Abstract. Recently, Garg, Gentry and Halevi (GGH) described the first candidate multilinear maps using ideal lattices. However, there exists zeroizing attack in the GGH construction. We first describe an improved construction of multilinear maps from ideal lattices, by multiplying matrices on both sides of the level-1 encoding of non-zero element. The security of our construction depends upon new hardness assumption, which is seemingly closely related to hardness problems of lattices. Then, we describe an asymmetric construction to avoid any nontrivial encoding of zero. Using our constructions over polynomial ring instead of integer ring, we implement one-round multipartite Diffie-Hellman key exchange protocol to decrease the public parameter size.
Keywords. Multilinear maps, Ideal Lattices, Diffie-Hellman key exchange, Zeroizing attack

1 Introduction

Boneh and Silverberg [BS03] first introduced the notion of multilinear maps, which are an extension of bilinear maps. There exist many applications on bilinear maps, such as [SOK00, Jou00, BF03, Sma03] and multilinear maps [BS03, RS09, PTT10, Rot13]. However, different from bilinear maps, which come from pairing of elliptic curves, constructing multilinear maps is a long-standing open problem. Recently, Garg, Gentry, and Halevi (GGH) described the first plausible construction of multilinear maps that use ideal lattices [GGH13]. Their multilinear maps, whose encodings were randomized with noise and bounded with a fixed maximum degree, were different from the ideal multilinear maps defined by Boneh and Silverberg. Following the basic framework of GGH, Coron, Lepoint, and Tibouchi [CLT13] (CLT) described a new and relatively practical construction that works over integers instead of ideal lattices, and is implemented using heuristic optimization techniques. However, Cheon, Han, Lee, Ryu, and Stehle had broken the CLT construction using level-1 encodings of zero. To avoid this zeroizing attack for CLT, Garg, Gentry, Halevi and Zhandry [GGH+14], and Boneh, Wu and Zimmerman [BWZ14] proposed two candidate fixes of multilinear maps over the integers. However, Coron, Lepoint, and Tibouchi showed that two candidate fixes of CLT can be defeated in polynomial time using extensions of the Cheon et al.'s attack. Moreover, to improve the efficiency of GGH, Langlois, Stehlé and Steinfeld[LSS14] constructed GGHLite, in which the re-randomization process of GGH was reanalyzed by applying the Rényi divergence. Recently, Hiromasa, Abe and Okamoto [HAO14] described a new construction of multilinear maps from fully homomorphic encryption (FHE) proposed by Gentry, Sahai, and Waters (GSW) [GSW13b]. However, the security of the construction [HAO14] is not reduced to LWE, although the security of the GSW's FHE is based on the learning with erros (LWE) assumption. At the same time, Gentry, Gorbunov and Halevi [GGH14] described a construction of graph-induced multilinear maps from lattices using approximate eigenvector, which encodes LWE samples in short square matices of higher dimensions. However, the security of the construction [GGH14] is also not reduced to LWE. Moreover, the efficiency of the constructions based on LWE is lower than previous schemes. Since the GGH construction is more efficient than other schemes, we will focus on the improvement of GGH in this paper.

We first recall the GGH construction of multilinear maps. GGH works in the polynomial rings $R = \mathbb{Z}[x]/ < x^n + 1 >$ and $R_q = R / qR$. GGH chooses a secret short $\mathbf{g} \in R$, and a secret

random element $\mathbf{z} \in R_q$. $I = \langle \mathbf{g} \rangle \subset R$ is the principal ideal generated by $\mathbf{g}$. Plaintexts are cosets of R/I. To encode a coset $e_I = \mathbf{e} + I$, set $[\mathbf{c}/\mathbf{z}]_q$ with short $\mathbf{c} \in e_I$ as the encoding of e_I. Because $\mathbf{g}, \mathbf{z}$ are hidden in GGH, the public parameters of GGH gave the encoding $\mathbf{y}$ of the coset $1 + I$. So, the encoding of e_I is computed as $[\mathbf{e} \cdot \mathbf{y}]_q$. The GGH construction is a graded encoding scheme, that is, $\mathbf{e}$ is a level-0 encoding, $[\mathbf{e} \cdot \mathbf{y}]_q$ a level-1 encoding, and $\left[\mathbf{e} \cdot \mathbf{y}^i\right]_q$ a level-i encoding. It is easy to verify that encodings can both be added and multiplied if the numerator norm remains smaller than q. For a level-κ encoding $\mathbf{u}$, the GGH can determine whether $\mathbf{u}$ is the encoding of zero using the zero-testing parameter $\mathbf{p}_{zt}$. This defines a degree-κ multilinear map for level-1 encodings.

Our results. Our main contribution is to describe a construction of multilinear maps using ideal lattices without encoding of zero. Our construction improves GGH in two aspects. First, we modify the zero-testing parameter of GGH. The public parameters of our construction only give some pairs of the encoding of non-zero element and the zero-testing parameter corresponding to this non-zero element. Second, we multiply short matrices on both sides of the public parameters. Unlike the GGH construction, our construction does not give level-1 encodings of "1" and "0", and cannot generate level-1 of given level-0 encoding. Moreover, our construction only generates a level-1 encoding for a hidden level-0 encoding, and the encoding in a sense is a deterministic encoding without re-randomization process.

Our second contribution is to describe an asymmetric variant of our symmetric version. In our symmetric construction, one can still compute hidden level-κ encoding of zero element even if our public parameters do not give level-1 encodings of zero elements. This is because one can obtain level-κ encoding of zero by cross-multiplying pairs of the encodings and the zero-testing parameters in the public parameters. To avoid this case, our asymmetric variant will not support multiplying the encoding by the zero-testing parameter with the same index set. Thus, one cannot generate any level encoding of zero in our asymmetric version. Namely, unlike GGH, there exist no easily computable quantities in our asymmetric construction.

Our third contribution is to describe the commutative variant of our constructions using polynomial rings instead of the ring of integers. To guarantee the security of our construction, we must make sure that the dimension of matrix in our construction is large enough. As a result, our construction is less practical than previous schemes. Thus, we use matrices of small dimension, and large degree polynomial ring to improve the efficiency of our constructions.

Our final contribution is to optimize and implement one-round multipartite Diffie-Hellman key exchange protocol using our commutative variant of multilinear maps. Experimental results demonstrate that our construction of multilinear maps from ideal lattices is practical.

Organization. We recall some background on multilinear maps in Section 2. In Section 3, we describe our symmetric construction, and in Section 4 we provide asymmetric construction. In Section 5, we construct the commutative variant of our constructions. In Section 6, we optimize and implement one round multipartite Diffie-Hellman key exchange protocol. Finally, we draw conclusion and open problem for this paper.

2 Preliminaries

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 a power of 2. Notation $[\![n]\!]$ denotes the set $\{1,2,\cdots,n\}$, and $[a]_q$ the absolute minimum residual system $[a]_q = a \bmod q \in (-q/2, q/2]$. Vectors and matrices are denoted in bold, such as $\mathbf{a},\mathbf{b},\mathbf{c}$ and $\mathbf{A},\mathbf{B},\mathbf{C}$. Let $\mathbf{I}$ be the identity matrix. The j-th entry of $\mathbf{a}$ is denoted as a_j, the element of the i-th row and j-th colomn of $\mathbf{A}$ is denoted as $A_{i,j}$ (or $A[i,j]$). Notation $\|\mathbf{a}\|_\infty$ ($\|\mathbf{a}\|$ for short) denotes the infinity norm of $\mathbf{a}$. The polynomial ring $\mathbb{Z}[X]/<x^n+1>$ is denoted by R, and $\mathbb{Z}_q[X]/<x^n+1>$ by R_q. The elements in R and R_q are denoted in bold as well. Similarly, notation $[\mathbf{a}]_q$ denotes each entry (or each coefficient) $a_i \in (-p/2, p/2]$ of $\mathbf{a}$.

2.2 Lattices and Ideal Lattices

An *n*-dimension 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 $L = \{\mathbf{Bz} : \mathbf{z} \in Z^n\}$. We say that $\mathbf{B}$ spans L if $\mathbf{B}$ is a basis for L. Given a basis $\mathbf{B}$ of L, we define $P(\mathbf{B}) = \{\mathbf{Bz} \mid \mathbf{z} \in \mathbb{R}^n, \forall i : -1/2 \le z_i < 1/2\}$ as the parallelization corresponding to $\mathbf{B}$. Let $\det(\mathbf{B})$ denote the determinant of $\mathbf{B}$.

Given $\mathbf{g} \in R$, let $I = <\mathbf{g}>$ be the principal ideal in R generated by $\mathbf{g}$, whose $\mathbb{Z}$-basis is $Rot(\mathbf{g}) = (\mathbf{g}, x \cdot \mathbf{g}, ..., x^{n-1} \cdot \mathbf{g})$.

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

2.3 Multilinear Maps

Definition 2.1 (Multilinear Map [BS03]). For $\kappa+1$ cyclic groups $G_1,...,G_\kappa,G_T$ of the same order q, a κ-multilinear map $e : G_1 \times \cdots \times G_\kappa \to G_T$ 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

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

(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 $e(g_1,\cdots,g_\kappa)$ is a generator of G_T.

Definition 2.2 (κ-Graded Encoding System [GGH13]). A κ-graded encoding system over R is a set system of $S = \{S_j^{(\alpha)} \subset R : \alpha \in R, j \in [\![\kappa]\!]\}$ with the following properties:

(1) For every index $j \in [\![\kappa]\!]$, the sets $\left\{S_j^{(\alpha)} : \alpha \in R\right\}$ are disjoint.

(2) Binary operations '$+$' and '$-$' exist, such that every α_1, α_2, every index $j \in [\![\kappa]\!]$, and every $u_1 \in S_j^{(\alpha_1)}$ and $u_2 \in S_j^{(\alpha_2)}$ hold that $u_1 + u_2 \in S_j^{(\alpha_1+\alpha_2)}$ and $u_1 - u_2 \in S_j^{(\alpha_1-\alpha_2)}$, where $\alpha_1 + \alpha_2$ and $\alpha_1 - \alpha_2$ are the addition and subtraction operations in R respectively.

(3) Binary operation '$\times$' exists, such that every α_1, α_2, every index $j_1, j_2 \in [\![\kappa]\!]$ with $j_1 + j_2 \le \kappa$, and every $u_1 \in S_{j_1}^{(\alpha_1)}$ and $u_2 \in S_{j_2}^{(\alpha_2)}$ hold that $u_1 \times u_2 \in S_{j_1+j_2}^{(\alpha_1\times\alpha_2)}$, where $\alpha_1 \times \alpha_2$ is the multiplication operation in R and $j_1 + j_2$ is the integer addition.

3 Construction of symmetric multilinear maps

In this section, we first describe the symmetric construction of multilinear maps. Then we give new hardness assumption and some known cryptanalysis for our construction.

Setting the parameters. Because our construction uses the GGH construction as the basic component, our parameter setting is set as that of GGH to conveniently describe and compare. Let λ be the security parameter, κ the multilinearity level, n the dimension of elements of R. Concrete parameters are set as $\sigma = \sqrt{\lambda n}$, $\sigma' = \lambda n^{1.5}$, $\sigma^* = 2^\lambda$, $q \ge 2^{8\kappa\lambda} n^{O(\kappa)}$, $n > \widetilde{O}(\kappa\lambda^2)$, $\tau = O(n^2)$.

3.1 Construction

The starting point of our construction is to remove level-1 encodings of zero in the public parameters. We modify the zero-testing parameter of GGH so that the public parameters in our construction only include some pairs of the level-1 encoding of non-zero element and the zero-testing parameter corresponding to this non-zero element. Moreover, we multiply both sides of these encodings and zero-testing parameters by random short matrices. Our construction is as follows:

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

(1) Choose a prime $q \ge 2^{8\kappa\lambda} n^{O(\kappa)}$;

(2) Choose an element $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$ in R so that $\left\|\mathbf{g}^{-1}\right\| \le n^2$;

(3) Choose elements $\mathbf{a}_i, \mathbf{e}_i \leftarrow D_{\mathbb{Z}^n,\sigma'}, \mathbf{b}_i \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}, i \in [\![\tau]\!]$ in R;

(4) Choose a random element $\mathbf{z} \leftarrow R_q$ so that $\mathbf{z}^{-1} \in R_q$;

(5) Choose two random matrices $\mathbf{T} \leftarrow D_{\mathbb{Z}^{n\times n},\sigma}$ and $\mathbf{S} \leftarrow D_{\mathbb{Z}^{n\times n},\sigma}$ so that $\mathbf{T}^{-1}, \mathbf{S}^{-1} \in \mathbb{Z}_q^{n\times n}$;

(6) Set $\mathbf{Y}_i = \left[\mathbf{T} Rot\left(\frac{\mathbf{a}_i\mathbf{g} + \mathbf{e}_i}{\mathbf{z}}\right)\mathbf{T}^{-1}\right]_q$ and $\mathbf{P}_{zt,i} = \left[\mathbf{T} Rot\left(\frac{\mathbf{z}^\kappa(\mathbf{b}_i\mathbf{g} + \mathbf{e}_i)}{\mathbf{g}}\right)\mathbf{S}\right]_q, i \in [\![\tau]\!]$;

(7) Output the public parameter $\text{par} = \left\{q, \left\{\mathbf{Y}_i, \mathbf{P}_{zt,i}\right\}, i \in [\![\tau]\!]\right\}$.

According to [GGH13], $\mathbf{g} \in R, \mathbf{z} \in R_q, \mathbf{a}_i, \mathbf{b}_i, \mathbf{e}_i \in R$ can be efficiently sampled. It is easy to see that $\mathbf{T}, \mathbf{S} \in \mathbb{Z}^{n\times n}$ can be sampled. This is because that if $\det(\mathbf{T}), \det(\mathbf{S})$ are not divisible by

q, then $\mathbf{T}^{-1}, \mathbf{S}^{-1} \in \mathbb{Z}_q^{n\times n}$. Without loss of generality, assume that $\det(\mathbf{T}), \det(\mathbf{S})$ are uniform over $\mathbb{Z}_q$. Thus, the probability that $\mathbf{T}, \mathbf{S}$ are invertible is about $1-O(1/q)$.

Generating level-1 encoding: $\mathbf{U} \leftarrow \mathrm{enc}(\mathrm{par}, 1, \mathbf{d})$.

Given a random vector $\mathbf{d} \leftarrow D_{\mathbb{Z}^\tau, \sigma^*}$, then $\mathbf{U} = \left[\sum_{i=1}^{\tau} (d_i \cdot \mathbf{Y}_i)\right]_q$ is the level-1 encoding of hidden level-0 encoding $\mathbf{e} = \sum_{i=1}^{\tau} (d_i \cdot \mathbf{e}_i)$.

Because both sides of $\mathbf{Y}_i$ are multiplied by matrices $\mathbf{T}, \mathbf{T}^{-1}$ respectively, $\mathbf{Y}_i$ multiplied by the scalar d_i can be commutative with $\mathbf{T}$ to obtain $d_i \cdot Rot(\frac{\mathbf{a}_i\mathbf{g}+\mathbf{e}_i}{\mathbf{z}})$. Thus, we have

$$\mathbf{U} = \left[\sum_{i=1}^{\tau} (d_i \cdot \mathbf{Y}_i)\right]_q = \left[\mathbf{T} Rot(\frac{\sum_{i=1}^{\tau} d_i(\mathbf{a}_i\mathbf{g}+\mathbf{e}_i)}{\mathbf{z}})\mathbf{T}^{-1}\right]_q = \left[\mathbf{T} Rot(\frac{\mathbf{a}\mathbf{g}+\mathbf{e}}{\mathbf{z}})\mathbf{T}^{-1}\right]_q \quad , \quad \text{where}$$

$\mathbf{a} = \sum_{i=1}^{\tau} (d_i \cdot \mathbf{a}_i)$ and $\mathbf{e} = \sum_{i=1}^{\tau} (d_i \cdot \mathbf{e}_i)$. That is, $\mathbf{U}$ is the level-1 encoding of hidden plaintext element $\mathbf{e}$.

In our construction, one cannot directly generate the level-1 encoding of a given level-0 encoding since one does not know the level-0 encoding $\mathbf{e}_i$ encoded by $\mathbf{Y}_i$. Although one can obtain a level-j encoding $\mathbf{U}_j = (\mathbf{Y}_i)^j$, but one cannot know the level-0 element $(\mathbf{e}_i)^j$ encoded by $\mathbf{U}_j$. This point is different from the GGH construction.

Adding encodings: $\mathbf{U} \leftarrow \mathrm{add}(\mathrm{par}, j, \mathbf{U}_1, \cdots, \mathbf{U}_m)$.

Given m level-j encodings $\mathbf{U}_l$, their sum $\mathbf{U} = \left[\sum_{l=1}^{m} \mathbf{U}_l\right]_q$ is a level-j encoding.

Because the level-j encoding $\mathbf{U}_l$ is the form of $\mathbf{U}_l = \left[\mathbf{T} Rot(\frac{\mathbf{r}_l\mathbf{g}+\mathbf{e}_l}{\mathbf{z}^j})\mathbf{T}^{-1}\right]_q$, their sum

$$\mathbf{U} = \left[\sum_{l=1}^{m} \mathbf{U}_l\right]_q = \left[\mathbf{T} Rot(\frac{\sum_{l=1}^{m} (\mathbf{r}_l\mathbf{g}+\mathbf{e}_l)}{\mathbf{z}^j})\mathbf{T}^{-1}\right]_q = \left[\mathbf{T} Rot(\frac{\mathbf{r}\mathbf{g}+\mathbf{e}}{\mathbf{z}^j})\mathbf{T}^{-1}\right]_q \quad \text{is a level-} j$$

encoding, where $\mathbf{r} = \sum_{l=1}^{m} \mathbf{r}_l$ and $\mathbf{e} = \sum_{l=1}^{m} \mathbf{e}_l$.

Multiplying encodings: $\mathbf{U} \leftarrow \mathrm{mul}(\mathrm{par}, 1, \mathbf{U}_1, \cdots, \mathbf{U}_\kappa)$.

Given κ level-1 encodings $\mathbf{U}_j$, their product $\mathbf{U} = \left[\prod_{j=1}^{\kappa} \mathbf{U}_j\right]_q$ is a level-κ encoding.

Because the level-1 encoding $\mathbf{U}_j$ is the form of $\mathbf{U}_j = \left[\mathbf{T} Rot(\frac{\mathbf{r}_j\mathbf{g}+\mathbf{e}_j}{\mathbf{z}})\mathbf{T}^{-1}\right]_q$, the product of κ level-1 encodings is:

$$
\begin{aligned}
\mathbf{U} &= \left[\prod_{j=1}^{\kappa}\mathbf{U}_j\right]_q \\
&= \left[\prod_{j=1}^{\kappa}\mathbf{T}Rot(\frac{\mathbf{r}_j\mathbf{g}+\mathbf{e}_j}{\mathbf{z}})\mathbf{T}^{-1}\right]_q \\
&= \left[\mathbf{T}Rot(\frac{\prod_{j=1}^{\kappa}(\mathbf{r}_j\mathbf{g}+\mathbf{e}_j)}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\right]_q \quad \text{, where } \mathbf{e}=\prod_{j=1}^{\kappa}\mathbf{e}_j, \mathbf{r}=(\prod_{j=1}^{\kappa}(\mathbf{r}_j\mathbf{g}+\mathbf{e}_j)-\mathbf{e})/\mathbf{g}. \\
&= \left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\right]_q
\end{aligned}
$$

We use $\mathbf{T}\times\mathbf{T}^{-1}=\mathbf{I}$ in third equation, and denote the level-κ encoding $\mathbf{U}$ as the standard form in the final equation.

Zero testing: $\text{isZero}(\text{par},\mathbf{U})$.

To determine whether $\mathbf{U}=\left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\right]_q$ is a level-κ encoding of zero, $\mathbf{V}=\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q$ is computed in $\mathbb{Z}_q^{n\times n}$ and checked whether $\|\mathbf{V}\|$ is short:

$$
\text{isZero}(\text{par},\mathbf{U})=\begin{cases}1 & \text{if } \left\|\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q\right\| < q^{3/4} \\ 0 & \text{otherwise}\end{cases}
\text{, where } \mathbf{P}_{zt}=\sum_{i=1}^{\tau}r_i\mathbf{P}_{zt,i} \text{ and } \mathbf{r}\leftarrow D_{\mathbb{Z}^{\tau},\sigma}.
$$

Since $\mathbf{P}_{zt}=\sum_{i=1}^{\tau}r_i\mathbf{P}_{zt,i}=\left[\mathbf{T}Rot(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q$, where $\mathbf{b}=\sum_{i=1}^{\tau}(r_i\mathbf{b}_i)$ and $\mathbf{c}=\sum_{i=1}^{\tau}(r_i\mathbf{e}_i)$. If $\mathbf{U}$ is a level-κ encoding of zero element, namely $\mathbf{e}=\mathbf{0}\bmod I$, then we have

$$
\mathbf{V}=\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q=\left[\mathbf{T}Rot(\frac{\mathbf{rg}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\cdot\mathbf{T}Rot(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q=\left[\mathbf{T}Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q.
$$

For our choice of parameter, $\|\mathbf{rg}+\mathbf{e}\|=\|\mathbf{rg}\|\le q^{1/8}$ and $\|\mathbf{T}\|_{\infty}=\|\mathbf{S}\|_{\infty}\le\sqrt{n}\sigma$. Moreover, $\mathbf{V}$ is not reduced modulo q, that is $\left[\mathbf{V}\right]_q=\mathbf{V}$. Thus, we have

$$
\begin{aligned}
\|\mathbf{V}\| &= \left\|\left[\mathbf{T}Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q\right\| \\
&= \left\|\mathbf{T}Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right\| \\
&= n^2\cdot\|\mathbf{T}\|\|Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\|\|\mathbf{S}\| \\
&= n^3\cdot\sqrt{n}\sigma\|Rot(\mathbf{r})\|\|Rot(\mathbf{bg}+\mathbf{c})\|\sqrt{n}\sigma. \\
&= n^4\sigma^2\cdot\|\mathbf{rg}\|\cdot\|\mathbf{g}^{-1}\|\|Rot(\mathbf{bg}+\mathbf{c})\| \\
&= n^4\sigma^2\cdot q^{1/8}\cdot poly(n)\cdot q^{1/2}\cdot poly(n) \\
&< q^{3/4}
\end{aligned}
$$

If $\mathbf{U}$ is a level-κ encoding of non-zero element, namely $\mathbf{e}\neq\mathbf{0}\bmod I$. Then, we have

$$\mathbf{V} = \left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q = \left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\cdot\mathbf{T}Rot(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q = \left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{g}}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q .$$

By lemma 4 in [GGH13], we have $\left\|\left[Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{g}})\right]_q\right\| \approx q$. Thus, $\|\mathbf{V}\| \approx q$.

Extraction: $sk \leftarrow \mathrm{ext}(\mathrm{par}, \mathbf{U})$.

Given a level-κ encoding $\mathbf{U}$, $\mathbf{U}$ is multiplied by $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} w_i \mathbf{P}_{zt,i}$, where $\mathbf{w} \leftarrow D_{\mathbb{Z}^{\tau},\sigma}$ and $(\log q)/4 - \lambda$ most-significant bits of each of the $n \times n$ entries of $\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q$ is collected:

$$\mathrm{ext}(\mathrm{par}, \mathbf{U}) = \mathrm{Extract}(\mathrm{msb}(\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q)).$$

Because $\mathbf{P}_{zt} = \sum_{i=1}^{\tau} w_i \mathbf{P}_{zt,i} = \left[\mathbf{T}Rot(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q$, where $\mathbf{b}=\sum_{i=1}^{\tau} w_i \mathbf{b}_i$ and $\mathbf{c}=\sum_{i=1}^{\tau} w_i \mathbf{e}_i$. Assume $\mathbf{U} = \left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\right]_q$ such that $\|\mathbf{rg}+\mathbf{e}\| \le q^{1/8}$, then we have

$$\begin{aligned}
\mathbf{V} &= \left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q \\
&= \left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^{\kappa}})\mathbf{T}^{-1}\cdot\mathbf{T}Rot(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q \\
&= \left[\mathbf{T}Rot(\frac{(\mathbf{rg}+\mathbf{e})(\mathbf{bg}+\mathbf{c})}{\mathbf{g}})\mathbf{S}\right]_q \\
&= \left[\mathbf{T}Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q + \left[\mathbf{T}Rot(\frac{\mathbf{e}}{\mathbf{g}}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q
\end{aligned} .$$

For our parameter setting, $\left\|\left[\mathbf{T}Rot(\mathbf{r}(\mathbf{bg}+\mathbf{c}))\mathbf{S}\right]_q\right\| < q^{3/4}$. By Lemma 4 in [GGH13], we have $\left\|Rot(\frac{\mathbf{e}}{\mathbf{g}}(\mathbf{bg}+\mathbf{c}))\right\| \approx q$ for $\mathbf{e} \neq \mathbf{0} \bmod I$. Therefore, the extraction algorithm can correctly work.

Remark 3.1 (1) Different from the GGH construction, our construction cannot directly generate level-1 encoding of a given level-0 encoding, and can only generate level-1 encoding of hidden level-0 encoding $\mathbf{e}=\sum_{i=1}^{\tau}(d_i \cdot \mathbf{e}_i)$. Moreover, the level-1 encoding of our construction is deterministic, and it is no longer random and without re-randomization process. However, we do not find the necessity generating given level-0 encoding or known level-0 encoding in our construction.

(2) Choose $\tau = O(n^2)$ is to erase the structure of input encoding applying re-randomization process in [GGH13]. Although our construction is deterministic, the level-1 encoding process

generating hidden level-0 encoding is same as the re-randomization process of the GGH construction. The cost using large τ is that the public parameter size of our construction is bigger a n factor than that of GGH. We notice that $\tau > n$ is the lowest requirement, otherwise attacker can directly solve $\mathbf{d}$ applying linear equation system.

(3) When constructing multipartite key exchange using our symmetric construction, every participant can compute the zero testing parameter corresponding to the hidden $\mathbf{e}=\sum_{i=0}^{n-1}(d_i \cdot \mathbf{e}_i)$ encoded by $\mathbf{U}=\left[\sum_{i=0}^{n-1}(d_i \cdot \mathbf{Y}_i)\right]_q$, that is, the zero testing parameter corresponding to level-0 encoding $\mathbf{e}$ is

$$\begin{aligned}\mathbf{P}_{zt} &= \left[\sum_{i=0}^{n-1} d_i \cdot \mathbf{P}_{zt,i}\right]_q \\ &= \left[\mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}\left(\sum_{i=0}^{n-1} d_i \mathbf{b}_i \mathbf{g} + \sum_{i=0}^{n-1} d_i \mathbf{e}_i\right)}{\mathbf{g}}\right)\mathbf{S}\right]_q . \\ &= \left[\mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}(\mathbf{bg}+\mathbf{e})}{\mathbf{g}}\right)\mathbf{S}\right]_q\end{aligned}$$

(4) $\mathbf{P}_{zt,i}$ or their combination $\mathbf{P}_{zt}$ can be used as zero testing parameter. In addition, the zero testing parameter generated by random combination of $\mathbf{P}_{zt,i}$ can thwart invalid encoding attack for only one zero testing parameter.

(5) The matrices $\mathbf{T}, \mathbf{S}$ in our construction are to thwart adversary not only generating less than level-k encoding of zero from the public parameter, but also getting the basis of the secret principal ideal lattices in our construction. This is because $\mathbf{P}_{zt,i}$ cannot directly be multiplied. For arbitrary $i, j \in [\![\tau]\!]$, we have

$$\begin{aligned}\mathbf{P} &= \mathbf{P}_{zt,i} \times \mathbf{P}_{zt,j} \\ &= \left[\mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}(\mathbf{b}_i\mathbf{g}+\mathbf{e}_i)}{\mathbf{g}}\right)\mathbf{S}\right]_q \times \left[\mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}(\mathbf{b}_j\mathbf{g}+\mathbf{e}_j)}{\mathbf{g}}\right)\mathbf{S}\right]_q . \\ &= \left[\mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}(\mathbf{b}_i\mathbf{g}+\mathbf{e}_i)}{\mathbf{g}}\right)\mathbf{S} \cdot \mathbf{T}Rot\left(\frac{\mathbf{z}^{\kappa}(\mathbf{b}_j\mathbf{g}+\mathbf{e}_j)}{\mathbf{g}}\right)\mathbf{S}\right]_q\end{aligned}$$

Since matrix multiplication does not support commutative rule, the second numerator $\mathbf{z}^{\kappa}$ in $\mathbf{P}$ cannot be canceled by multiplying a level-2κ encoding. Therefore, we may sample $\mathbf{b}_i \leftarrow D_{\mathbb{Z}^n,\sigma'}$ and set $q \geq 2^{4\kappa\lambda} n^{O(\kappa)}$ to decrease by half the size of the public parameter. Moreover, using $\mathbf{z}^{\kappa}$

guarantees that $\mathbf{P}_{zt,i}$ can only be used as the zero-testing for a level-κ encoding.

3.2 Security

Similar to the previous constructions [GGH13, CLT13, LSS14], the security of our construction cannot be reduced to classic hardness assumptions. In [GGH13], the security of GGH is defined as the hardness assumptions of graded computational Diffie-Hellman (GCDH) and graded decisional Diffie-Hellman (GDDH). That is, given the public parameters and $\kappa+1$ level-1 encodings of random elements, it is unfeasible to generate a level-κ encoding of their product or distinguish it from random elements. Langlois, Stehlé and Steinfeld[9] introduced the hardness assumptions ext-GCDH/ext-GDD, which is variant of GCDH/GDDH defined in [GGH13]. The security of our construction relies on new hardness assumption ext-GCDH/ext-GDDH. In the following, we adaptively define the ext-GCDH/ext-GDDH in [LSS14] to our construction.

Consider the following security experiment:

(1) $\text{par} \leftarrow \text{InstGen}(1^{\lambda},1^{\kappa})$

(2) For $j=0$ to κ:

Sample $\mathbf{r}_j, \mathbf{w}_j \leftarrow D_{\mathbb{Z}^{\tau},\sigma^*}$;

Generate level-1 encoding of hidden $\mathbf{d}_j = \sum_{i=1}^{\tau} w_{j,i}\mathbf{e}_i$: $\mathbf{U}_j = \left[\sum_{i=1}^{\tau} w_{j,i}\mathbf{Y}_i\right]_q$.

(3) Compute $\mathbf{U}^* = \left[\prod_{j=1}^{\kappa} \mathbf{U}_j\right]_q$.

(4) Compute $\mathbf{V}_C = \mathbf{V}_D = \left[\mathbf{U}^*\mathbf{P}_{zt}\right]_q$, where $\mathbf{P}_{zt} = \left[\sum_{i=1}^{\tau} w_{0,i}\mathbf{P}_{zt,i}\right]_q$.

(5) Compute $\mathbf{V}_R = \left[\mathbf{U}^*\mathbf{P}_{zt\_rand}\right]_q$, where $\mathbf{P}_{zt\_rand} = \left[\sum_{i=1}^{\tau} r_{0,i}\mathbf{P}_{zt,i}\right]_q$.

Definition 3.2 (ext-GCDH/ext-GDDH). According to the above 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 $\mathbf{W} \in \mathbb{Z}_q^{n\times n}$ such that $\left\|\left[\mathbf{V}_C - \mathbf{W}\right]_q\right\|_{\infty} \le q^{3/4}$.

Level-κ extraction DDH (ext-GDDH): Given $\{\text{par}, \mathbf{U}_0, \cdots, \mathbf{U}_{\kappa}, \mathbf{V}\}$, distinguish between $D_{ext-GDDH} = \{\text{par}, \mathbf{U}_0, \cdots, \mathbf{U}_{\kappa}, \mathbf{V}_D\}$ and $D_{ext-RAND} = \{\text{par}, \mathbf{U}_0, \cdots, \mathbf{U}_{\kappa}, \mathbf{V}_R\}$.

In our construction, the ext-GCDH is harder than the ext-GDDH. This is because given $\mathbf{V} \in \{D_{ext-GDDH}, D_{ext-RAND}\}$, one can compute $\mathbf{W}$ using the oracle of solving ext-GCDH, and further determine $\mathbf{V}$.

It is easy to verify that breaking our construction is harder than breaking the GGH construction. If there exists an algorithm A which breaks our construction, then there exists an algorithm B using A, which breaks the GGH construction. This is because one can sample the matrices $\mathbf{T},\mathbf{S}$, generate the public parameters of our construction using the instance generation, and call A to solve the corresponding problem.

In the following, we will show that the matrices of both sides of the public parameters cannot be removed only using arithmetic operations.

Lemma 3.3 Given the public parameters $\text{par}=\left\{q,\left\{\mathbf{Y}_i,\mathbf{P}_{zt,i}\right\},i\in[\![\tau]\!]\right\}$ of our symmetric construction, using arithmetic operations cannot remove the matrices, which are multiplied on both sides of $\mathbf{Y}_i,\mathbf{P}_{zt,i}$.

Proof. (1) By the instance generation, both sides of $\mathbf{Y}_i,\mathbf{P}_{zt,i}$ are multiplied by matrices $\mathbf{T}$, $\mathbf{T}^{-1}$ and $\mathbf{T}$, $\mathbf{S}$, respectively. (2) Assume $\mathbf{X}_1,\mathbf{X}_2\in\left\{\mathbf{Y}_i,\mathbf{P}_{zt,i},i\in[\![\tau]\!]\right\}$ and $\mathbf{X}_1=\mathbf{T}_1\mathbf{X}_1^{'}\mathbf{S}_1,\mathbf{X}_2=\mathbf{T}_2\mathbf{X}_2^{'}\mathbf{S}_2$ with $\mathbf{X}_1^{'},\mathbf{X}_2^{'}$ generated by some principal ideal lattices. It is obvious that both sides of the results $\mathbf{X}_1+\mathbf{X}_2,\mathbf{X}_1-\mathbf{X}_2$ have the matrices if addition or subtraction operations can be supported. For multiplication, the left and right sides of $\mathbf{X}_1\times\mathbf{X}_2$ will have $\mathbf{T}_1$ and $\mathbf{S}_2$ respectively. Similarly, both sides of $\mathbf{X}_2\times\mathbf{X}_1,\mathbf{X}_1\times(\mathbf{X}_2)^{-1},(\mathbf{X}_1)^{-1}\times\mathbf{X}_2$ also have random matrices. (3) Using recursive method, we show that arbitrary arithmetic operations over $\mathbf{Y}_i,\mathbf{P}_{zt,i}$ cannot remove the matrices of both sides of generating result. □

3.3 Cryptanalysis

In this subsection, we describe easily computable quantities in our construction, and then analyze possible attacks for our construction using these quantities.

Easily computable quantities. Because $\mathbf{Y}_i,\mathbf{P}_{zt,i}$ encode the same level-0 encoding $\mathbf{e}_i$, for arbitrary $i,j,t\in[\![\tau]\!]$ with $i\neq j$, one can compute $\mathbf{V}_{i,j,t}$ as follows:

$$
\begin{aligned}
&\mathbf{V}_{i,j,t}\\
&=\left[\mathbf{Y}_t^{\kappa-1}(\mathbf{Y}_i\times\mathbf{P}_{zt,j}-\mathbf{Y}_j\times\mathbf{P}_{zt,i})\right]_q\\
&=\left[\mathbf{T}(Rot(\frac{\mathbf{a}_t\mathbf{g}+\mathbf{e}_t}{\mathbf{z}}))^{\kappa-1}\cdot(Rot(\frac{\mathbf{a}_i\mathbf{g}+\mathbf{e}_i}{\mathbf{z}}\times\frac{\mathbf{z}^{\kappa}(\mathbf{b}_j\mathbf{g}+\mathbf{e}_j)}{\mathbf{g}})-Rot(\frac{\mathbf{a}_j\mathbf{g}+\mathbf{e}_j}{\mathbf{z}}\times\frac{\mathbf{z}^{\kappa}(\mathbf{b}_i\mathbf{g}+\mathbf{e}_i)}{\mathbf{g}}))\mathbf{S}\right]_q\cdot\\
&=\left[\mathbf{T}(Rot(\mathbf{a}_t\mathbf{g}+\mathbf{e}_t))^{\kappa-1}\cdot Rot(\mathbf{a}_i\mathbf{b}_j\mathbf{g}+\mathbf{a}_i\mathbf{e}_j+\mathbf{b}_j\mathbf{e}_i-\mathbf{a}_j\mathbf{b}_i\mathbf{g}-\mathbf{a}_j\mathbf{e}_i-\mathbf{b}_i\mathbf{e}_j)\mathbf{S}\right]_q
\end{aligned}
$$

According to our parameter setting, it is easy to see that $\mathbf{V}_{i,j,t}$ is not reduced modulo q, namely $\left\lceil \mathbf{V}_{i,j,t} \right\rfloor_q = \mathbf{V}_{i,j,t}$. Thus, one can obtain many $\mathbf{V}_{i,j,t} \in \mathbb{Z}^{n\times n}$ using different combinations $i, j, t \in [\![\tau]\!]$. These $\mathbf{V}_{i,j,t}$'s have the form $\mathbf{V}_{i,j,t} = \mathbf{T}(Rot(\mathbf{r}_{i,j,t}\mathbf{g} + \mathbf{e}_{i,j,t}))\mathbf{S}$.

Compute the norm of ideal. By computing the determinant $\det(\mathbf{V}_{i,j,t})$ of $\mathbf{V}_{i,j,t}$, one can obtain the norm of the ideal $\mathbf{a}_t\mathbf{g} + \mathbf{e}_t$ using GCD algorithm. When knowing the norm p, one factors $x^n + 1 = \prod_{i=1}^{n}(x - \alpha_i) \bmod p$, and solves the generator of the principal ideal lattice $\langle \mathbf{a}_t\mathbf{g} + \mathbf{e}_t \rangle$ generated by two element (p, α_i). If $\mathbf{a}_t\mathbf{g} + \mathbf{e}_t$ can be solved, then our construction is broken. This is because given $\mathbf{a}_1\mathbf{g} + \mathbf{e}_1$ and $\mathbf{a}_2\mathbf{g} + \mathbf{e}_2$, one solves the matrix $\mathbf{T}$ by $\mathbf{V}_{i,j,1}(\mathbf{V}_{i,j,2})^{-1} = \mathbf{T}((\mathbf{a}_1\mathbf{g} + \mathbf{e}_1)(\mathbf{a}_2\mathbf{g} + \mathbf{e}_2)^{-1})\mathbf{T}^{-1}$. Using the same method, one also obtains $\mathbf{S}$. However, currently there exists no efficient algorithm which solves the generator of principal ideal lattice for large dimension n.

Eigenvalue attack [CHL+14]. Because $\mathbf{V}_{i,j,t} = \mathbf{T}(Rot(\mathbf{r}_{i,j,t}\mathbf{g} + \mathbf{e}_{i,j,t}))\mathbf{S} = \mathbf{T}\mathbf{E}_{i,j,t}\mathbf{S}$, one can generate $\mathbf{V}_{i,j,t}(\mathbf{V}_{i',j',t'})^{-1} = \mathbf{T}\mathbf{E}_{i,j,t}(\mathbf{E}_{i',j',t'})^{-1}\mathbf{T}^{-1}$ and $(\mathbf{V}_{i',j',t'})^{-1}\mathbf{V}_{i,j,t} = \mathbf{S}^{-1}(\mathbf{E}_{i',j',t'})^{-1}\mathbf{E}_{i,j,t}\mathbf{S}$. However, the matrices $\mathbf{E}_{i,j,t}(\mathbf{E}_{i',j',t'})^{-1}$ and $(\mathbf{E}_{i',j',t'})^{-1}\mathbf{E}_{i,j,t}$ are not diagonal. Therefore, the attack in [CHL+14] cannot work for this case.

Lattice reduction attack. Given $\mathbf{V}_{i,j,t}$, one can obtain the bases of the lattices generated by $\mathbf{T}$ and $\mathbf{S}$. However, at present there exists no efficient algorithm, which computes $\mathbf{T}$ and $\mathbf{S}$ for large dimension n. Without loss of generality, assume that $\mathbf{T}' = \mathbf{T}\cdot\mathbf{C}_1$ and $\mathbf{S}' = \mathbf{C}_2\cdot\mathbf{S}$ are the bases of the lattices generated by $\mathbf{T}$ and $\mathbf{S}$, where $\mathbf{C}_1, \mathbf{C}_2$ are unimodular matrices, one can compute $(\mathbf{V}_{i,j,t})' = (\mathbf{T}')^{-1}\mathbf{V}_{i,j,t}(\mathbf{S}')^{-1} = (\mathbf{C}_1)^{-1}(Rot(\mathbf{r}_{i,j,t}\mathbf{g} + \mathbf{e}_{i,j,t}))(\mathbf{C}_2)^{-1}$. However, one cannot remove the matrices $(\mathbf{C}_1)^{-1}, (\mathbf{C}_2)^{-1}$ of both sides of $(\mathbf{V}_{i,j,t})'$. Thus, one cannot get the principal ideal $\mathbf{r}_{i,j,t}\mathbf{g} + \mathbf{e}_{i,j,t}$ in $\mathbf{V}_{i,j,t}$.

Lattice reduction attack for level-1 encoding. Because $\mathbf{U} = \left[\sum_{i=1}^{\tau}(d_i \cdot \mathbf{Y}_i)\right]_q$, then the entry $U_{j,t} = \left[\sum_{i=1}^{\tau} d_i \cdot Y_{i,j,t}\right]_q, \forall j, t \in [\![n]\!]$. Thus, $U_{j,t}, Y_{i,j,t}, q$ consist of a generalizing subset sum problem. However, for large τ there exist no efficient algorithm, which solves this generalizing subset sum problem. Moreover, it is easy to verify that one cannot also use linear

equation system to solve $d_i, i \in [\![\tau]\!]$ since $\tau > n$.

4 Construction of asymmetric multilinear maps

Although our symmetric construction does not give level-1 encoding of zero, one can also generate level-κ encodings of zero by using the public parameters. In this section, we describe a construction of asymmetric multilinear maps to avoid any level encoding of zero.

4.1 Construction

In our symmetric construction, the level-κ encodings of zero is generated by cross-multiplying the level-1 encoding and the zero-testing parameter in the public parameters. If in a scheme, its level-1 encoding cannot multiply by the zero-testing parameter belonging to same group, then the level-κ encodings of zero cannot be generated. Therefore, the starting point of our work is to construct an asymmetric version, which assigns "index set" to the encodings and the zero-testing parameters in the public parameter. As a result, an encoding and a zero-testing parameter cannot be multiplied if their "index sets" are not disjoint. Our asymmetric construction is as follows:

Instance generation: $(\text{par}_1) \leftarrow \text{InstGen}_1(1^\lambda, 1^\kappa)$

(1) Choose a prime q;

(2) Choose $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$ such that $\left\|\mathbf{g}^{-1}\right\| \le n^2$;

(3) Choose $\mathbf{a}_{j,i}, \mathbf{e}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sigma'}, \mathbf{b}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}, j \in [\![\kappa]\!], i \in [\![\tau]\!]$;

(4) Choose a random element $\mathbf{z}_j \leftarrow R_q$, $j \in [\![\kappa]\!]$ such that $\mathbf{z}_j^{-1} \in R_q$;

(5) Choose matrices $\mathbf{S}_j \leftarrow D_{\mathbb{Z}^{n\times n},\sigma}, j \in \{0,1,...,\kappa\}$ such that $\mathbf{S}_j^{-1} \in \mathbb{Z}_q^{n\times n}, j \in [\![\kappa]\!]$;

(6) Set $\mathbf{z}_j^* = (\prod_{t=1}^{\kappa} \mathbf{z}_t)/\mathbf{z}_j, j \in [\![\kappa]\!]$, and $\mathbf{T}_j = \mathbf{S}_j, j \in \{0,1,...,\kappa-1\}$, $\mathbf{T}_\kappa = (\mathbf{S}_\kappa)^{-1}$.

For $j \in [\![\kappa]\!]$, $i \in [\![\tau]\!]$,

set $\mathbf{Y}_{\{j\},i} = \left[\mathbf{T}_{j-1} Rot(\frac{\mathbf{a}_{j,i}\mathbf{g}+\mathbf{e}_{j,i}}{\mathbf{z}_j})\mathbf{T}_j^{-1}\right]_q$, $\mathbf{P}_{\{j\},i} = \left[\mathbf{T}_{j-1} Rot(\frac{\mathbf{z}_j^*(\mathbf{b}_{j,i}\mathbf{g}+\mathbf{e}_{j,i})}{\mathbf{g}})\mathbf{T}_j^{-1}\right]_q$.

(7) Output the public parameter $\text{par}_1 = \left\{q, \left\{\mathbf{Y}_{\{j\},i}, \mathbf{P}_{\{j\},i}\right\}, j \in [\![\kappa]\!], i \in [\![\tau]\!]\right\}$.

Generating encodings with index $\{j\}$**:** $\mathbf{U}_{\{j\}} \leftarrow \text{enc}(\text{par}_1, \{j\}, \mathbf{d})$.

Given $\mathbf{d} \leftarrow D_{\mathbb{Z}^\tau,\sigma^*}$, an index-$\{j\}$ encoding of hidden $\mathbf{e}_j = \sum_{i=1}^{\tau}(d_i \cdot \mathbf{e}_{j,i})$ is computed as

$\mathbf{U}_{\{j\}} = \left[\sum_{i=1}^{\tau}(d_i \cdot \mathbf{Y}_{\{j\},i})\right]_q = \left[\mathbf{T}_{j-1} Rot(\frac{\mathbf{a}_j\mathbf{g}+\mathbf{e}_j}{\mathbf{z}_j})\mathbf{T}_j^{-1}\right]_q$, where $\mathbf{a}_j = \sum_{i=1}^{\tau}(d_i \mathbf{a}_{j,i})$.

Adding encodings with index $S = [\![j+t]\!] \setminus [\![j]\!]$**:** $\mathbf{U}_S \leftarrow \text{add}(\text{par}_1, \mathbf{U}_{S,1}, \cdots, \mathbf{U}_{S,m})$.

Given m encodings $\mathbf{U}_{S,l}, l \in [\![m]\!]$ with index S, their sum $\mathbf{U}_S = \left[\sum_{l=1}^{m} \mathbf{U}_{S,l}\right]_q$ is an encoding with index S.

Multiplying encodings: $\mathbf{U}_S \leftarrow \text{mul}(\text{par}_1, S, \mathbf{U}_{\{j_1+1\}}, \cdots, \mathbf{U}_{\{j_1+t\}})$.

Given t encodings $\mathbf{U}_{\{j\}}$ for $j \in S = [\![j_1 + t]\!] \setminus [\![j_1]\!]$, their product $\mathbf{U}_S = \left[\mathbf{U}_{\{j_1+1\}} \times \cdots \times \mathbf{U}_{\{j_1+t\}} \right]_q$ is an encoding with index $S = [\![j_1 + t]\!] \setminus [\![j_1]\!]$.

Zero testing: $\mathrm{isZero}(\mathrm{par}_1, \mathbf{U}_S)$.

For simplicity, we assume $S = [\![\kappa - 1]\!]$. To determine whether $\mathbf{U}_S$ with index S is an encoding of zero, $\mathbf{V} = \left[\mathbf{U}_S \times \mathbf{P}_{\{\kappa\}} \right]_q$ is computed in $\mathbb{Z}_q^{n \times n}$ and checked whether $\|\mathbf{V}\|$ is short:

$$\mathrm{isZero}(\mathrm{par}_1, \mathbf{U}_S) = \begin{cases} 1 & \text{if } \left\| \left[\mathbf{U}_S \times \mathbf{P}_{\{\kappa\}} \right]_q \right\| < q^{3/4} \\ 0 & \text{otherwise} \end{cases},$$

where $\mathbf{P}_{\{\kappa\}} = \sum_{i=1}^{\tau} r_i \mathbf{P}_{\{\kappa\},i}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}^\tau, \sigma}$.

For $j \in S = [\![\kappa - 1]\!]$, assume $\mathbf{U}_{\{j\}} = \left[\mathbf{T}_{j-1} Rot(\frac{\mathbf{a}_j \mathbf{g} + \mathbf{e}_j}{\mathbf{z}_j}) \mathbf{T}_j^{-1} \right]_q$, then we have

$$\mathbf{U}_S = \left[\prod_{j=1}^{\kappa-1} \mathbf{U}_{\{j\}} \right]_q = \left[\mathbf{T}_1 Rot(\frac{\mathbf{ag} + \mathbf{e}}{\mathbf{z}_\kappa^*}) (\mathbf{T}_{\kappa-1})^{-1} \right]_q,$$

where $\mathbf{e} = \prod_{j=1}^{\kappa-1} \mathbf{e}_j$, $\mathbf{a} = (\prod_{j=1}^{\kappa-1} (\mathbf{a}_j \mathbf{g} + \mathbf{e}_j) - \mathbf{e}) / \mathbf{g}$.

Since $\mathbf{P}_{\{\kappa\}} = \sum_{i=1}^{\tau} r_i \mathbf{P}_{\{\kappa\},i} = \left[\mathbf{T}_{\kappa-1} Rot(\frac{\mathbf{z}_\kappa^* (\mathbf{bg} + \mathbf{c})}{\mathbf{g}}) (\mathbf{T}_\kappa)^{-1} \right]_q$, where $\mathbf{c} = \sum_{i=1}^{\tau} r_i \mathbf{e}_{\kappa,i}$,

$\mathbf{b} = \sum_{i=1}^{\tau} r_i \mathbf{b}_{\kappa,i}$, then we have

$$\begin{aligned} \mathbf{V} &= \left[\mathbf{U}_S \times \mathbf{P}_{\{\kappa\}} \right]_q \\ &= \left[\mathbf{T}_0 Rot(\frac{\mathbf{ag} + \mathbf{e}}{\mathbf{z}_\kappa^*}) (\mathbf{T}_{\kappa-1})^{-1} \cdot \mathbf{T}_{\kappa-1} Rot(\frac{\mathbf{z}_\kappa^* (\mathbf{bg} + \mathbf{c})}{\mathbf{g}}) \mathbf{T}_\kappa \right]_q \\ &= \left[\mathbf{T}_0 Rot(\frac{\mathbf{ag} + \mathbf{e}}{\mathbf{g}} (\mathbf{bg} + \mathbf{c})) (\mathbf{T}_\kappa)^{-1} \right]_q \\ &= \left[\mathbf{S}_0 Rot(\frac{\mathbf{ag} + \mathbf{e}}{\mathbf{g}} (\mathbf{bg} + \mathbf{c})) \mathbf{S}_\kappa \right]_q \end{aligned} .$$

If $\mathbf{U}_S$ is an encoding of zero, namely $\mathbf{e} = \mathbf{0} \bmod I$, then $\mathbf{V}$ is not reduced modulo q and $\|\mathbf{V}\|$ is small. Otherwise, $\mathbf{e} \neq \mathbf{0} \bmod I$, and $\left\| \left[Rot(\frac{\mathbf{ag} + \mathbf{e}}{\mathbf{g}}) \right]_q \right\| \approx q$ by lemma 4 in [GGH13]. Hence, $\mathbf{P}_{\{\kappa\}}$ is a zero testing parameter of $\mathbf{U}_S$ with index $S = [\![\kappa - 1]\!]$.

For $S = [\![j]\!], 1 \le j < \kappa$, one can determine whether $\mathbf{U}_S$ is an encoding of zero. Without loss of generality, assume $\mathbf{U}_{S_1}$ is an arbitrary encoding with index $S_1 = [\![\kappa]\!] \setminus [\![j+1]\!]$, and $\mathbf{P}_{\{j+1\}} = \sum_{i=1}^{\tau} r_i \mathbf{P}_{\{j+1\},i}$ is a random zero-testing parameter for $\mathbf{U}_S$. Then

$\mathbf{V} = \left[\mathbf{U}_S \times \mathbf{P}_{\{j+1\}} \times \mathbf{U}_{S_1} \right]_q$ is computed and checked $\|\mathbf{V}\| < q^{3/4}$.

Similarly, for other index $S = [\![j_1 + t]\!] \setminus [\![j_1]\!], S \subset [\![\kappa]\!]$, one can determine whether $\mathbf{U}_S$ is an encoding of zero by using $\mathbf{P}_{\{t\}}, t \in [\![j_1]\!]$.

Extraction: $sk \leftarrow \mathrm{ext}(\mathrm{par}_1, \mathbf{U}_S)$.

Assume $S = [\![\kappa - 1]\!]$. Given an index-S encoding $\mathbf{U}_S$, $\mathbf{U}_S$ is multiplied by a zero-testing parameter $\mathbf{P}_{\{\kappa\}}$, where $\mathbf{P}_{\{\kappa\}} = \sum_{i=1}^{\tau} r_i \mathbf{P}_{\{\kappa\},i}$, $\mathbf{r} \leftarrow D_{\mathbb{Z}^\tau, \sigma}$ and $(\log q)/4 - \lambda$ most-significant bits of each entry of the $n \times n$-matrix $\left[\mathbf{U}_S \times \mathbf{P}_{\{\kappa\}} \right]_q$ is collected:

$$\mathrm{ext}(\mathrm{par}_1, \mathbf{U}_S) = \mathrm{Extract}(\mathrm{msb}(\left[\mathbf{U}_S \times \mathbf{P}_{\{\kappa\}} \right]_q)).$$

For arbitrary valid index-S encoding $\mathbf{U}_S$, one can extract bit string using the similar method.

Remark 4.1 (1) Because both sides of them are multiplied by random matrices in our asymmetric construction, the encodings that have same index can be added, and the encodings that have adjacent index can be multiplied. (2) One cannot generate any level non-trivial encoding of zero using the public parameter in our construction. Although $\mathbf{Y}_{\{j\},i}$, $\mathbf{P}_{\{j\},i}$ encode the same coset of R/I, they cannot be cross-multiplied since $\mathbf{Y}_{\{j\},i_1}\mathbf{P}_{\{j\},i_2} - \mathbf{Y}_{\{j\},i_2}\mathbf{P}_{\{j\},i_1}$ is not an encoding of zero. (3) When constructing one-round multipartite Diffie-Hellman key exchange using our asymmetric scheme, the j-th party generates an index-$\{j\}$ encoding $\mathbf{U}_{\{j\}} = \left[\sum_{i=1}^{\tau} (d_{j,i} \cdot \mathbf{Y}_{\{j\},i}) \right]_q$ and the corresponding zero-testing parameter $\mathbf{P}_{\{j\}} = \left[\sum_{i=1}^{\tau} d_{j,i} \mathbf{T}_{j-1} Rot(\frac{\mathbf{z}_j^* (\mathbf{b}_{j,i}\mathbf{g} + \mathbf{e}_{j,i})}{\mathbf{g}})(\mathbf{T}_j)^{-1} \right]_q$. Given $\mathbf{U}_{\{1\}}, ..., \mathbf{U}_{\{\kappa\}}$, the j-th party computes $\mathbf{V} = \left[\mathbf{U}_{\{1\}} \cdots \mathbf{U}_{\{j-1\}} \cdot \mathbf{P}_{\{j\}} \cdot \mathbf{U}_{\{j+1\}} \cdots \mathbf{U}_{\kappa} \right]_q$ and extracts the common bit string by using $\mathrm{Extract}(\mathrm{msb}(\mathbf{V}))$.

4.2 Security

Currently, we cannot also reduce the security of our asymmetric construction to classical hardness assumptions. The security of our construction relies on new hardness assumption.

Consider the following security experiment:

(1) $\mathrm{par}_1 \leftarrow \mathrm{InstGen}_1(1^\lambda, 1^\kappa)$.

(2) For $j = 1$ to κ:

Sample $\mathbf{r}_j, \mathbf{w}_j \leftarrow D_{\mathbb{Z}^\tau, \sigma^*}$;

Generate $\{j\}$-index encoding of hidden $\mathbf{d}_j = \sum_{i=1}^{\tau} w_{j,i} \mathbf{e}_{j,i}$:

$$\mathbf{U}_{\{j\}} = \left[\sum_{i=1}^{\tau} w_{j,i} \mathbf{Y}_{\{j\},i} \right]_q.$$

(3) Set $\mathbf{U}_{[\![\kappa-1]\!]} = \left[\prod_{j=1}^{\kappa-1} \mathbf{U}_{\{j\}} \right]_q$.

(4) Set $\mathbf{V}_C = \mathbf{V}_D = \left[\mathbf{U}_{[\![\kappa-1]\!]} \mathbf{P}_{\{\kappa\}} \right]_q$, where $\mathbf{P}_{\{\kappa\}} = \left[\sum_{i=1}^{\tau} w_{\kappa,i} \mathbf{P}_{\{\kappa\},i} \right]_q$.

(5) Set $\mathbf{V}_R = \left[\mathbf{U}_{[\![\kappa-1]\!]} \mathbf{P}_{\{\kappa\_r\}} \right]_q$, where $\mathbf{P}_{\{\kappa\_r\}} = \left[\sum_{i=1}^{\tau} r_{\kappa,i} \mathbf{P}_{\{\kappa\},i} \right]_q$.

Definition 4.2 (ext-GCDH/ext-GDDH). According to the above experiment, the ext-GCDH and ext-GDDH are defined as follows:

Extraction GCDH (ext-GCDH): Given $\left\{ \mathrm{par}_1, \mathbf{U}_{\{1\}}, \cdots, \mathbf{U}_{\{\kappa\}} \right\}$, output an extraction encoding $\mathbf{W} \in \mathbb{Z}_q^{n \times n}$ such that $\left\| \left[\mathbf{V}_C - \mathbf{W} \right]_q \right\|_\infty \le q^{3/4}$.

Extraction GDDH (ext-GDDH): Given $\left\{ \mathrm{par}_1, \mathbf{U}_{\{1\}}, \cdots, \mathbf{U}_{\{\kappa\}}, \mathbf{V} \right\}$, distinguish between $D_{ext-GDDH} = \left\{ \mathrm{par}_1, \mathbf{U}_{\{1\}}, \cdots, \mathbf{U}_{\{\kappa\}}, \mathbf{V}_D \right\}$ and $D_{ext-RAND} = \left\{ \mathrm{par}_1, \mathbf{U}_{\{1\}}, \cdots, \mathbf{U}_{\{\kappa\}}, \mathbf{V}_R \right\}$.

5 Commutative Variant

In our symmetric/asymmetric construction, the dimension n requires to be large enough to guarantee security and $\tau > n$ is the lowest requirement to avoid algebraic equation attack. As a result, the public parameter size of our construction is larger than that of GGH. To decrease the public parameter size, we use polynomial ring instead of the ring of integers. Moreover, we will also use polynomial drowning method of Rényi divergence which is used in the security analysis of [LLS14].

We use $R^y = \mathbb{Z}[y] / < y^m + 1 >$ and $R_q^y = \mathbb{Z}_q[y] / < y^m + 1 >$ instead of $\mathbb{Z}$ and $\mathbb{Z}_q$ for our symmetric/asymmetric constructions. It is easy to verify that our constructions are still correct under this case.

Let λ be the security parameter, $m = \lambda^{O(1)}$ and n constant number (e.g. $n = 4, 8$). Let $R^{yx} = R^y[x] / < x^n + 1 >$ and $R_q^{yx} = \mathbb{Z}_q[y][x] / < y^m + 1 >< x^n + 1 >$. In this section, we let $\|\mathbf{a}\|$ denote the infinity norm of $\mathbf{v} = (\|\mathbf{a}_1\|, ..., \|\mathbf{a}_n\|)$ for $\mathbf{a} \in R^{yx}$.

For completeness, we adaptively describe the commutative variant of the symmetric construction in Section 3.1 as follows:

Instance generation: $(\mathrm{par}_2) \leftarrow \mathrm{InstGen}_2(1^\lambda, 1^\kappa)$.

(1) Pick a prime q;

(2) Choose $\mathbf{g} \leftarrow D_{\mathbb{Z}^{n \times m}, \sigma}$ over R^{yx} such that $\left\| \mathbf{g}^{-1} \right\| \le n^2$, where $\mathbf{g}^{-1} \in \mathbb{Q}[y][x] / < y^m + 1 >< x^n + 1 >$;

(3) Choose $\mathbf{a}_i, \mathbf{b}_i, \mathbf{e}_i \leftarrow D_{\mathbb{Z}^{n \times m}, \sigma^*}, i \in [\![\tau]\!]$ over R^{yx};

(4) Choose randomly $\mathbf{z} \leftarrow R_q$ over R_q^{yx} such that $\mathbf{z}^{-1} \in R_q$;

(5) Choose matrices $\mathbf{T} \leftarrow D_{\mathbb{Z}^{n \times n \times m}, \sigma'}$, $\mathbf{S} \leftarrow D_{\mathbb{Z}^{n \times n \times m}, \sigma'}$ over $(R^y)^{n \times n}$ so that $\mathbf{T}^{-1} \in (R_q^y)^{n \times n}$;

(6) For $i \in [\![\tau]\!]$, set $\mathbf{Y}_i = \left[\mathbf{T} Rot^y \left(\frac{\mathbf{a}_i \mathbf{g} + \mathbf{e}_i}{\mathbf{z}} \right) \mathbf{T}^{-1} \right]_q$ and $\mathbf{P}_{zt,i} = \left[\mathbf{T} Rot^y \left(\frac{\mathbf{z}^\kappa (\mathbf{b}_i \mathbf{g} + \mathbf{e}_i)}{\mathbf{g}} \right) \mathbf{S} \right]_q$

over $(R_q^y)^{n\times n}$;

(7) Output the public parameter $\text{par}_2=\left\{q,\left\{\mathbf{Y}_i,\mathbf{P}_{zt,i}\right\},i\in[\![\tau]\!]\right\}$.

Generating level-1 encoding: $\mathbf{U}\leftarrow\text{enc}(\text{par}_2,1,\mathbf{d}_i)$.

Given τ elements $\mathbf{d}_i\leftarrow D_{\mathbb{Z}^m,\sigma^*}$, then $\mathbf{U}=\left[\sum_{i=1}^{\tau}(\mathbf{d}_i\cdot\mathbf{Y}_i)\right]_q$ is a level-1 encoding of hidden level-0 encoding $\mathbf{e}$=$\sum_{i=1}^{\tau}(\mathbf{d}_i\cdot\mathbf{e}_i)$.

Adding encodings: $\mathbf{U}\leftarrow\text{add}(\text{par}_2,j,\mathbf{U}_1,\cdots,\mathbf{U}_m)$.

Given m level-j encodings $\mathbf{U}_l$, their sum $\mathbf{U}=\left[\sum_{l=1}^{m}\mathbf{U}_l\right]_q$ is a level-j encoding.

Multiplying encodings: $\mathbf{U}\leftarrow\text{mul}(\text{par}_2,1,\mathbf{U}_1,\cdots,\mathbf{U}_\kappa)$.

Given κ level-1 encodings $\mathbf{U}_j$, their product $\mathbf{U}=\left[\prod_{j=1}^{\kappa}\mathbf{U}_j\right]_q$ is a level-κ encoding.

Zero testing: $\text{isZero}(\text{par}_2,\mathbf{U})$.

To determine whether $\mathbf{U}=\left[\mathbf{T}Rot(\frac{\mathbf{rg}+\mathbf{e}}{\mathbf{z}^\kappa})\mathbf{T}^{-1}\right]_q$ is a level-κ encoding of zero, $\mathbf{V}=\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q$ is computed in $(R_q^y)^{n\times n}$ and checked whether $\|\mathbf{V}\|$ is short:

$$\text{isZero}(\text{par},\mathbf{U})=\begin{cases}1 & \text{if } \left\|\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q\right\|<q^{3/4}\\ 0 & \text{otherwise}\end{cases}, \text{ where } \mathbf{P}_{zt}=\sum_{i=1}^{\tau}\mathbf{r}_i\cdot\mathbf{P}_{zt,i},\ \mathbf{r}_i\leftarrow D_{\mathbb{Z}^m,\sigma}.$$

Extraction: $sk\leftarrow\text{ext}(\text{par}_2,\mathbf{d}_i,\mathbf{U})$.

Given a level-κ encoding $\mathbf{U}$, $\mathbf{U}$ is multiplied by $\mathbf{P}_{zt}=\sum_{i=1}^{\tau}(\mathbf{d}_i\cdot\mathbf{P}_{zt,i})$ and $(\log q)/4-\lambda$ most-significant bits of each coefficient of each entry in $\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q$ is collected:

$$\text{ext}(\text{par}_2,\mathbf{d}_i,\mathbf{U})=\text{Extract}(\text{msb}(\left[\mathbf{U}\cdot\mathbf{P}_{zt}\right]_q)).$$

Similarly, we can construct the commutative variant of our asymmetric multilinear maps in Section 4.1.

6 Simplified variant of asymmetric construction

In this section, we give a simplified variant of our asymmetric multilinear maps using polynomial ring, instead of the ring of integers, to reduce the public parameter size. In fact, our simplified variant sets $\mathbf{S}_i=\mathbf{I}$ for our asymmetric construction in Section 4.1.

Our simplified asymmetric construction is an asymmetric variant in [GGH13]. In a sense, our asymmetric simplified variant is an extension of the multilinear Jigsaw puzzles [GGH+13a]. The main difference is that our construction modifies the zero-testing parameter, which also encodes the hidden plaintext encoded by the level-1 encoding. Hence, in our construction, one can generate level-1 encoding of hidden plaintext, which can be used according to the corresponding zero-testing parameter. Moreover, the aim setting $\mathbf{b}_{j,i}\leftarrow D_{\mathbb{Z}^n,\sqrt{q}}$ is to guarantee that one cannot

generate any level nontrivial encoding of zero for our asymmetric simplified variant. To reduce the public parameter size, we use polynomial drowning method of Rényi divergence which is used in the security analysis of [LLS14] and set $\tau=2$.

For completeness, we give our simplified variant as follows:

Instance generation: $(\text{par}_3) \leftarrow \text{InstGen}_3(1^\lambda, 1^\kappa)$

(1) Choose a prime q;

(2) Choose $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\sigma}$ such that $\left\|\mathbf{g}^{-1}\right\| \le n^2$;

(3) Choose $\mathbf{a}_{j,i}, \mathbf{e}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sigma'}, \mathbf{b}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}, j \in [\![\kappa]\!], i \in [\![\tau]\!]$;

(4) Choose random element $\mathbf{z}_j \leftarrow R_q$, $j \in [\![\kappa]\!]$ such that $\mathbf{z}_j^{-1} \in R_q$;

(5) Set $\mathbf{z}_j^* = (\prod_{t=1}^{\kappa} \mathbf{z}_t) / \mathbf{z}_j, j \in [\![\kappa]\!]$. For $j \in [\![\kappa]\!]$, $i \in [\![\tau]\!]$,

set $\mathbf{y}_{\{j\},i} = \left[\frac{\mathbf{a}_{j,i}\mathbf{g} + \mathbf{e}_{j,i}}{\mathbf{z}_j}\right]_q$, $\mathbf{p}_{\{j\},i} = \left[\frac{\mathbf{z}_j^*(\mathbf{b}_{j,i}\mathbf{g} + \mathbf{e}_{j,i})}{\mathbf{g}}\right]_q$;

(6) Output the public parameter $\text{par}_3 = \left\{q, \left\{\mathbf{y}_{\{j\},i}, \mathbf{p}_{\{j\},i}\right\}, j \in [\![\kappa]\!], i \in [\![\tau]\!]\right\}$.

Generating encodings: $\mathbf{u}_i \leftarrow \text{enc}(\text{par}_3, \{j\}, \mathbf{d})$.

Given $\mathbf{d}_i \leftarrow D_{\mathbb{Z}^n,\sigma^*}, i \in [\![\tau]\!]$, an index-$\{j\}$ encoding of hidden $\mathbf{e}_j = \sum_{i=1}^{\tau}(\mathbf{d}_i \cdot \mathbf{e}_{j,i})$ is computed as $\mathbf{u}_{\{j\}} = \left[\sum_{i=1}^{\tau}(\mathbf{d}_i \cdot \mathbf{y}_{\{j\},i})\right]_q = \left[\frac{\mathbf{a}_j\mathbf{g} + \mathbf{e}_j}{\mathbf{z}_j}\right]_q$, where $\mathbf{a}_j = \sum_{i=1}^{\tau}(\mathbf{d}_i \cdot \mathbf{a}_{j,i})$.

Adding encodings: $\mathbf{u}_S \leftarrow \text{add}(\text{par}_3, \mathbf{u}_{S,1}, \cdots, \mathbf{u}_{S,m})$.

Given m encodings $\mathbf{u}_{S,l}, l \in [\![m]\!]$ with index $S \subset [\![\kappa]\!]$, their sum $\mathbf{u}_S = \left[\sum_{l=1}^{m} \mathbf{u}_{S,l}\right]_q$ is an encoding with index S.

Multiplying encodings: $\mathbf{u}_{S_1 \cup S_2} \leftarrow \text{mul}(\text{par}_3, \mathbf{u}_{S_1}, \mathbf{u}_{S_2})$.

Given encodings $\mathbf{u}_{S_1}$, $\mathbf{u}_{S_2}$ with index $S_1, S_2 \subset [\![\kappa]\!], S_1 \cap S_2 = \varnothing$, their product $\mathbf{u}_{S_1 \cup S_2} = \left[\mathbf{u}_{S_1} \times \mathbf{u}_{S_2}\right]_q$ is an encoding with index $S = S_1 \cup S_2$.

Zero testing: $\text{isZero}(\text{par}_3, \mathbf{u}_S)$.

Assume $S = [\![\kappa - 1]\!]$. To determine whether $\mathbf{u}_S$ with index S is an encoding of zero, $\mathbf{v} = \left[\mathbf{u}_S \times \mathbf{p}_{\{\kappa\}}\right]_q$ is computed in R_q and checked whether $\|\mathbf{v}\|$ is short:

$$\text{isZero}(\text{par}_2, \mathbf{u}_S) = \begin{cases} 1 & \text{if } \left\|\left[\mathbf{u}_S \times \mathbf{p}_{\{\kappa\}}\right]_q\right\| < q^{3/4} \\ 0 & \text{otherwise} \end{cases},$$

where $\mathbf{p}_{\{\kappa\}} = \sum_{i=1}^{\tau} \mathbf{r}_i \cdot \mathbf{p}_{\{\kappa\},i}$ with $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}$.

Extraction: $sk \leftarrow \text{ext}(\text{par}_3, \mathbf{u}_S)$.

Given an encoding $\mathbf{u}_S$ with index $S = [1, \kappa - 1]$, $\mathbf{u}_S$ is multiplied by a zero-testing parameter $\mathbf{p}_{\{\kappa\}}$ with $\mathbf{p}_{\{\kappa\}} = \sum_{i=1}^{\tau} \mathbf{r}_i \cdot \mathbf{p}_{\{\kappa\},i}$, $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}$, and $(\log q)/4 - \lambda$ most-significant

bits of each coefficient of $\left[\mathbf{u}_S \times \mathbf{p}_{\{\kappa\}}\right]_q$ is collected:

$$\mathrm{ext}(\mathrm{par}_3, \mathbf{u}_S) = \mathrm{Extract}(\mathrm{msb}(\left[\mathbf{u}_S \times \mathbf{p}_{\{\kappa\}}\right]_q)).$$

Lemma 6.1 For the simplified asymmetric variant, one cannot generate a quantity that is not reduced modulo q from the public parameters.

Proof. Because $\mathbf{y}_{\{j\},i}, \mathbf{p}_{\{j\},i}$ with same index $\{j\}$ encode the same coset $e'_{j,i} = \mathbf{e}_{j,i} + I$ of R/I, we have

$$\begin{aligned}\mathbf{u} &= \left[\mathbf{y}_{\{j\},i_1}\mathbf{p}_{\{j\},i_2} - \mathbf{y}_{\{j\},i_2}\mathbf{p}_{\{j\},i_1}\right]_q \\ &= \left[\frac{\mathbf{z}_j^*(\mathbf{a}_{j,i_1}\mathbf{b}_{j,i_2}\mathbf{g} + \mathbf{a}_{j,i_1}\mathbf{e}_{j,i_2} + \mathbf{b}_{j,i_2}\mathbf{e}_{j,i_1} - \mathbf{a}_{j,i_2}\mathbf{b}_{j,i_1}\mathbf{g} - \mathbf{a}_{j,i_2}\mathbf{e}_{j,i_1} - \mathbf{b}_{j,i_1}\mathbf{e}_{j,i_2})}{\mathbf{z}_j}\right]_q. \\ &= \left[\frac{\mathbf{z}_j^*\mathbf{b}_j}{\mathbf{z}_j}\right]_q\end{aligned}$$

To cancel the denominator $\mathbf{z}_j$ of $\mathbf{u}$, one must multiply $\mathbf{u}$ by some $\mathbf{p}_{\{t\}} = \left[\frac{\mathbf{z}_j^*\mathbf{b}_t}{\mathbf{g}}\right]_q, t \neq j$.

However, by $\mathbf{b}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sqrt{q}}$ we know $\|\mathbf{b}_j\| > \sqrt{q}$ and $\|\mathbf{b}_t\| > \sqrt{q}$. Thus, $\mathbf{v} = \left[\mathbf{u} \cdot \mathbf{u}' \cdot \mathbf{p}_{\{t\}}\right]_q$ must be reduced modulo q, where $\mathbf{u}'$ is an arbitrary rational function of $\mathbf{y}_{\{j\},i}, \mathbf{p}_{\{j\},i}$.

On the other hand, since $\mathbf{y}_{\{j_1\},i}, \mathbf{y}_{\{j_2\},i}$ with different index encode the different hidden coset $e'_{j_1,i} = \mathbf{e}_{j_1,i} + I$, $e'_{j_2,i} = \mathbf{e}_{j_2,i} + I$, one cannot obtain an encoding of zero using arithmetic operations for them. Similarly, one cannot obtain a zero-testing encoded zero from $\mathbf{p}_{\{j_1\},i}, \mathbf{p}_{\{j_2\},i}$.□

7 One round multipartite Diffie-Hellman key exchange

In this section, we first describe the construction of one round multipartite Diffie-Hellman key exchange protocol using commutative variant and asymmetric variant of ideal lattices. Then we optimize and implement one round multipartite Diffie-Hellman key exchange protocol.

7.1 Construction

7.1.1 Construction based on commutative variant

We describe the construction of one round multipartite Diffie-Hellman key exchange using our symmetric commutative variant as follows:

Setup$(1^\lambda, 1^N)$. Output $(\mathrm{par}_2) \leftarrow \mathrm{InstGen}_2(1^\lambda, 1^\kappa)$ as the public parameters. Let $N = \kappa + 1$, $\mathrm{par}_3 = \left\{q, \left\{\mathbf{Y}_i, \mathbf{P}_{zt,i}\right\}, i \in [\![\tau]\!]\right\}$

Publish$(\mathrm{par}_2, \mathbf{p}_{zt}, j)$. The j-th party samples $\mathbf{d}_{j,i} \leftarrow D_{\mathbb{Z}^m,\sigma^*}, i \in [\![\tau]\!]$, computes and

publishes $\mathbf{U}_j = \left[\sum_{i=1}^{\tau}(\mathbf{d}_{j,i} \cdot \mathbf{Y}_i)\right]_q$.

KeyGen$(\text{par}_2, j, \mathbf{d}_{j,i}, \{\mathbf{U}_k\}_{k\neq j})$. The j-th party computes $\mathbf{C}_j = \prod_{k\neq j}\mathbf{U}_k$ and extracts the common secret key $sk = \text{ext}(\text{par}_2, \mathbf{d}_{j,i}, \mathbf{C}_j) = \text{Extract}(\text{msb}(\left[\mathbf{C}_j \cdot (\sum_{i=1}^{\tau}\mathbf{d}_{j,i} \cdot \mathbf{P}_{zt,i})\right]_q))$.

Theorem 7.1 Suppose the ext-GCDH/ext-GDDH defined in Section 3.2 is hard, then our construction is one round multipartite Diffie-Hellman key exchange protocol.

Proof. The proof is similar as Theorem 2 in GGH13.□

7.1.2 Construction based on simplified asymmetric variant

We describe the construction of one round multipartite Diffie-Hellman key exchange using our simplified asymmetric variant as follows:

Setup$(1^{\lambda}, 1^{N})$. Output $(\text{par}_3) \leftarrow \text{InstGen}_3(1^{\lambda}, 1^{\kappa})$ as the public parameters. Let $N = \kappa$, $\text{par}_3 = \left\{q, \left\{\mathbf{y}_{\{j\},i}, \mathbf{p}_{\{j\},i}\right\}, j \in [\![\kappa]\!], i \in [\![\tau]\!]\right\}$

Publish(par_2, j). The j-th party samples $\mathbf{d}_{j,i} \leftarrow D_{\mathbb{Z}^n,\sigma^*}, i \in [\![\tau]\!]$, computes and publishes $\mathbf{u}_{\{j\}} = \left[\sum_{i=1}^{\tau}(\mathbf{d}_{j,i} \cdot \mathbf{y}_{\{j\},i})\right]_q$.

KeyGen$(\text{par}_3, j, \mathbf{d}_{j,i}, \{\mathbf{u}_{\{k\}}\}_{k\neq j})$. The j-th party computes $\mathbf{u}_{S_j} = \prod_{k\neq j}\mathbf{u}_{\{k\}}, S_j = [\![\kappa]\!] - \{j\}$ and extracts the common secret key $sk = \text{ext}(\text{par}_3, \mathbf{d}_{j,i}, \mathbf{u}_{S_j})$.

Theorem 7.2 Suppose the ext-GCDH/ext-GDDH defined in Section 4.2 is hard, then our construction is one round multipartite Diffie-Hellman key exchange protocol.

Proof. The proof is similar as Theorem 2 in GGH13.□

7.2 Implementation

7.2.1 Implement the construction based on the commutative variant

We implement our one round multipartite Diffie-Hellman key exchange protocol using NTL [Sho09].

Setting parameters. Let λ be the security parameter, $m = O(\lambda)$, $n = 4$, $\tau = 8$, $N = \kappa + 1 = 7$. Let $R^y = \mathbb{Z}[y]/<y^m + 1>$, $R^{yx} = R^y[x]/<x^n + 1>$, $R_q^{yx} = \mathbb{Z}_q[y][x]/<y^m + 1><x^n + 1>$. When setting concrete parameters, the coefficients $\mathbf{g}_j, \mathbf{a}_{i,j}, \mathbf{b}_{i,j}, \mathbf{e}_{i,j} \in R^y$ in $\mathbf{g}, \mathbf{a}_i, \mathbf{b}_i, \mathbf{e}_i \in R^{yx}$ are satisfied to $\|\mathbf{g}_j\| = \|\mathbf{a}_{i,j}\| = \|\mathbf{b}_{i,j}\| = \|\mathbf{e}_{i,j}\| = 1$, the entry of the matrices $\mathbf{S}, \mathbf{T} \in (R^y)^{n\times n}$ is satisfied to $\|\mathbf{S}_{i,j}\| = \|\mathbf{T}_{i,j}\| < 3$, and $\mathbf{S}, \mathbf{T}$ are

invertible over R_q. Random sampling $\mathbf{d}_j \in (R^y)^\tau$ is satisfied to $\left\|\mathbf{d}_{j,i}\right\| = 1$. After sampling these parameters, we first compute $\mathbf{V} = \mathbf{T}(Rot(\mathbf{g}\sum_{i=1}^{\tau} d_{j,i}\mathbf{a}_i)^{\kappa+1})\mathbf{S}$ over $(R^y)^{n\times n}$ and $l = \left|q_1\right|, q_1 = \max\left\{\left\|\mathbf{V}_{i,j}\right\| \mid i, j \in [\![n]\!]\right\}$, then set $l+\delta, (20 \le \delta < 25)$ as the bit length of modulo q. When extracting common bits, we only extract one bit from each coefficient. As a result, the probability that the common bits for all parties are inconsistent is about $O(2^{-20})$.

Table 1: The parameters of implementing the protocol based on the commutative variant

n	m	τ	l	δ	|q|	pk size	Setup time	Publish time	Key generation time	Security estimation
2	128	3	70	20	90	100KB	13.1s	0.1s	0.17s	50
2	256	3	79	21	100	205KB	54.2s	0.2s	0.33s	60
2	512	3	88	22	110	425KB	233.1s	0.4s	2.86s	70
2	1024	3	97	23	120	910KB	1220.5s	1.0s	11.5s	80
4	64	8	70	20	90	457KB	15.9s	0.2s	0.21s	50
4	128	8	79	21	100	1034KB	70.4s	0.4s	1.23s	60
4	256	8	88	22	110	2265KB	290.5s	1.0s	5.87s	70
4	512	8	97	23	120	4857KB	1350.5s	2.0s	92.8s	80

Remark 7.3. (1) All algorithms run over single processor (Intel Xeon E5620 4-core CPU, 2.4GHz). In setup stage, solving $\mathbf{g}^{-1}, \mathbf{z}^{-1}, \mathbf{T}^{-1}$ is the most cost time computation. (2) $\left|q\right|$ denotes the bit length of q. (3) Security estimation is the time computing approximate short vector of a lattice using BKZ [CN11].

7.2.2 Implement the construction based on the simplified asymmetric variant

Setting parameters. Let λ be the security parameter, $n = O(\lambda)$, $\tau = 2$, $N = \kappa = 7$. Let $R = R[x]/ < x^n + 1 >$, $R_q = \mathbb{Z}_q[x]/ < x^n + 1 >$. When choosing the parameters, their coefficients $\mathbf{g}_j, \mathbf{a}_{i,j}, \mathbf{b}_{i,j}, \mathbf{e}_{i,j} \in R$ in $\mathbf{g}, \mathbf{a}_i, \mathbf{b}_i, \mathbf{e}_i \in R$ are satisfied to $\left\|\mathbf{g}_j\right\| = \left\|\mathbf{a}_{i,j}\right\| = \left\|\mathbf{b}_{i,j}\right\| = \left\|\mathbf{e}_{i,j}\right\| = 1$. Random sampling $\mathbf{d}_j \in (R)^\tau$ is satisfied to $\left\|\mathbf{d}_{j,i}\right\| = 1$. After choosing these parameters, we first compute $\mathbf{v} = (\mathbf{g}\sum_{i=1}^{\tau} \mathbf{d}_{j,i}\mathbf{a}_i)^{\kappa}$ over R and $l = \left|q_1\right|, q_1 = \max\left\{\left\|\mathbf{v}_i\right\| \mid i \in [\![n]\!]\right\}$, then set $l+\delta, (20 \le \delta < 30)$ as the bit length of modulo q. When extracting common bits, we only

extract one bit from each coefficient. As a result, the probability that the common bits for all parties are inconsistent is about $O(2^{-20})$.

Table 2: The parameters of implementing the protocol based on the simplified asymmetric variant

n	l	δ	|q|	pk size	Setup time	Publish time	Key generation time	Security estimation
256	73	20	93	210KB	17.1s	0.05s	0.3s	50
512	84	21	105	469KB	228.1s	0.10s	1.1s	60
1024	92	23	115	1.0MB	1750.8s	0.30s	2.8s	70
2048	103	23	126	2.1MB	15682.6s	0.65s	5.8s	80

Remark 7.4. Because $\mathbf{u} = \left[\mathbf{y}_{\{j\},i_1}\mathbf{p}_{\{j\},i_2} - \mathbf{y}_{\{j\},i_2}\mathbf{p}_{\{j\},i_1}\right]_q = \left[\frac{\mathbf{z}_j^*\mathbf{b}_j}{\mathbf{z}_j}\right]_q$ by Lemma 6.1, $\mathbf{u}$ must be multiplied by some $\mathbf{p}_{\{t\}}, t \neq j$ to remove $\mathbf{z}_j$. Assume $\mathbf{v}' = \mathbf{u}\cdot\mathbf{p}_{\{t\}} = \frac{\mathbf{z}_j^*\mathbf{z}_t^*\mathbf{r}}{\mathbf{z}_j}$. Under this case, one requires to multiply two times for almost every index encoding to cancel numerator $\mathbf{z}_j^*\mathbf{z}_t^*$. By our setting of modulo q, we have $\left\|(\mathbf{g}\sum_{i=1}^{\tau}\mathbf{d}_{j,i}\mathbf{a}_i)^{2\kappa}\right\| > q$. Thus, setting $\left\|\mathbf{b}_{i,j}\right\| = 1$, one cannot still obtain a nontrivial quantity which is not reduced modulo q.

8 Conclusion and open problem

In this paper, we describe an improved construction of multilinear maps from ideal lattices, multiplying by matrices the level-1 encoding of non-zero. The security of our construction depends upon new hardness assumption, which is seemly closely related to hardness problems of lattices. We also describe an asymmetric construction to avoid any nontrivial encoding of zero. Furthermore, we implement one-round multipartite Diffie-Hellman key exchange protocol to decrease the public parameter size according to the commutative variant and the simplified asymmetric variant.

The security of all current schemes relies on hardness assumption, which cannot be reduced to classical hardness problem. An open problem is to reduce the security of the construction of multilinear maps to classical hardness problem.

References

[BF03] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing, SIAM Journal on Computing, 32(3):586–615, 2003.

[BGG+14] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully keyhomomorphic encryption, arithmetic circuit abe and compact garbled circuits. EUROCRYPT 2014, LNCS 8441, pp. 533-556.

[BR14] Z. Brakerski and G. N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. TCC 2014, LNCS 8349, pp. 1-25.

[BS03] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 324:71–90, 2003.

[BWZ14] D. Boneh, D. J. Wu, and J. Zimmerman. Immunizing multilinear maps against zeroizing attacks. http://eprint.iacr.org/2014/930.

[BZ14]D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. CRYPTO 2014, LNCS 8616, pp. 480-499.

[CHL+14] 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.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0 Better Lattice Security Estimates, ASIACRYPT 2011, LNCS 7073, pp. 1–20.

[CLT13] J. S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. CRYPTO 2013, LNCS 8042, pp. 476–493.

[CLT14] 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.

[GGH13] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. EUROCRYPT 2013, LNCS 7881, pp. 1–17.

[GGH14] C. Gentry, S. Gorbunov, S. Halevi. Graph-induced Multilinear Maps from Lattices. http://eprint.iacr.org/2014/645.

[GGH+13a] 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.

[GGH+13b] 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.

[GGH+14] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. http://eprint.iacr.org/2014/666.

[GSW13a] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. STOC 2013, pp. 467-476.

[GSW13b] C. Gentry, A. Sahai and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. CRYPTO (1) 2013, LNCS 8042, pp. 75-92.

[HAO14] R. Hiromasa, M. Abe and T. Okamoto. Multilinear Maps on LWE. SCIS 2014, pp. 1-8.

[HIL+99] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. SIAM Journal on Computing, 1999, 28(4):1364-1396.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. ANTS 1998, LNCS 1423, pp. 267-288.

[Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. ANTS 2000, LNCS 1838, pp. 385–394.

[LSS14] A. Langlois, D. Stehlé, and R. Steinfeld, GGHLite: More Efficient Multilinear Maps from Ideal Lattices, EUROCRYPT 2014, LNCS 8441, 2014, pp. 239–256.

[PTT10] C. Papamanthou, R. Tamassia, and N. Triandopoulos. Optimal authenticated data structures with multilinear forms. Pairing 2010, LNCS 6487, pp. 246–264.

[Rot13] R. Rothblum. On the circular security of bit-encryption. TCC 2013, LNCS 7785, 2013, pp. 579–598.

[RS09] M. Rückert and D. Schröder. Aggregate and verifiably encrypted signatures from multilinear maps without random oracles. ISA 2009, LNCS 5576, pp. 750–759.

[Sho09] V. Shoup. NTL: A Library for doing Number Theory. http://shoup.net/ntl/, Version 5.5.2, 2009. 2009.08.14.

[Sma03] Smart, N.P. An identity based authenticated key agreement protocol based on the Weil pairing, Electronics Letters, 38(13), pp. 630-632, 2002.

[SOK00] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing, the 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, 2000.

[SS11] D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices, EUROCRYPT 2011, LNCS 6632, pp. 27–47.