Ideal Multilinear Maps Based on Ideal Lattices

Gu Chunsheng

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

May 13, 2015

Abstract. Cryptographic multilinear maps have found many applications, such as multipartite Diffie-Hellman key exchange, general software obfuscation. However, currently only three constructions are known, and are "noisy" and bounded to polynomial degree. In this paper, we first describe our basic constructions of ideal multilinear maps using ideal lattices, which support arbitrary multilinearity level. The security of our construction depends on hardness assumption over ideal lattices. Then, we describe a construction of ideal multilinear maps using unimodular matrix, which removes a usable restriction of our basic construction. Finally, we describe two application based on our ideal constructions: multipartite Diffie-Hellman key exchange and witness encryption.
Keywords. Ideal multilinear maps, ideal lattices, multipartite Diffie-Hellman key exchange, witness encryption, Zeroizing attack

1 Introduction

The construction of multilinear maps has been a long-standing open problem since 2003. Many studies on the applications of bilinear maps, such as [SOK00, Jou00, BF01, Sma03], have influenced research on cryptographic multilinear maps [BS03, RS09, PTT10, Rot13]. Boneh and Silverberg [BS03] first introduced the notion of multilinear maps, which are an extension of bilinear maps. However, they suspected that such maps come from the realm of algebraic geometry.

Garg, Gentry, and Halevi (GGH) recently described the first candidate construction of multilinear maps from ideal lattices [GGH13]. The GGH construction, whose encodings are randomized with noise and bounded with a fixed maximum degree, is different from the ideal multilinear maps envisioned by Boneh and Silverberg [BS03]. Construction security depends on the new hardness assumptions of GCDH/GDDH, which provided extensive cryptanalysis in [GGH13]. Langlois, Stehlé, and Steinfeld [LSS14] presented a variant of GGH by reanalyzing its re-randomization process to improve its efficiency. However, by using the zeroizing attack proposed by [GGH13], the application of multipartite key exchange (MPKE) based on GGH was broken by Hu and Jia [HJ15a]. Recently, Gu [Gu15] described a construction of multilinear maps without encoding of zero by designing new zero-testing parameters.

One line of work focused on new constructions of multilinear maps. Following the GGH framework, the second candidate construction of multilinear maps was presented by Coron, Lepoint, and Tibouchi (CLT) [CLT13]. The CLT construction changes from working over ideal lattices to working over integers and is implemented by using many heuristic optimization techniques. However, by using the zeroizing attack, the CLT construction was broken by Cheon et al. [CHL+14]. Boneh, Wu, and Zimmerman [BWZ14] and Garg, Gentry, Halevi, and Zhandry [GGHZ14] proposed two independent approaches to fix the CLT construction [CLT13]. However, Coron, Lepoint, and Tibouchi [CTL14] showed that two fixes can be broken by using an extension of the attack proposed by Cheon et al. [CHL+14]. Recently, Coron, Lepoint, and Tibouchi [CTL15] presented a new variant of CLT by modifying the zero-testing parameter.

The third candidate construction of graph-induced multilinear maps from lattices was proposed by Gentry, Gorbunov, and Halevi [GGH15]. The security of their construction depends on new hardness assumptions and cannot be reduced to LWE or other classic hard

assumptions.

Another line of work focused on the new cryptographic applications of multilinear maps: witness encryption [GGS+13], general program obfuscation [GGH+13b, Zim15], function encryption [GGH+13b], and other applications [GGH+13a, BZ14].

However, all known constructions are noisy multilinear maps. These noisy encodings restrict the number of operations that can be performed and further restrict their applications. In this study, we propose the construction of ideal multilinear maps that can support any multilinearity degree.

However, all current constructions follow the framework of the GGH construction, whose levels are in advance fixed and encodings have noisy. In this paper, we will describe a construction of ideal multilinear maps from ideal lattices.

Our Results. Our main contribution is to describe a basic construction of ideal multilinear maps that use ideal lattices. Construction security depends on a new hardness assumption. Our construction works in a polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$, where n is a positive integer. Given secret ring elements $\mathbf{f}_j, \mathbf{g}_j \in R, j = 1,...,m$, we denote $\mathbf{f} = \prod_{j=1}^{m} \mathbf{f}_j$. A level-$1$ encoding of level-0 element $\mathbf{a} \in R$ is $\mathbf{c} = (\mathbf{a} \cdot \mathbf{g}) \bmod \mathbf{f}$, where $\mathbf{g} = \mathbf{g}_j \bmod \mathbf{f}_j$, $\mathbf{a} = \mathbf{a}_j \bmod \mathbf{f}_j$, $j = 1,...,m$. If only given $\mathbf{a}$ and $\mathbf{c}$, then one can compute $\mathbf{g} = (\mathbf{c}/\mathbf{a}) \bmod \mathbf{f}$. So, we provide the multiplier $\mathbf{q} = \mathbf{q}_0 \cdot \mathbf{f}$ of $\mathbf{f}$ in the public parameters to avoid this simple attack. Now, we transform the encoding $\mathbf{c}$ to a new encoding $\mathbf{u} = (\mathbf{c} + \mathbf{r} \cdot \mathbf{f}) \bmod \mathbf{q}$. To decide whether or not $\mathbf{u}$ is an encoding of zero, we provide a zero-testing parameter $\mathbf{p}_{zt} = (\sum_{j=1}^{m} \mathbf{h}_j \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q$ in the public parameters. If the norm of $[\mathbf{p}_{zt} \cdot \mathbf{u}]_q$ is small, then $\mathbf{u}$ is the encoding of zero; otherwise, it is the encoding of non-zero. To use level-0 encoding of level-1 encoding, we generate a list of level-1 encodings and zero-testing parameters $(\mathbf{y}_i, \mathbf{p}_{zt,i})$ such that the level-0 encoding of $\mathbf{y}_i$ is hidden in the corresponding zero-testing parameter $\mathbf{p}_{zt,i}$. This defines an arbitrary degree multilinear map.

Our second contribution is to describe a variant of our basic ideal construction to avoid zeroizing attack. This is because $\mathbf{q}$ is an encoding of zero. To support arbitrary multilinearity levels, we must provide this encoding of zero. So in the variant, we take a large enough multiplier $\mathbf{q}_0$ so that a non-reduced quantity over the modulo q cannot be directly obtained multiplying $\mathbf{q}$ by $\mathbf{p}_{zt}$. In the public parameters, we provide a list of non-zero encodings and its corresponding zero-testing parameters to gradually reduce encoding. We have used this method of constructing new zero-testing parameters in [Gu15] to improve the GGH construction [GGH13].

Our third contribution is to describe a construction of ideal multilinear maps using unimodular matrix. We choose two unimodular matrices $\mathbf{T}, \mathbf{S}$. A level-0 and level-1 encoding $\mathbf{a}, \mathbf{c} = \mathbf{a} \cdot \mathbf{g}$ are transformed into $\mathbf{X} = \mathbf{S}Rot(\mathbf{a} + \mathbf{ef})\mathbf{S}^{-1}$, $\mathbf{Y} = \mathbf{T}^{-1}Rot(\mathbf{ag} + \mathbf{tf})\mathbf{T}$, respectively, where $Rot(\mathbf{a})$ is anti-cyclic matrix of $\mathbf{a} \in R$. The zero-testing parameter is modified as $\mathbf{P}_{zt} = [\mathbf{S}Rot(\mathbf{p}_{zt})\mathbf{T}]_q$. In addition, we provide some encodings of zero $\mathbf{Q}_{x,t} = \mathbf{S}Rot(\mathbf{b}_{x,t}\mathbf{q})\mathbf{S}^{-1}$, $\mathbf{Q}_{y,t} = \mathbf{T}^{-1}Rot(\mathbf{b}_{y,t}\mathbf{q})\mathbf{T}$, where $\mathbf{q} = \mathbf{q}_0 \cdot \mathbf{f}$. This defines a construction of ideal multilinear maps.

Our final contribution is to describe two applications using our ideal construction: multipartite Diffie–Hellman key exchange protocol (MPKE), which supports any number of participants, and witness encryption scheme (WE). In our construction, the size of modulus q does not depend on the multilinearity levels. Thus, the MPKE and WE using our ideal multilinear maps are practical.

The remainder of this paper is organized as follows: we recall some preliminaries in Section 2. We describe a basic construction of ideal multilinear maps that use ideal lattices in Section 3, and a variant of our basic ideal construction in Section 4. We describe a construction of ideal multilinear maps using unimodular matrix in Section 5, and extend its asymmetric ideal variant in Section 6, and commutative variant in Section 7. Finally, we propose two applications using our ideal construction in Section 8.

2 Preliminaries

2.1 Notations

We denote $\mathbb{Z},\mathbb{Q},\mathbb{R}$ the integer ring, rational number field and real number field. Vectors and matrices are denoted in bold. We denote $[\![k]\!]=\{1,2,\cdots,k\}$ for $k\in\mathbb{N}$. We take n as a power of two, the polynomial ring $R=\mathbb{Z}[x]/<x^n+1>$ and polynomial field $\mathbb{k}=\mathbb{Q}[x]/<x^n+1>$. For $\mathbf{a}\in R$, $\|\mathbf{a}\|_\infty$ ($\|\mathbf{a}\|$ for short) denotes the infinity norm of the vector corresponding to $\mathbf{a}$.

We use the absolute minimum residual system, namely $[a]_q = a \bmod q \in (-q/2, q/2]$. Similarly, notation $[\mathbf{a}]_q$ denotes each entry (or each coefficient) $a_i \in (-p/2, p/2]$.

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 $\sum_{i=1}^n y_i\mathbf{b}_i$ 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 $L=\{\mathbf{By}:\mathbf{y}\in\mathbb{Z}^n\}$. We say that $\mathbf{B}$ spans L if $\mathbf{B}$ is a basis for L. For any lattice basis, we define $P(\mathbf{B})=\{\mathbf{Bz}\mid\mathbf{z}\in\mathbb{R}^n,\forall i:-1/2\le z_i<1/2\}$. Let $\det(\mathbf{B})$ denote the determinant of the matrix $\mathbf{B}$.

Given $\mathbf{a},\mathbf{g}\in R$, we let the principal ideal $I=<\mathbf{g}>$ with the R_1-basis $Rot(\mathbf{g})=(\mathbf{g},x_2\cdot\mathbf{g},...,x_2^{n-1}\cdot\mathbf{g})$ and $[\mathbf{a}]_\mathbf{g}$ denote the modulo reduction of $I=<\mathbf{g}>$, namely, $[\mathbf{a}]_\mathbf{g}\in P(Rot(\mathbf{g}))$ and $(\mathbf{a}-[\mathbf{a}]_\mathbf{g})\in L(Rot(\mathbf{g}))$.

Given $\mathbf{c}\in\mathbb{R}^n,\sigma>0$, we define $D_{L,\sigma,\mathbf{c}}=\rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(L)$ the Gaussian distribution of a lattice L, where $\mathbf{x}\in L$, $\rho_{\sigma,\mathbf{c}}(\mathbf{x})=\exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$, $\rho_{\sigma,\mathbf{c}}(L)=\sum_{x\in L}\rho_{\sigma,\mathbf{c}}(\mathbf{x})$. 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 $\mathbf{d}\leftarrow D_{I,\sigma}$) over the lattice L (or ideal lattice I).

2.3 Multilinear Maps

Definition 2.1 (Multilinear Map [BS03]). For $k+1$ cyclic groups $G_1,...G_k,G_T$ of the same order p, a k-multilinear map $e:G_1\times\cdots\times G_k\to G_T$ has the following properties:

(1)Elements $\{g_j\in G_j\}_{j=1,...,k}$, index $j\in[\![k]\!]$, and integer $a\in Z_p$ hold that

$$e(g_1,\cdots,a\cdot g_j,\cdots,g_k)=a\cdot e(g_1,\cdots,g_k).$$

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

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

(1) For every index $j\in[\![k]\!]$, 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 $j\in[\![k]\!]$, 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 $j_1,j_2\in[\![k]\!]$ with $j_1+j_2\le k$, 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 Basic ideal multilinear maps

In this section, we first construct symmetric multilinear maps over ideal lattices. Then we show the correctness of our construction. Next, we show the security of our construction. Finally, we give known cryptanalysis for our construction.

3.1 Construction

Setting the parameters. Let λ be the security parameter, n the dimension of R. Concrete parameters are set as $\sigma=\sqrt{\lambda n}$, $\sigma'=\lambda n^{1.5}$, $q\ge 2^{\eta+O(\lambda)}n^{O(1)}$, $n=O(\lambda^2)$, $m=O(\lambda)$, $l=O(n^2)$, $\tau=O(n^2)$.

Instance generation: $(\mathrm{par}_1)\leftarrow \mathrm{InstGen}_1(1^\lambda)$.

(1) Choose a large enough prime q.

(2) Sample $\mathbf{f}_j\leftarrow D_{\mathbb{Z}^n,\sigma}$, $j\in[\![m]\!]$ and $\mathbf{p}_0\leftarrow D_{\mathbb{Z}^n,\sigma'}$ such that ideal lattices $\mathbf{f}_j$'s are coprime, $\mathbf{f}_j^{-1}\in\mathbb{k}$ and $\left\|\mathbf{f}_j^{-1}\right\|\le l$.

(3) Set $\mathbf{f}=\prod_{j=1}^m\mathbf{f}_j$, $\mathbf{q}=\mathbf{p}_0\cdot\mathbf{f}$ and $\mu=\left\|\mathbf{f}\right\|$.

(4) Sample $\mathbf{g}\leftarrow D_{\mathbb{Z}^n,\mu}$, $\mathbf{a}_i\leftarrow D_{\mathbb{Z}^n,\mu}$, $\mathbf{t}_i\leftarrow D_{\mathbb{Z}^n,\sigma'}$, $i\in[\![\tau]\!]$.

(5) Set $\mathbf{y}_i=(\mathbf{a}_i\mathbf{g}+\mathbf{t}_i\mathbf{f})\bmod\mathbf{q}$, $i\in[\![\tau]\!]$.

(6) Sample $\mathbf{h}_j\leftarrow D_{\mathbb{Z}^n,\sigma}$, $\mathbf{s}_{i,j}\leftarrow D_{\mathbb{Z}^n,\sigma'}$, $j\in[\![m]\!]$, $i\in[\![\tau]\!]$.

(7) Set $\mathbf{p}_{zt,i}=(\sum_{j=1}^m\mathbf{h}_j(\mathbf{a}_i+\mathbf{s}_{i,j}\cdot\mathbf{f}_j)\cdot(\mathbf{f}_j^{-1}\bmod q))\bmod q$.

(8) Output the public parameters $\mathrm{par}_1=\left\{q,\mathbf{q},\left\{\mathbf{y}_i,\mathbf{p}_{zt,i}\right\}_{i\in[\![\tau]\!]}\right\}$.

Generating a random level-k encoding: $\mathbf{u}\leftarrow\mathrm{Enc}_1(\mathrm{par}_1,k,\left\{\mathbf{w}_i\right\}_{i\in[\![\tau]\!]})$**.**

Choose $\mathbf{w}_i\leftarrow D_{\mathbb{Z}^n,\sigma}$, $i\in[\![\tau]\!]$, generate a level-k encoding $\mathbf{u}=\sum_{i=1}^\tau\mathbf{w}_i\cdot(\mathbf{y}_i)^k\bmod\mathbf{q}$.

Adding encodings: $\mathbf{u} \leftarrow \mathrm{Add}_1(\mathrm{par}_1, k, \mathbf{u}_1, \cdots, \mathbf{u}_s)$.

Given s level-k encodings $\mathbf{u}_t, t \in [\![s]\!]$, their sum $\mathbf{u} = \sum_{t=1}^{s} \mathbf{u}_t \bmod \mathbf{q}$ is a level-k encoding.

Multiplying encodings: $\mathbf{u} \leftarrow \mathrm{Mul}_1(\mathrm{par}_1, 1, \mathbf{u}_1, \cdots, \mathbf{u}_k)$.

Given k level-1 encodings $\mathbf{u}_t, t \in [\![k]\!]$, their product $\mathbf{u} = \prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{q}$ is a level-k encoding.

Zero Testing: $\mathrm{isZero}_1(\mathrm{par}_1, \mathbf{u}, \{\mathbf{r}_i\}_{i=1}^{\tau})$.

Given a level-k encoding $\mathbf{u}$ and $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$, determine whether $\mathbf{u}$ is an encoding of zero, $\mathbf{v} = [\mathbf{p}_{zt} \cdot \mathbf{u}]_q$ is computed in R_q and checked whether $\|\mathbf{v}\|$ is short:

$$\mathrm{isZero}_1(\mathrm{par}_1, \mathbf{u}, \{\mathbf{r}_i\}_{i=1}^{\tau}) = \begin{cases} 1 & \text{if } \left\|[\mathbf{p}_{zt} \cdot \mathbf{u}]_q\right\| < q/2^{\eta} \\ 0 & \text{otherwise} \end{cases}, \text{ where } \mathbf{p}_{zt} = \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}.$$

Extract: $sk \leftarrow \mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$.

Given a level-k encoding $\mathbf{u}$ and $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$,

$$\mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \mathrm{Extract}_s(\mathrm{msbs}_{\eta}([\mathbf{p}_{zt} \cdot \mathbf{u}]_q)),$$

where $\mathbf{p}_{zt} = \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}$.

In this paper, we omit the seed s and concrete extraction algorithm $\mathrm{Extract}$.

Remark 3.1 (1) In the above construction, one can generate a level-k encoding, whose level-0 encoding is hidden. However, one can use hidden level-0 encoding $\mathbf{d} = \sum_{i=1}^{\tau} (\mathbf{w}_i \cdot \mathbf{a}_i)$ corresponding to level-1 encoding $\mathbf{u} = \sum_{i=1}^{\tau} \mathbf{w}_i \cdot \mathbf{y}_i \bmod \mathbf{q}$ by zero-testing parameter $\mathbf{p}_{zt} = \sum_{i=1}^{\tau} \mathbf{w}_i \mathbf{p}_{zt,i}$. (2) Our construction supports arbitrary level encoding.

3.2 Correctness

Lemma 3.2 $\mathrm{InstGen}_1(1^{\lambda})$ is a probabilistic polynomial time algorithm.

Proof. A prime q can be efficiently generated. By [GGH13], $\mathbf{p}_0 \leftarrow D_{\mathbb{Z}^n,\sigma'}$, $\mathbf{f}_j \leftarrow D_{\mathbb{Z}^n,\sigma}$, $j \in [\![m]\!]$ can be sampled and are satisfied to that $\mathbf{f}_j$'s are coprime, $\mathbf{f}_j^{-1} \in \mathbb{k}$ and $\left\|\mathbf{f}_j^{-1}\right\| \le l$. It is easy to solve $\mathbf{f}_j^{-1} \bmod q$ when $\gcd(\det(Rot(\mathbf{f}_j)), q) = 1$. So, the public parameters par_1 can be generated in polynomial time. ■

Lemma 3.3 $\mathbf{u} \leftarrow \mathrm{Enc}_1(\mathrm{par}_1, k, \mathbf{d})$ is a level-k encoding**.**

Proof. Since $\mathbf{u} = \sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{y}_i)^k \bmod \mathbf{q}$, we have

$$
\begin{aligned}
&\mathbf{u} \bmod \mathbf{f}_j \\
&= (\sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{y}_i)^k \bmod \mathbf{q}) \bmod \mathbf{f}_j \\
&= (\sum_{i=1}^{\tau} \mathbf{w}_i \cdot ((\mathbf{a}_i \mathbf{g} + \mathbf{t}_i \mathbf{f}) \bmod \mathbf{q})^k \bmod \mathbf{q}) \bmod \mathbf{f}_j \\
&= (\sum_{i=1}^{\tau} \mathbf{w}_i \cdot ((\mathbf{a}_i \mathbf{g} + \mathbf{t}_i \mathbf{f}) \bmod \mathbf{f}_j)^k) \bmod \mathbf{f}_j \\
&= (\sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{a}_i \mathbf{g}_j)^k) \bmod \mathbf{f}_j \\
&= (\sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{a}_{i,j})^k) \cdot (\mathbf{g}_j)^k \bmod \mathbf{f}_j \\
&= \mathbf{v}_j \cdot (\mathbf{g}_j)^k \bmod \mathbf{f}_j
\end{aligned}
$$

where $\mathbf{a}_{i,j} = \mathbf{a}_i \bmod \mathbf{f}_j$, $\mathbf{g}_j = \mathbf{g} \bmod \mathbf{f}_j$, $\mathbf{v}_j = \sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{a}_{i,j})^k \bmod \mathbf{f}_j$. So, $\mathbf{u}$ is a level-k encoding of $\mathbf{d}$. ■

Lemma 3.4 $\mathbf{u} \leftarrow \mathrm{Add}_1(\mathrm{par}_1, k, \mathbf{u}_1, \cdots, \mathbf{u}_s)$ is a level-k encoding.

Proof. Using modulo operation, it is easy to verify that $\mathbf{u}$ is a level-k encoding. ■

Lemma 3.5 $\mathbf{u} \leftarrow \mathrm{Mul}_1(\mathrm{par}_1, 1, \mathbf{u}_1, \cdots, \mathbf{u}_k)$ is a level-k encoding.

Proof. Since $\mathbf{u}_t, t \in [\![k]\!]$ are level-1 encodings, we have $\mathbf{u}_t \bmod \mathbf{f}_j = \mathbf{u}_{t,j} \mathbf{g}_j$. Then, we have

$$
\begin{aligned}
&\mathbf{u} \bmod \mathbf{f}_j \\
&= (\prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{q}) \bmod \mathbf{f}_j \\
&= (\prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{f}_j) \bmod \mathbf{f}_j \\
&= (\prod_{t=1}^{k} (\mathbf{u}_t \bmod \mathbf{f}_j)) \bmod \mathbf{f}_j \\
&= (\prod_{t=1}^{k} \mathbf{u}_{t,j} \mathbf{g}_j) \bmod \mathbf{f}_j \\
&= (\prod_{t=1}^{k} \mathbf{u}_{t,j}) (\mathbf{g}_j)^k \bmod \mathbf{f}_j
\end{aligned}
$$

So, $\mathbf{u}$ is a level-k encoding. ■

Lemma 3.6 For an arbitrary integer $k > 0$, the zero-testing algorithm $\mathrm{isZero}_1(\mathrm{par}_1, \mathbf{u})$ correctly determines whether a level-k encoding $\mathbf{u}$ is an encoding of zero.

Proof. Given an arbitrary level-k encoding $\mathbf{u}$, we have $\mathbf{u} = \mathbf{d} + \mathbf{r} \cdot \mathbf{f}$ and $\|\mathbf{u}\| < \|\mathbf{q}\|$ with $\|\mathbf{d}\| < \|\mathbf{f}\|$.

(1) If $\mathbf{u}$ is an encoding of zero, then $\mathbf{u} \bmod \mathbf{f}_j = 0, j \in [\![m]\!]$. Since $\mathbf{f}_j, j \in [\![m]\!]$ are coprime, $\mathbf{u} \bmod \mathbf{f} = 0$. That is, $\mathbf{d} \bmod \mathbf{f} = 0$ and $\mathbf{d} = 0$ according to $\|\mathbf{d}\| < \|\mathbf{f}\|$. So, we have

$$
\begin{aligned}
\mathbf{v} &= \left\| \left[\mathbf{p}_{zt} \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| \left[(\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}) \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j (\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{a}_i + \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{s}_{i,j} \cdot \mathbf{f}_j) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \cdot \mathbf{u}) \bmod q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{u} \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{r} \cdot \mathbf{f} \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \right\| \\
&= \left\| (\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{r} \cdot \mathbf{f} / \mathbf{f}_j) \bmod q \right\| \\
&\le \left\| \sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{r} \cdot \mathbf{f} / \mathbf{f}_j \right\| \\
&\le \sum_{j=1}^{m} \left\| \mathbf{h}_j^{'} \cdot \mathbf{r} \cdot \mathbf{f} / \mathbf{f}_j \right\| \\
&\le \sum_{j=1}^{m} \left\| \mathbf{h}_j^{'} \right\| \cdot \left\| \mathbf{r} \right\| \cdot \left\| \mathbf{f} / \mathbf{f}_j \right\| \\
&\le q / 2^{\eta} \qquad ,
\end{aligned}
$$

where $\mathbf{h}_j^{'} = \mathbf{h}_j (\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{a}_i + \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{s}_{i,j} \cdot \mathbf{f}_j)$.

(2) If $\mathbf{u}$ is not an encoding of zero, then $\mathbf{u} \bmod \mathbf{f} \neq 0$. That is, $\exists j \in [\![m]\!], \mathbf{d} \bmod \mathbf{f}_j \neq 0$. So,

$$
\begin{aligned}
\mathbf{v} &= \left\| \left[\mathbf{p}_{zt} \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| \left[(\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}) \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j (\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{a}_i + \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{s}_{i,j} \cdot \mathbf{f}_j) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \cdot \mathbf{u} \right]_q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \cdot \mathbf{u}) \bmod q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{u} \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \right\| \\
&= \left\| ((\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot (\mathbf{d} + \mathbf{r} \cdot \mathbf{f}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \right\| \\
&= \left\| (\sum_{j=1}^{m} \mathbf{h}_j^{'} \cdot \mathbf{r} \cdot \mathbf{f} / \mathbf{f}_j) + (\sum_{j=1}^{m} (\mathbf{h}_j^{'} \cdot \mathbf{d}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \right\| \\
&\ge \left\| (\sum_{j=1}^{m} (\mathbf{h}_j^{'} \cdot \mathbf{d}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \right\| - \sum_{j=1}^{m} \left\| \mathbf{h}_j^{'} \right\| \cdot \left\| \mathbf{r} \right\| \cdot \left\| \mathbf{f} / \mathbf{f}_j \right\| \\
&\ge q^{1-\varepsilon} - q / 2^{\eta} \\
&\ge q^{1-\varepsilon'}
\end{aligned}
$$

Thus, $\mathrm{isZero}_1(\mathrm{par}_1, \mathbf{u})$ correctly decides the encoding of $\mathbf{u}$. ■

Lemma 3.7 Suppose two level-k encodings $\mathbf{u}_1, \mathbf{u}_2$ encode same level-0 encodings, that is, $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{a}_j (\mathbf{g}_j)^k \bmod \mathbf{f}_j$, $\forall j \in [\![m]\!]$, then

$$\mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}_1, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}_2, \{\mathbf{r}_i\}_{i=1}^{\tau}\right).$$

Proof. Since $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{a}_j(\mathbf{g}_j)^k \bmod \mathbf{f}_j, j \in [\![m]\!]$ and $\mathbf{f}_j, j \in [\![m]\!]$ are co-prime, we have $\mathbf{u}_1 = \mathbf{d} + \mathbf{r}_1 \cdot \mathbf{f}, \mathbf{u}_2 = \mathbf{d} + \mathbf{r}_2 \cdot \mathbf{f}$.

Given $\{\mathbf{r}_i\}_{i=1}^{\tau}$, we have $\mathbf{p}_{zt} = \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}$ and $\mathbf{h}'_j = \mathbf{h}_j(\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{a}_i + \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{s}_{i,j} \cdot \mathbf{f}_j)$. So,

$$\begin{aligned}
\left[\mathbf{p}_{zt} \cdot \mathbf{u}_1\right]_q &= ((\sum_{j=1}^{m} \mathbf{h}'_j \cdot (\mathbf{d} + \mathbf{r}_1 \cdot \mathbf{f}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \\
&= (\sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{r}_1 \cdot \mathbf{f} / \mathbf{f}_j) + (\sum_{j=1}^{m} (\mathbf{h}'_j \cdot \mathbf{d}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \\
\left[\mathbf{p}_{zt} \cdot \mathbf{u}_2\right]_q &= ((\sum_{j=1}^{m} \mathbf{h}'_j \cdot (\mathbf{d} + \mathbf{r}_2 \cdot \mathbf{f}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q) \bmod q \\
&= (\sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{r}_2 \cdot \mathbf{f} / \mathbf{f}_j) + (\sum_{j=1}^{m} (\mathbf{h}'_j \cdot \mathbf{d}) \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q
\end{aligned}.$$

By Lemma 3.6, $(\sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{r}_1 \cdot \mathbf{f} / \mathbf{f}_j)$, $(\sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{r}_2 \cdot \mathbf{f} / \mathbf{f}_j)$ are less than $q / 2^{\eta}$, respectively. Hence, the most significant bits of $\left[\mathbf{p}_{zt} \cdot \mathbf{u}_1\right]_q$, $\left[\mathbf{p}_{zt} \cdot \mathbf{u}_2\right]_q$ are decided by $(\sum_{j=1}^{m} (\mathbf{h}'_j \cdot \mathbf{d}) \cdot (\mathbf{f}_j^{-1} \bmod q))$. That is, $\mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}_1, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}_2, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$. ■

3.3 Security

The security of our constructions depends on new hardness assumptions, and seems to rely on hardness to solve shortest generator problems for principal ideal lattices. However, we currently do not know how to reduce the security of our construction to the shortest principal ideal generator problems.

Hardness assumptions for multilinear maps in [GGH13] are modeled as discrete logarithms and DDH assumptions in multilinear groups. Generating a level-k encoding of the product or distinguishing the product from random elements is unfeasible given the public parameters and $k+1$ level-1 encodings of random elements.

Garg, Gentry, and Halevi introduced the definition of GCDH/GDDH in [GGH13] to describe the hardness assumption of the GGH construction. Langlois, Stehlé, and Steinfeld extended the GCDH/GDDH to the ext-GCDH/ext-GDDH in [LSS14] to describe the security of the GGHLite scheme.

In the following, we adapt the definition of ext-GCDH/ext-GDDH in [LSS14] to our constructions. Consider the following process:

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

(2) Choose an arbitrary positive integer k.

(3) For $t = 0$ to k:

Sample $\mathbf{w}_{t,i} \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$,

Generate level-1 encoding $\mathbf{u}_t = (\sum_{i=1}^{\tau} \mathbf{w}_{t,i} \cdot \mathbf{y}_i) \bmod \mathbf{q}$.

(4) Sample $\mathbf{r}_{0,i} \leftarrow D_{\mathbb{Z}^n,\sigma'}, i \in [\![\tau]\!]$.

(5) Compute $\mathbf{u} = \prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{q}$.

(6) Set $\mathbf{v}_C = \mathbf{v}_D = \mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}, \{\mathbf{w}_{0,i}\}_{i=1}^{\tau}\right)$.

(7) Set $\mathbf{v}_R = \mathrm{Ext}_1\left(\mathrm{par}_1, \mathbf{u}, \{\mathbf{r}_{0,i}\}_{i=1}^{\tau}\right)$.

Definition 3.8 (ext-GCDH/ext-GDDH). The extraction k-graded CDH problem (ext-GCDH)

is, on input $\{\text{par}_1, \mathbf{u}_0, \cdots, \mathbf{u}_k\}$, to output the extraction encoding $\mathbf{v}_C$. The extraction k-graded DDH problem (ext-GDDH) distinguishes between $\mathbf{v}_D$ and $\mathbf{v}_R$, that is, between the distributions $D_{GDDH} = \{\text{par}_1, \mathbf{u}_0, \cdots, \mathbf{u}_k, \mathbf{v}_D\}$ and $D_{RAND} = \{\text{par}_1, \mathbf{u}_0, \cdots, \mathbf{u}_k, \mathbf{v}_R\}$.

As in [GGH13], our construction security depends on new assumptions that are unlikely to be reducible to more classical assumptions. We assume the ideal-GCDH/ ideal-GDDH is hard in our scheme.

3.4 Cryptanalysis

In this section, we describe some known attacks for our construction. In the following section, we will present a variant construction to thwart these attacks.

3.4.1 Average Attack

Since $\mathbf{q} = \mathbf{p}_0 \cdot \mathbf{f}$ is an encoding of zero in our construction, $[\mathbf{q} \cdot \mathbf{p}_{zt}]_q$ is not reduced modulo q. So, the following quantities are easily computed from public parameters through algebraic transformation.

$$\begin{aligned} \mathbf{u} &= [\mathbf{q} \cdot \mathbf{p}_{zt}]_q \\ &= (\mathbf{p}_0 \cdot \mathbf{f})(\sum_{j=1}^{m} \mathbf{h}'_j \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q \\ &= (\mathbf{p}_0 \cdot \sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{f} / \mathbf{f}_j) \bmod q \\ &= \mathbf{p}_0 \cdot \sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{f} / \mathbf{f}_j \\ &= \mathbf{p}_0 \cdot \mathbf{h} \end{aligned} ,$$

where $\mathbf{p}_{zt} = \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}$, $\mathbf{h}'_j = \mathbf{h}_j (\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{a}_i + \sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{s}_{i,j} \cdot \mathbf{f}_j)$, $\mathbf{h} = \sum_{j=1}^{m} \mathbf{h}'_j \cdot \mathbf{f} / \mathbf{f}_j$.

The above fourth equality holds because $\mathbf{f} / \mathbf{f}_j \in R, j \in [\![m]\!]$ and $\|\mathbf{p}_0\|, \|\mathbf{h}'_j\|, \|\mathbf{f} / \mathbf{f}_j\|$ all are small according to our parameter setting. That is, $\mathbf{u} = \mathbf{p}_0 \cdot \mathbf{h}$ is not reduced modulo q. So, one can compute a basis $\mathbf{P}_0$ of $\mathbf{p}_0$ from $\mathbf{u}, \mathbf{q}$ and a basis $\mathbf{F}$ of $\mathbf{f}$. However, the short generators for $\mathbf{p}_0, \mathbf{f}$ cannot currently be found using $\mathbf{P}_0$, $\mathbf{F}$ and $\mathbf{u}, \mathbf{q}$.

For the averaging attacks considered in [GGH13, LS14], the current countermeasure is to increase dimension of ideal lattice of our scheme. The security of our scheme is based on the difficulty of finding any short element of the secret element $\mathbf{f}$.

3.4.2 Lattice reduction attack

Given $\mathbf{u}_t = (\sum_{i=1}^{\tau} \mathbf{w}_{t,i} \cdot \mathbf{y}_i) \bmod \mathbf{q}$, to solve $\{\mathbf{w}_{t,i}\}_{i=1}^{\tau}$. If τ is set large enough, then this attack does not work for our construction.

4 Variant of basic ideal multilinear maps

From the cryptanalysis above, there exist easily computable bases in our scheme. These bases are related to secret ring elements and can threaten our scheme security. This is because $\mathbf{q}$ is an encoding of zero in our scheme above. The reason why we include the encoding of zero is

to obtain ideal multilinear maps.

4.1 Construction

Setting the parameters. Let λ be the security parameter, n the dimension of R. Concrete parameters are set as $\sigma = \sqrt{\lambda n}$, $\sigma' = \lambda n^{1.5}$, $q \geq 2^{\eta + O(\lambda)} n^{O(1)}$, $n = O(\lambda^2)$, $m = O(\lambda)$, $l = O(n^2)$, $\tau = O(n^2)$.

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

(1) Choose a large enough prime q.

(2) Sample $\mathbf{f}_j \leftarrow D_{\mathbb{Z}^n,\sigma}$, $j \in [\![m]\!]$ and $\mathbf{p}_0 \leftarrow D_{\mathbb{Z}^n,q}$ such that all ideal lattices $\mathbf{f}_j$'s are coprime, $\mathbf{f}_j^{-1} \in \mathbb{k}$ and $\left\|\mathbf{f}_j^{-1}\right\| \leq l$.

(3) Set $\mathbf{f} = \prod_{j=1}^{m} \mathbf{f}_j$, $\mathbf{q} = \mathbf{p}_0 \cdot \mathbf{f}$ and $\beta = \|\mathbf{f}\|$.

(4) Sample $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\beta}, \mathbf{a}_i \leftarrow D_{\mathbb{Z}^n,\beta}, \mathbf{t}_i \leftarrow D_{\mathbb{Z}^n,\sigma'}, i \in [\![\tau]\!]$.

(5) Set $\mathbf{y}_i = (\mathbf{a}_i \mathbf{g} + \mathbf{t}_i \mathbf{f}) \bmod \mathbf{q}, i \in [\![\tau]\!]$.

(6) Sample $\mathbf{h}_j \leftarrow D_{\mathbb{Z}^n,\sigma}$, $\mathbf{s}_{0,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma'}$, $j \in [\![m]\!], i \in [\![\tau]\!]$.

(7) Set $\mathbf{p}_{zt,i} = \left[\sum_{j=1}^{m} \mathbf{h}_j (\mathbf{a}_{i,j} + \mathbf{s}_{0,i,j} \cdot \mathbf{f}_j) \cdot (\mathbf{f}_j^{-1} \bmod q)\right]_q$, $i \in [\![\tau]\!]$, where $\mathbf{a}_{i,j} = \mathbf{a}_i \bmod \mathbf{f}_j$, $j \in [\![m]\!]$.

(8) Let $\mu = \lceil \log_\sigma q \rceil$ and $q_t = \sigma^{\mu - t}, t \in [\![\mu - 1]\!]$

(9) Set $\mathbf{q}_t = \mathbf{e}_t + \mathbf{p}_t \cdot \mathbf{f}, t \in [\![\mu - 1]\!]$ such that $\left\|\mathbf{q}_t^{-1}\right\| \leq n(\|\mathbf{q}_t\|)^{-1}$ and $\|\mathbf{q}_{t-1}\| \leq n\sigma \|\mathbf{q}_t\|$, where $\mathbf{p}_t \leftarrow D_{\mathbb{Z}^n,q_t}, \mathbf{e}_t \leftarrow D_{\mathbb{Z}^n,\beta}$.

(10) Set $\mathbf{p}_{zt,t,i} = (\sum_{j=1}^{m} \mathbf{h}_j (\mathbf{e}_{t,j} \mathbf{a}_{i,j} + \mathbf{s}_{t,i,j} \mathbf{f}_j)(\mathbf{f}_j^{-1} \bmod q)) \bmod q$, $t \in [\![\mu - 1]\!]$, $i \in [\![\tau]\!]$, where $\mathbf{e}_{t,j} = \mathbf{e}_t \bmod \mathbf{f}_j$, $\mathbf{s}_{t,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma}, t \in [\![\mu - 1]\!], i \in [\![\tau]\!], j \in [\![m]\!]$.

(11) Output the public parameters

$$\mathrm{par}_2 = \left\{q, \mathbf{q}, \left\{\mathbf{y}_i, \mathbf{p}_{zt,i}\right\}_{i \in [\![\tau]\!]}, \left\{\mathbf{q}_t, \left\{\mathbf{p}_{zt,t,i}\right\}_{i \in [\![\tau]\!]}\right\}_{t \in [\![\mu - 1]\!]}\right\}.$$

Generating a random level-k encoding: $\mathbf{u} \leftarrow \mathrm{Enc}_2(\mathrm{par}_2, k, \{\mathbf{w}_i\}_{i \in [\![\tau]\!]})$.

Choose $\mathbf{w}_i \leftarrow D_{\mathbb{Z}^n,\sigma'}$, $i \in [\![\tau]\!]$, generate a level-k encoding $\mathbf{u} = \sum_{i=1}^{\tau} \mathbf{w}_i \cdot (\mathbf{y}_i)^k \bmod \mathbf{q}$.

Adding encodings: $\mathbf{u} \leftarrow \mathrm{Add}_2(\mathrm{par}_2, k, \mathbf{u}_1, \cdots, \mathbf{u}_s)$.

Given s level-k encodings $\mathbf{u}_t, t \in [\![s]\!]$, their sum $\mathbf{u} = \sum_{t=1}^{s} \mathbf{u}_t \bmod \mathbf{q}$ is a level-k encoding.

Multiplying encodings: $\mathbf{u} \leftarrow \mathrm{Mul}_2(\mathrm{par}_2, 1, \mathbf{u}_1, \cdots, \mathbf{u}_k)$.

Given k level-1 encodings $\mathbf{u}_t$, $t \in [\![k]\!]$, their product $\mathbf{u} = \prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{q}$ is a level-k encoding.

Zero-testing: $\mathrm{isZero}_2\left(\mathrm{par}_2, \mathbf{u}_0, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$.

Given a level-k encoding $\mathbf{u}_0$ and $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$, determine whether $\mathbf{u}_0$ is an encoding of zero:

(1) For $t=1$ to $\mu-1$

Compute $\mathbf{u}_t = \mathbf{u}_{t-1} \bmod \mathbf{q}_t$ and $\mathbf{k}_t = (\mathbf{u}_{t-1} - \mathbf{u}_t)/\mathbf{q}_t$.

(2) Compute $\mathbf{v} = \left[\mathbf{p}_{zt} \cdot \mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{p}_{zt,t} \cdot \mathbf{k}_t\right]_q$, where $\mathbf{p}_{zt} = \left[\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}\right]_q$ and $\mathbf{p}_{zt,t} = \left[\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,t,i}\right]_q$.

(3) Checked whether $\|\mathbf{v}\|$ is short:

$$\mathrm{isZero}_2\left(\mathrm{par}_2, \mathbf{u}_0, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \begin{cases} 1 & \text{if } \|\mathbf{v}\| < q/2^{\eta} \\ 0 & \text{otherwise} \end{cases}.$$

Extract: $sk \leftarrow \mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}_0, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$.

Given a level-k encoding $\mathbf{u}_0$ and $\mathbf{r}_i \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$, extract a bit string as follows:

(1) For $t=1$ to $\mu-1$

Compute $\mathbf{u}_t = \mathbf{u}_{t-1} \bmod \mathbf{q}_t$ and $\mathbf{k}_t = (\mathbf{u}_{t-1} - \mathbf{u}_t)/\mathbf{q}_t$.

(2) Compute $\mathbf{v} = \left[\mathbf{p}_{zt} \cdot \mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{p}_{zt,t} \cdot \mathbf{k}_t\right]_q$, where $\mathbf{p}_{zt} = \left[\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,i}\right]_q$ and $\mathbf{p}_{zt,t} = \left[\sum_{i=1}^{\tau} \mathbf{r}_i \mathbf{p}_{zt,t,i}\right]_q$.

(3) Extract the most significant bits $\mathrm{Ext}_2(\mathrm{par}_2, \mathbf{u}_0, \{\mathbf{r}_i\}_{i=1}^{\tau}) = \mathrm{Extract}(\mathrm{msbs}_{\eta}(\mathbf{v}))$.

Remark 4.1 (1) If one sets $m = O(\lambda^2)$ and $n=1$ for our variant, then the variant is extended to the ring $\mathbb{Z}$ of integers. In some sense this case is similar to the construction in [CLT15], however our construction is ideal multilinear maps, whereas their construction is approximate multilinear maps. Moreover, our variant needs to use the method of constructing new zero-testing parameter in [Gu15]. (2) One cannot set $m = O(\lambda^2)$ and $n=1$ for the ideal multilinear maps in Section 3. This is because the multiplier $\mathbf{q}_0$ is an integer and can be computed in this case. (3) One can set $m=1$ and $n = O(\lambda^2)$ for our variant to decrease time of computing inverse elements in generating instance algorithm.

4.2 Correctness

We first give Lemma 4.2 to show the correctness of our variant construction.

Lemma 4.2 If $\|\mathbf{p}^{-1}\| \le n(\|\mathbf{p}\|)^{-1}$ and $\|\mathbf{u}\|/\|\mathbf{p}\| \le \alpha$, then $\mathbf{u} \bmod \mathbf{p} = \mathbf{u} - \mathbf{k} \cdot \mathbf{p}$ such that $\|\mathbf{k}\| \le n^2(\alpha+1)$.

Proof. Since $\mathbf{u} \bmod \mathbf{p} = \mathbf{u} - \mathbf{k} \cdot \mathbf{p}$, then $\mathbf{k} = \mathbf{p}^{-1}(\mathbf{u} - \mathbf{u} \bmod \mathbf{p})$. So , we have

$$\begin{aligned}\|\mathbf{k}\| &= \|\mathbf{p}^{-1}(\mathbf{u}-\mathbf{u} \bmod \mathbf{p})\| \\ &\le n\|\mathbf{p}^{-1}\|\|(\mathbf{u}-\mathbf{u} \bmod \mathbf{p})\| \\ &\le n\|\mathbf{p}^{-1}\|(\|\mathbf{u}\|_\infty + \|\mathbf{u} \bmod \mathbf{p}\|) \\ &\le n\|\mathbf{p}^{-1}\|(\|\mathbf{u}\| + \|\mathbf{p}\|) \\ &\le n\|\mathbf{p}^{-1}\|(\alpha\|\mathbf{p}\| + \|\mathbf{p}\|) \\ &\le n^2(\alpha+1)\end{aligned}$$

The proof is complete. ■

Similar as that in the construction of ideal multilinear maps, it is easy to prove that $\mathrm{InstGen}_2$, Enc_2, Add_2, Mul_2 are correct. Here we only require to prove that isZero_2 and Ext_2 are correct.

Lemma 4.3 For an arbitrary integer $k > 0$, the zero-testing algorithm $\mathrm{isZero}_2(\mathrm{par}_2, \mathbf{u}_0)$ correctly determines whether a level-k encoding $\mathbf{u}_0$ is an encoding of zero.

Proof. Given an arbitrary level-k encoding $\mathbf{u}_0$, we have $\mathbf{u}_0 = \mathbf{d} + \mathbf{r}\cdot\mathbf{f}$ and $\|\mathbf{u}_0\| < \|\mathbf{q}\|$ with $\|\mathbf{d}\| < \|\mathbf{f}\|$.

For $t \in [\![\mu - 1]\!]$, $\mathbf{u}_t = \mathbf{u}_{t-1} \bmod \mathbf{q}_t$, then $\|\mathbf{u}_t\| \le \|\mathbf{q}_t\|$. So, we have

$$\|\mathbf{u}_{t-1}\| / \|\mathbf{q}_t\| \le \|\mathbf{q}_{t-1}\| / \|\mathbf{q}_t\| \le n\sigma .$$

By Lemma 4.2 and $\|\mathbf{q}_t^{-1}\| \le n(\|\mathbf{q}_t\|)^{-1}$, we have $\|\mathbf{k}_t\| \le n^2(n\sigma+1)$.

For $t \in [\![\mu - 1]\!]$, $\mathbf{u}_t = \mathbf{u}_{t-1} \bmod \mathbf{q}_t$, $\mathbf{k}_t = (\mathbf{u}_{t-1} - \mathbf{u}_t)/\mathbf{q}_t$, then $\mathbf{u}_t = \mathbf{u}_{t-1} - \mathbf{k}_t \cdot \mathbf{q}_t$.

So, we have

$$\mathbf{u}_0 = \mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{q}_t = \mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot (\mathbf{e}_t + \mathbf{p}_t \cdot \mathbf{f}) .$$

Namely,

$$\mathbf{u}_0 \bmod \mathbf{f} = (\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot (\mathbf{e}_t + \mathbf{p}_t \cdot \mathbf{f})) \bmod \mathbf{f} = (\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{e}_t) \bmod \mathbf{f} ,$$

$$\mathbf{u}_0 \bmod \mathbf{f}_j = (\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{e}_{t,j}) \bmod \mathbf{f}_j .$$

Moreover, we have

$$\begin{aligned}&\left\|\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{e}_t\right\| \\ \le &\left\|\mathbf{u}_{\mu-1}\right\| + \sum_{t=1}^{\mu-1} \|\mathbf{k}_t \cdot \mathbf{e}_t\| \\ \le &\left\|\mathbf{u}_{\mu-1}\right\| + n\sum_{t=1}^{\mu-1} \|\mathbf{k}_t\|\|\mathbf{e}_t\| \\ \le &\left\|\mathbf{e}_{\mu-1} + \mathbf{p}_{\mu-1}\cdot\mathbf{f}\right\| + n\sum_{t=1}^{\mu-1} \|\mathbf{k}_t\|\|\mathbf{f}\| \\ \le &(\left\|\mathbf{p}_{\mu-1}\right\| + 1)\|\mathbf{f}\| + \mu n^3(n\sigma+1)\|\mathbf{f}\| \\ \le &\, 2\mu n^4 \sigma\beta\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\left\|\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\mathbf{k}_t\cdot\mathbf{e}_{t,j}\right\| \\
&\le\left\|\mathbf{u}_{\mu-1}\right\|+\sum_{t=1}^{\mu-1}\left\|\mathbf{k}_t\cdot\mathbf{e}_{t,j}\right\| \\
&\le(n\sigma+1)\|\mathbf{f}\|+n\mu\|\mathbf{k}_t\|\left\|\mathbf{e}_{t,j}\right\| \\
&\le 2n\sigma\beta+n^3\sigma^2\mu
\end{aligned}.$$

$$\begin{aligned}
\mathbf{p}_{zt}&=\left[\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{p}_{zt,i}\right]_q \\
&=\left[\sum_{i=1}^{\tau}\mathbf{r}_i\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{a}_{i,j}+\mathbf{s}_{0,i,j}\cdot\mathbf{f}_j)\cdot(\mathbf{f}_j^{-1}\bmod q)\right]_q \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j(\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{a}_{i,j}+\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{s}_{0,i,j}\cdot\mathbf{f}_j)\cdot(\mathbf{f}_j^{-1}\bmod q)\right]_q, \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{a}_j+\mathbf{s}_{0,j}\cdot\mathbf{f}_j)\cdot(\mathbf{f}_j^{-1}\bmod q)\right]_q
\end{aligned}$$

$$\begin{aligned}
\mathbf{p}_{zt,t}&=\left[\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{p}_{zt,t,i}\right]_q \\
&=\left[\sum_{i=1}^{\tau}(\mathbf{r}_i\cdot\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{e}_{t,j}\mathbf{a}_{i,j}+\mathbf{s}_{t,i,j}\mathbf{f}_j)(\mathbf{f}_j^{-1}\bmod q))\right]_q \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j(\sum_{i=1}^{\tau}\mathbf{e}_{t,j}\mathbf{r}_i\mathbf{a}_{i,j}+\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{s}_{t,i,j}\mathbf{f}_j)(\mathbf{f}_j^{-1}\bmod q)\right]_q, \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{e}_{t,j}\mathbf{a}_j+\mathbf{s}_{t,j}\mathbf{f}_j)(\mathbf{f}_j^{-1}\bmod q)\right]_q
\end{aligned}$$

where $\mathbf{a}_j=\sum_{i=1}^{\tau}\mathbf{r}_i\mathbf{a}_{i,j}$, $\mathbf{s}_{t,j}=\sum_{t=0}^{\mu-1}\mathbf{r}_i\mathbf{s}_{t,i,j}$.

Thus, we have

$$\begin{aligned}
&\mathbf{v} \\
&=\left[\mathbf{p}_{zt}\cdot\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\mathbf{p}_{zt,t}\cdot\mathbf{k}_t\right]_q \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{a}_j+\mathbf{s}_{0,j}\cdot\mathbf{f}_j)\cdot\left[\mathbf{f}_j^{-1}\right]_q\cdot\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\sum_{j=1}^{m}\mathbf{h}_j(\mathbf{e}_{t,j}\mathbf{a}_j+\mathbf{s}_{t,j}\mathbf{f}_j)\left[\mathbf{f}_j^{-1}\right]_q\cdot\mathbf{k}_t\right]_q, \\
&=\left[\sum_{j=1}^{m}\mathbf{h}_j\mathbf{a}_j(\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\mathbf{k}_t\mathbf{e}_{t,j}+\mathbf{s}_j\mathbf{f}_j)\cdot\left[\mathbf{f}_j^{-1}\right]_q\right]_q
\end{aligned}$$

where $\mathbf{s}_j=\sum_{t=0}^{\mu-1}\mathbf{k}_t\mathbf{s}_{t,j}$.

(1) If $\mathbf{u}_0$ is an encoding of zero, then $\mathbf{u}_0\bmod\mathbf{f}_j=0, j\in[\![m]\!]$, $\mathbf{u}\bmod\mathbf{f}=0$ and $\mathbf{d}=0$. So, $\mathbf{u}_0\bmod\mathbf{f}_j=(\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\mathbf{k}_t\cdot\mathbf{e}_{t,j})\bmod\mathbf{f}_j=0$, $\mathbf{u}_{\mu-1}+\sum_{t=1}^{\mu-1}\mathbf{k}_t\cdot\mathbf{e}_{t,j}=\mathbf{t}_j\mathbf{f}_j$.

Namely, $\mathbf{v}=\left[\sum_{j=1}^{m}\mathbf{h}_j\mathbf{a}_j(\mathbf{t}_j+\mathbf{s}_j)\right]_q=\sum_{j=1}^{m}\mathbf{h}_j\mathbf{a}_j(\mathbf{t}_j+\mathbf{s}_j)$.

Thus, we have

$$
\begin{aligned}
&\|\mathbf{v}\| \\
&= \left\| \sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j (\mathbf{t}_j + \mathbf{s}_j) \right\| \\
&\le m \left\| \mathbf{h}_j \mathbf{a}_j (\mathbf{t}_j + \mathbf{s}_j) \right\| \\
&\le mn^2 \left\| \mathbf{h}_j \right\| \left\| \mathbf{a}_j \right\| (\left\| \mathbf{t}_j \right\| + \left\| \mathbf{s}_j \right\|) \\
&\le mn^2 \cdot n\sigma \cdot \tau n^2 \sigma^2 \cdot 2(2n\sigma\beta + n^3\sigma^2\mu) \\
&\le n^{O(1)} \\
&\le q / 2^{\eta}
\end{aligned}
$$

(2) If $\mathbf{u}_0$ is not an encoding of zero, then $\mathbf{u}_0 \bmod \mathbf{f} \neq 0$. That is, $\exists j \in [\![m]\!], \mathbf{d} \bmod \mathbf{f}_j \neq 0$. So, $\mathbf{u}_0 \bmod \mathbf{f}_j = (\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{e}_{t,j}) \bmod \mathbf{f}_j = \mathbf{d} \bmod \mathbf{f}_j$, $\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \cdot \mathbf{e}_{t,j} = \mathbf{t}_j \mathbf{f}_j + \mathbf{d}$.

Thus, we have

$$
\begin{aligned}
\|\mathbf{v}\| &= \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j (\mathbf{u}_{\mu-1} + \sum_{t=1}^{\mu-1} \mathbf{k}_t \mathbf{e}_{t,j} + \mathbf{s}_j \mathbf{f}_j) \cdot \left[\mathbf{f}_j^{-1} \right]_q \right]_q \right\| \\
&= \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \left[\mathbf{f}_j^{-1} \right]_q \cdot (\mathbf{d} + \mathbf{t}_j \mathbf{f}_j + \mathbf{s}_j \mathbf{f}_j) \right]_q \right\| \\
&= \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{d} \left[\mathbf{f}_j^{-1} \right]_q + \sum_{j=1}^{m} \mathbf{h}_j (\mathbf{t}_j + \mathbf{s}_j) \right]_q \right\| \\
&\ge \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{d} \left[\mathbf{f}_j^{-1} \right]_q \right]_q \right\| - \left\| \left[\sum_{j=1}^{m} \mathbf{h}_j (\mathbf{t}_j + \mathbf{s}_j) \right]_q \right\| \\
&\ge q^{1-\varepsilon} - q / 2^{\eta} \\
&\ge q^{1-\varepsilon'}
\end{aligned}
$$

Thus, $\mathrm{isZero}_2(\mathrm{par}_2, \mathbf{u}_0)$ can correctly decide whether the encoding of $\mathbf{u}_0$ is zero. ■

Lemma 4.4 If two level-k encodings $\mathbf{u}_0^{(1)}, \mathbf{u}_0^{(2)}$ encode same level-0 element, then $\mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}_0^{(1)}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}_0^{(2)}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$.

Proof. Since $\mathbf{u}_0^{(1)} = \mathbf{u}_0^{(2)} = \mathbf{a}_j (\mathbf{g}_j)^k \bmod \mathbf{f}_j, \forall j \in [\![m]\!]$ and $\mathbf{f}_j, j \in [\![m]\!]$ are co-prime, we have $\mathbf{u}_0^{(1)} = \mathbf{d} + \mathbf{r}^{(1)} \cdot \mathbf{f}, \mathbf{u}_0^{(2)} = \mathbf{d} + \mathbf{r}^{(2)} \cdot \mathbf{f}$.

For $t \in [\![\mu - 1]\!]$, $\mathbf{u}_t^{(1)} = \mathbf{u}_{t-1}^{(1)} \bmod \mathbf{q}_t$, $\mathbf{k}_t^{(1)} = (\mathbf{u}_{t-1}^{(1)} - \mathbf{u}_t^{(1)}) / \mathbf{q}_t$, we have $\mathbf{u}_t^{(1)} = \mathbf{u}_{t-1}^{(1)} - \mathbf{k}_t^{(1)} \cdot \mathbf{q}_t$.

So,

$$
\mathbf{u}_0^{(1)} = \mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot \mathbf{q}_t = \mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot (\mathbf{e}_t + \mathbf{p}_t \cdot \mathbf{f}).
$$

$$
\begin{aligned}
&\mathbf{u}_0^{(1)} \bmod \mathbf{f} \\
&= (\mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot (\mathbf{e}_t + \mathbf{p}_t \cdot \mathbf{f})) \bmod \mathbf{f} \\
&= \mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot \mathbf{e}_t) \bmod \mathbf{f} \\
&= \mathbf{d} \bmod \mathbf{f}
\end{aligned}
,
$$

$$
\mathbf{u}_0^{(1)} \bmod \mathbf{f}_j = (\mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot \mathbf{e}_{t,j}) \bmod \mathbf{f}_j = \mathbf{d} \bmod \mathbf{f}_j.
$$

Namely, we get $\mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \cdot \mathbf{e}_{t,j} = \mathbf{d} + \mathbf{t}_j^{(1)} \mathbf{f}_j$.

So, we have

$$\begin{aligned}
\mathbf{v}^{(1)} &= \left[\mathbf{p}_{zt} \cdot \mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{p}_{zt,t} \cdot \mathbf{k}_t^{(1)} \right]_q \\
&= \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j (\mathbf{u}_{\mu-1}^{(1)} + \sum_{t=1}^{\mu-1} \mathbf{k}_t^{(1)} \mathbf{e}_{t,j} + \mathbf{s}_j^{(1)} \mathbf{f}_j) \cdot \left[\mathbf{f}_j^{-1} \right]_q \right]_q \\
&= \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \left[\mathbf{f}_j^{-1} \right]_q \cdot (\mathbf{d} + \mathbf{t}_j^{(1)} \mathbf{f}_j + \mathbf{s}_j^{(1)} \mathbf{f}_j) \right]_q \\
&= \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \left[\mathbf{f}_j^{-1} \right]_q \cdot (\mathbf{d} + (\mathbf{t}_j^{(1)} + \mathbf{s}_j^{(1)}) \mathbf{f}_j) \right]_q
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\mathbf{v}^{(2)} &= \left[\mathbf{p}_{zt} \cdot \mathbf{u}_{\mu-1}^{(2)} + \sum_{t=1}^{\mu-1} \mathbf{p}_{zt,t} \cdot \mathbf{k}_t^{(2)} \right]_q \\
&= \left[\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \left[\mathbf{f}_j^{-1} \right]_q \cdot (\mathbf{d} + (\mathbf{t}_j^{(2)} + \mathbf{s}_j^{(2)}) \mathbf{f}_j) \right]_q
\end{aligned}$$

By Lemma 4.3, we have

$$\left\| \sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \cdot (\mathbf{t}_j^{(1)} + \mathbf{s}_j^{(1)}) \right\| \le q / 2^{\eta},$$

$$\left\| \sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \cdot (\mathbf{t}_j^{(2)} + \mathbf{s}_j^{(2)}) \right\| \le q / 2^{\eta}.$$

Hence, the most significant bits of $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are decided by $\sum_{j=1}^{m} \mathbf{h}_j \mathbf{a}_j \mathbf{d} \cdot \left[\mathbf{f}_j^{-1} \right]_q$.

Namely, $\mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}_0^{(1)}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right) = \mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}_0^{(2)}, \{\mathbf{r}_i\}_{i=1}^{\tau}\right)$. ■

4.3 Security

Variant security depends on new assumptions, and the extraction variant of GCDH/GDDH is called ext-GCDH/ext-GDDH. The ext-GCDH/ext-GDDH is introduced in [LSS14] to prove the security of the GGHLite scheme. We describe the security experiment of our variant below:

(1) $(\mathrm{par}_2) \leftarrow \mathrm{InstGen}_2(1^{\lambda})$.

(2) Choose an arbitrary positive integer k.

(3) For $t = 0$ to k:

Sample $\mathbf{w}_{t,i} \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [\![\tau]\!]$,

Generate level-1 encoding $\mathbf{u}_t = (\sum_{i=1}^{\tau} \mathbf{w}_{t,i} \cdot \mathbf{y}_i) \bmod \mathbf{q}$.

(4) Sample $\mathbf{r}_{0,i} \leftarrow D_{\mathbb{Z}^n,\sigma'}, i \in [\![\tau]\!]$.

(5) Compute $\mathbf{u} = \prod_{t=1}^{k} \mathbf{u}_t \bmod \mathbf{q}$.

(6) Set $\mathbf{v}_C = \mathbf{v}_D = \mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}, \{\mathbf{w}_{0,i}\}_{i=1}^{\tau}\right)$.

(7) Set $\mathbf{v}_R = \mathrm{Ext}_2\left(\mathrm{par}_2, \mathbf{u}, \{\mathbf{r}_{0,i}\}_{i=1}^{\tau}\right)$..

Definition 4.5 The extraction k-graded CDH problem (ext-GCDH) is, on input $\{\mathrm{par}_2, \mathbf{u}_0, \cdots, \mathbf{u}_k\}$, to output an extraction encoding $\mathbf{v}_C$. The extraction k-graded DDH problem (ext-GDDH) distinguishes between $\mathbf{v}_D$ and $\mathbf{v}_R$, that is, between the distributions

$D_{GDDH} = \{\text{par}_2, \mathbf{u}_0, \cdots, \mathbf{u}_k, \mathbf{v}_D\}$ and $D_{RAND} = \{\text{par}_2, \mathbf{u}_0, \cdots, \mathbf{u}_k, \mathbf{v}_R\}$.

4.4 Cryptanalysis

4.4.1 Easily computable quantities

Since $\mathbf{q} = \mathbf{p}_0 \cdot \mathbf{f}$, one can compute $\det(Rot(\mathbf{q}))$. If one can factor $\det(Rot(\mathbf{q}))$ to get $\det(Rot(\mathbf{q}_0))$ and $p_j = \det(Rot(\mathbf{f}_j))$, then one obtain principal ideal lattice generated by two elements (p_j, α_j), and reduce the principal ideal lattice to find $\mathbf{f}_j$. However, there currently exists efficient algorithm which computes small generator of principal ideal lattice.

Because $\mathbf{q}_i = \mathbf{e}_i + \mathbf{p}_i \cdot \mathbf{f}$, $\mathbf{p}_{zt,i} = \left[\sum_{j=1}^{m} \mathbf{h}_j (\mathbf{e}_{i,j} + \mathbf{r}_{i,j}\mathbf{f}_j)(\left[\mathbf{f}_j^{-1}\right]_q)\right]_q$, $i \in [\![\mu - 1]\!]$, one can compute cross-multiplying $\mathbf{z}_{i_1,i_2} = \left[\mathbf{q}_{i_1}\mathbf{p}_{zt,i_2} - \mathbf{q}_{i_2}\mathbf{p}_{zt,i_1}\right]_q$. It is easy to see that $\mathbf{z}_{i_1,i_2}$ is not reduced modulo q for some α such that $i_1, i_2 \geq \alpha \geq 1$. However, one cannot get common factor from $\mathbf{z}_{i_1,i_2}$.

5 Ideal multilinear maps using unimodular matrix

For the above ideal constructions, there are mainly two problems: (1) One can compute a basis of secret ideal lattice using the public parameters in Section 3; (2) One can only generate usable plaintext in level-1 encoding, because the plaintext of encoding is included in corresponding zero-testing parameter, and is unknown.

In this section, we introduce unimodular matrices to solve the above problems. The reason using unimodular matrix is our construction work over the integers. We apply different unimodular matrices for level-0 encoding and level-1 encoding to thwart cross-multiplying of encodings.

5.1 Construction

Setting the parameters. We let λ be the security parameter and n be the dimension of polynomial ring R. Concrete parameters are set as $\sigma = \sqrt{\lambda n}$, $\sigma' = \lambda n^{1.5}$, $n = O(\lambda^2)$, $m = O(\lambda)$, $\xi = \lambda^{O(m)}$, $q \geq mn^{21}\xi^2 2^{\eta}$, and $\tau = O(n^2)$.

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

(1) Select a sufficiently large prime q.

(2) Generate parameters in the inner layer:

(2.1) Sample $\mathbf{f}_j \leftarrow D_{\mathbb{Z}^n,\sigma}$, $j \in [\![m]\!]$, and $\mathbf{p}_0 \leftarrow D_{\mathbb{Z}^n,\sigma'}$, such that $\mathbf{f}_j$ is a co-prime, $\mathbf{f}_j^{-1} \in \mathbb{k}$, and $\left\|\mathbf{f}_j^{-1}\right\| \leq O(n^2)$.

(2.2) Set $\mathbf{f} = \prod_{j=1}^{m} \mathbf{f}_j$, $\mathbf{q} = \mathbf{p}_0 \cdot \mathbf{f}$, such that matrix $Rot(\mathbf{q})$ is invertible.

(2.3) Sample $\mathbf{g} \leftarrow D_{\mathbb{Z}^n,\mu}$, $\mathbf{a}_i \leftarrow D_{\mathbb{Z}^n,\mu}$, $\mathbf{e}_i, \mathbf{t}_i \leftarrow D_{\mathbb{Z}^n,\sigma'}$, $i \in [\![\tau]\!]$, where $\mu = \|\mathbf{f}\|$.

(2.4) Set $\mathbf{x}_i = (\mathbf{a}_i + \mathbf{e}_i\mathbf{f}) \bmod \mathbf{q}$, $\mathbf{y}_i = (\mathbf{a}_i\mathbf{g} + \mathbf{t}_i\mathbf{f}) \bmod \mathbf{q}$, $i \in [\![\tau]\!]$.

(2.5) Set $\mathbf{p}_{zt} = (\sum_{j=1}^{m} \mathbf{h}_j \cdot (\mathbf{f}_j^{-1} \bmod q)) \bmod q$, where $\mathbf{h}_j \leftarrow D_{\mathbb{Z}^n,\sigma}$, $j \in [\![m]\!]$.

(2.6) Sample $\mathbf{m}_t, \mathbf{n}_t \leftarrow D_{\mathbb{Z}^n,\sigma}$, $t \in [\![n]\!]$, such that $\mathbf{M}' = (\mathbf{m}_1\mathbf{q}, \cdots, \mathbf{m}_n\mathbf{q})$ and $\mathbf{N}' = (\mathbf{n}_1\mathbf{q}, \cdots, \mathbf{n}_n\mathbf{q})$ are invertible.

(3) Generate parameters in the outer layer:

(3.1) Sample randomly unimodular matrices $\mathbf{S}$ and $\mathbf{T}$ such that $\|\mathbf{S}\| = \|\mathbf{T}\| \le O(n^2)$.

(3.2) Set $\mathbf{X}_i = \mathbf{S}Rot(\mathbf{x}_i)\mathbf{S}^{-1}$, $\mathbf{Y}_i = \mathbf{T}^{-1}Rot(\mathbf{y}_i)\mathbf{T}$, $i \in [\![\tau]\!]$.

(3.3) Set $\mathbf{M}_t = \mathbf{S}Rot(\mathbf{m}_t\mathbf{q})\mathbf{S}^{-1}$, $\mathbf{N}_t = \mathbf{T}^{-1}Rot(\mathbf{n}_t\mathbf{q})\mathbf{T}$, $t \in [\![n]\!]$. We denote $\mathbf{M}, \mathbf{N} \in \mathbb{Z}^{n^2 \times n}$ as the matrix of column vectors $\mathbf{M}_t$ and $\mathbf{N}_t$, where $\mathbf{M}_t$ and $\mathbf{N}_t$ are considered n^2-dimensional column vectors. We let $\|\mathbf{M}\| = \|\mathbf{N}\| = \xi$.

(4) Generate the parameters of zero-testing and extraction:

(4.1) Sample $\mathbf{s}, \mathbf{t} \leftarrow D_{\mathbb{Z}^n,\sigma}$.

(4.2) Randomly select $\mathbf{z}_s, \mathbf{z}_t \in R_q$, such that $(\mathbf{z}_s)^{-1}, (\mathbf{z}_t)^{-1} \in R_q$.

(4.3) Set $\mathbf{s}^* = \mathbf{s}^T Rot(\mathbf{z}_s)^{-1}\mathbf{S}^{-1}$ and $\mathbf{t}^* = \mathbf{T}^{-1}Rot(\mathbf{z}_t)^{-1}\mathbf{t}$.

(4.4) Set $\mathbf{P}_{zt} = [\mathbf{S}Rot(\mathbf{z}_s\mathbf{z}_t\mathbf{p}_{zt})\mathbf{T}]_q$.

(5) Output the public parameters $\mathrm{par}_3 = \{q, \mathbf{M}, \mathbf{N}, \{\mathbf{X}_i, \mathbf{Y}_i\}_{i \in [\![\tau]\!]}, \mathbf{P}_{zt}, \mathbf{s}^*, \mathbf{t}^*\}$.

Generating level-k random encodings: $\mathbf{U} \leftarrow \mathrm{Enc}_3(\mathrm{par}_3, k, \{w_i\}_{i \in [\![\tau]\!]})$.

Select $w_i \leftarrow D_{\mathbb{Z},\sigma'}$, $i \in [\![\tau]\!]$ and generate a Level 0 encoding $\mathbf{D} = \sum_{i=1}^{\tau} w_i \cdot (\mathbf{X}_i)^k \bmod \mathbf{M}$ and a level-k encoding $\mathbf{U} = \sum_{i=1}^{\tau} w_i \cdot (\mathbf{Y}_i)^k \bmod \mathbf{N}$.

Adding encodings: $\mathbf{U} \leftarrow \mathrm{Add}_3(\mathrm{par}_3, k, \mathbf{U}_1, \cdots, \mathbf{U}_\beta)$.

Given β level-k encodings $\mathbf{U}_\alpha$, $\alpha \in [\![\beta]\!]$, their sum $\mathbf{U} = \sum_{\alpha=1}^{\beta} \mathbf{U}_\alpha \bmod \mathbf{N}$ is a level-k encoding.

Multiplying encodings: $\mathbf{U} \leftarrow \mathrm{Mul}_3(\mathrm{par}_3, 1, \mathbf{U}_1, \cdots, \mathbf{U}_k)$.

Given k level-1 encodings $\mathbf{U}_\alpha$, $\alpha \in [\![k]\!]$, their product $\mathbf{U} = \prod_{\alpha=1}^{k} \mathbf{U}_\alpha \bmod \mathbf{N}$ is a level-k encoding.

Zero-testing: $\mathrm{isZero}_3(\mathrm{par}_3, \mathbf{D}, \mathbf{U})$.

Given a level-k encoding $\mathbf{U}$ and a level-0 encoding $\mathbf{D}$, we determine whether $\mathbf{U}$ is an encoding of zero. We compute $v = [\mathbf{s}^* \cdot \mathbf{D} \cdot \mathbf{P}_{zt} \cdot \mathbf{U} \cdot \mathbf{t}^*]_q$ in $\mathbb{Z}_q$ and check whether v is small, as follows:

$$\mathrm{isZero}_3(\mathrm{par}_3, \mathbf{D}, \mathbf{U}) = \begin{cases} 1 & \text{if } |v| < q/2^\eta \\ 0 & \text{otherwise} \end{cases}.$$

Extract: $sk \leftarrow \mathrm{Ext}_3(\mathrm{par}_3, \mathbf{D}, \mathbf{U})$.

Given a level-k encoding $\mathbf{U}$ and a level-0 encoding $\mathbf{D}$, $\mathrm{Ext}_3(\mathrm{par}, \mathbf{D}, \mathbf{U}) = \mathrm{Extract}_s(\mathrm{msbs}_\gamma([\mathbf{s}^* \cdot \mathbf{D} \cdot \mathbf{P}_{zt} \cdot \mathbf{U} \cdot \mathbf{t}^*]_q))$, where msbs_γ extracts the $\gamma = \eta - \lambda$ most significant bits from the result. $\mathrm{Extract}_s$ is a strong randomness extractor using the seed s.

Remark 5.1 (1) We can set $m=1$, that is, we can regard $\mathbf{f}=\prod_{j=1}^{m}\mathbf{f}_j$ as one ring element. Given that the final results generated by the zero-testing and extraction algorithms are integers, we do not know how to attack our construction by using these integers. (2) We must set $\tau \geq n^2+\lambda$ to prevent the algebraic equation attack. (3) The matrix $\mathbf{D}$ can be taken an identity matrix. Our aim is to demonstrate how to use level-0 encodings when constructing the MPKE protocol. (4) When our construction is applied to multipartite Diffie–Hellman key exchange, $\mathbf{P}_{zt}, \mathbf{X}_i$ in the public parameters can be replaced by $\mathbf{P}_{zt,i}=\left[\mathbf{X}_i\mathbf{P}_{zt}\right]_q$ and $\mathbf{S}$ does not require a unimodular matrix. Furthermore, the matrix $\mathbf{P}_{zt,i}=\left[\mathbf{X}_i\mathbf{P}_{zt}\right]_q$ may be further modified into vector $\mathbf{p}_{zt,i}=\left[\mathbf{s}^*\mathbf{X}_i\mathbf{P}_{zt}\right]_q$.

5.2 Correctness

Lemma 5.2 $\mathrm{InstGen}_3(1^\lambda)$ is a probabilistic polynomial time algorithm.

Proof. Unimodular matrices $\mathbf{S}$ and $\mathbf{T}$ can be generated by the method of [GGH15]. All other elements in $\mathrm{InstGen}(1^\lambda)$ can be computed in polynomial time. ■

Lemma 5.3 The ranks of $\mathbf{M}$ and $\mathbf{N}$ in the public parameter par_3 are n.

Proof. We prove the result by contradiction and assume that the rank of $\mathbf{M}$ is less than n. Without loss of generality, assume that there exist n non-all-zero real numbers k_t such that $\sum_{t=1}^{n}k_t\mathbf{M}_t=\mathbf{0}^{n\times n}$. Thus, we derive the following expression:

$$\sum_{t=1}^{n}k_t\mathbf{M}_t=\mathbf{S}(\sum_{t=1}^{n}k_t Rot(\mathbf{m}_t\mathbf{q}))\mathbf{S}^{-1}=\mathbf{0}^{n\times n}.$$

Given that $\mathbf{S}, Rot(\mathbf{q})$ are invertible over $\mathbb{R}$, we derive $\sum_{t=1}^{n}k_t Rot(\mathbf{m}_t\mathbf{q})=\mathbf{0}^{n\times n}$, that is, $\sum_{t=1}^{n}k_t\mathbf{m}_t\mathbf{q}=\mathbf{0}^{n}$.

On the basis of the rank n of $\mathbf{M}'=(\mathbf{m}_1\mathbf{q},\cdots,\mathbf{m}_n\mathbf{q})$, a contradiction is generated. Similarly, we can prove that the rank of $\mathbf{N}$ is n. ■

Lemma 5.4 If we assume that $Space(\mathbf{M})$ and $Space(\mathbf{N})$ are linear spaces spanned by $\mathbf{M}$ and $\mathbf{N}$, respectively, then $\mathbf{X}_i\in Space(\mathbf{M})$ and $\mathbf{Y}_i\in Space(\mathbf{N})$ for $\{\mathbf{X}_i\}_{i\in[\![\tau]\!]},\{\mathbf{Y}_i\}_{i\in[\![\tau]\!]}$ in the public parameter par.

Proof. Given that $\mathbf{M}'$ is invertible, vector $\mathbf{k}=(\mathbf{M}')^{-1}\cdot\mathbf{x}_i$ for $\mathbf{x}_i$ of $\mathbf{X}_i=\mathbf{S}Rot(\mathbf{x}_i)\mathbf{S}^{-1}$, that is, $\mathbf{M}'\cdot\mathbf{k}=\mathbf{x}_i$. Thus, we derive the following expression:

$$\sum_{t=1}^{n}k_t\mathbf{m}_t\mathbf{q}=\mathbf{x}_i.$$

The j-th column of $Rot(\mathbf{x}_i)$ is $\mathbf{x}_i\cdot x^j$, $(\sum_{t=1}^{n}k_t\mathbf{m}_t\mathbf{q})\cdot x^j=\mathbf{x}_i\cdot x^j$. As such, $\mathbf{X}_i=\sum_{t=1}^{n}k_t\mathbf{M}_t$ and $\mathbf{X}_i\in Space(\mathbf{M})$. Similarly, $\mathbf{Y}_i\in Space(\mathbf{N})$. ■

Lemma 5.5 Encoding $\mathbf{U}\leftarrow\mathrm{Enc}_3(\mathrm{par}_3,k,\{w_i\}_{i\in[\![\tau]\!]})$ is a level-k encoding.

Proof. Given that $\mathbf{D}=\sum_{i=1}^{\tau}w_i\cdot(\mathbf{X}_i)^k \bmod \mathbf{M}$, we derive the following expressions:

$$\begin{aligned}\mathbf{D} &= \sum_{i=1}^{\tau} w_i \cdot (\mathbf{X}_i)^k \bmod \mathbf{M} \\ &= \sum_{i=1}^{\tau} w_i \cdot (\mathbf{X}_i)^k - \sum_{t=1}^{n} \alpha_t \cdot \mathbf{M}_t \\ &= \mathbf{S}(\sum_{i=1}^{\tau} w_i \cdot Rot(\mathbf{x}_i)^k - \sum_{t=1}^{n} \alpha_t \cdot Rot(\mathbf{m}_t\mathbf{q}))\mathbf{S}^{-1} \\ &= \mathbf{S}Rot(\mathbf{d})\mathbf{S}^{-1}\end{aligned},$$

$$\begin{aligned}&\mathbf{d} \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{x}_i)^k - \sum_{t=1}^{n} \alpha_t \cdot (\mathbf{m}_t\mathbf{q})) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{x}_i)^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot ((\mathbf{a}_i + \mathbf{e}_i\mathbf{f}) \bmod \mathbf{q})^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{a}_i \bmod \mathbf{f}_j)^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{a}_{i,j})^k) \bmod \mathbf{f}_j\end{aligned}.$$

Given that $\mathbf{U} = \sum_{i=1}^{\tau} w_i \cdot (\mathbf{Y}_i)^k \bmod \mathbf{N}$, we derive the following expressions:

$$\begin{aligned}\mathbf{U} &= \sum_{i=1}^{\tau} w_i \cdot (\mathbf{Y}_i)^k \bmod \mathbf{N} \\ &= \sum_{i=1}^{\tau} w_i \cdot (\mathbf{Y}_i)^k - \sum_{t=1}^{n} \beta_t \cdot \mathbf{N}_t \\ &= \mathbf{T}^{-1}(\sum_{i=1}^{\tau} w_i \cdot Rot(\mathbf{y}_i)^k - \sum_{t=1}^{n} \beta_t \cdot Rot(\mathbf{n}_t\mathbf{q}))\mathbf{T} \\ &= \mathbf{T}^{-1}Rot(\mathbf{u})\mathbf{T}\end{aligned},$$

$$\begin{aligned}&\mathbf{u} \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{y}_i)^k - \sum_{t=1}^{n} \beta_t \cdot (\mathbf{n}_t\mathbf{q})) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{y}_i)^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot ((\mathbf{a}_i\mathbf{g} + \mathbf{t}_i\mathbf{f}) \bmod \mathbf{q})^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{a}_i \bmod \mathbf{f}_j)^k (\mathbf{g} \bmod \mathbf{f}_j)^k) \bmod \mathbf{f}_j \\ &= (\sum_{i=1}^{\tau} w_i \cdot (\mathbf{a}_{i,j})^k \cdot (\mathbf{g}_j)^k) \bmod \mathbf{f}_j \\ &= ((\mathbf{d} \bmod \mathbf{f}_j) \cdot (\mathbf{g}_j)^k) \bmod \mathbf{f}_j\end{aligned}.$$

In the previously presented expressions, we use notations $\mathbf{a}_{i,j} = \mathbf{a}_i \bmod \mathbf{f}_j$ and $\mathbf{g}_j = \mathbf{g} \bmod \mathbf{f}_j$.

As such, $\mathbf{U}$ is a level-k encoding of $\mathbf{D}$. ■

Lemma 5.6 Encoding $\mathbf{U} \leftarrow \mathrm{Add}_3(\mathrm{par}_3, k, \mathbf{U}_1, \cdots, \mathbf{U}_s)$ is a level-k encoding.

Proof. According to the modulo arithmetic rule, the sum $\mathbf{U}$ of level-k encodings $\mathbf{U}_t, t \in [\![s]\!]$ is a level-k encoding. ■

Lemma 5.7 $\mathbf{U} \leftarrow \mathrm{Mul}_3(\mathrm{par}_3, 1, \mathbf{U}_1, \cdots, \mathbf{U}_k)$ is a level-k encoding.

Proof. Given that $\mathbf{U}_t, t \in [\![k]\!]$ are level-1 encodings, we obtain $\mathbf{U}_t = \mathbf{T}^{-1}Rot(\mathbf{u}_t)\mathbf{T}$ and $\mathbf{u}_t = (\mathbf{d}_t\mathbf{g} + \mathbf{r}_t\mathbf{f}) \bmod \mathbf{q}$. As such, we derive the following expressions:

$$\begin{aligned}\mathbf{U} &= \prod_{t=1}^{k}\mathbf{U}_t \bmod \mathbf{N}\\ &= \mathbf{T}^{-1}Rot(\prod_{t=1}^{k}\mathbf{u}_t)\mathbf{T} - \sum_{t=1}^{n}\beta_t\cdot\mathbf{N}_t\\ &= \mathbf{T}^{-1}Rot(\prod_{t=1}^{k}\mathbf{u}_t - \sum_{t=1}^{n}\beta_t\cdot\mathbf{n}_t\mathbf{q})\mathbf{T}\\ &= \mathbf{T}^{-1}Rot(\mathbf{u})\mathbf{T}\end{aligned},$$

$$\begin{aligned}&\mathbf{u} \bmod \mathbf{f}_j\\ &= (\prod_{t=1}^{k}\mathbf{u}_t - \sum_{t=1}^{n}\beta_t\cdot\mathbf{n}_t\mathbf{q}) \bmod \mathbf{f}_j\\ &= (\prod_{t=1}^{k}(\mathbf{d}_t\mathbf{g}+\mathbf{r}_t\mathbf{f}) \bmod \mathbf{f}_j)) \bmod \mathbf{f}_j\\ &= (\prod_{t=1}^{k}\mathbf{d}_{t,j}\mathbf{g}_j) \bmod \mathbf{f}_j\\ &= (\prod_{t=1}^{k}\mathbf{d}_{t,j})(\mathbf{g}_j)^k \bmod \mathbf{f}_j\end{aligned}.$$

Thus, $\mathbf{U}$ is a level-k encoding. ■

Lemma 5.8 For an arbitrary integer $k>0$, the zero-testing algorithm $\text{isZero}_3(\text{par}_3, \mathbf{D}, \mathbf{U})$ correctly determines whether $\mathbf{U}$ is an encoding of zero.

Proof. Given a level-0 encoding $\mathbf{D}$ and an arbitrary level-k encoding $\mathbf{U}$, we obtain $\mathbf{D} = \mathbf{S}Rot(\mathbf{d})\mathbf{S}^{-1}$, $\mathbf{U} = \mathbf{T}^{-1}Rot(\mathbf{u})\mathbf{T}$, $\mathbf{u} = \mathbf{a} + \mathbf{r}\cdot\mathbf{f}$ and $\|\mathbf{d}\| \le n^2\|\mathbf{S}\|\|\mathbf{D}\|\|\mathbf{S}^{-1}\|$, $\|\mathbf{a}\| < \|\mathbf{f}\|$, $\|\mathbf{u}\| \le n^2\|\mathbf{T}\|\|\mathbf{U}\|\|\mathbf{T}^{-1}\|$.

On the basis of $\|\mathbf{D}\| \le \|\mathbf{M}\|$ and $\|\mathbf{U}\| \le \|\mathbf{N}\|$, we obtain $\|\mathbf{d}\| \le n^6\xi$ and $\|\mathbf{u}\| \le n^6\xi$.

(1) If $\mathbf{U}$ is an encoding of zero, then $\mathbf{u} \bmod \mathbf{f}_j = 0, j \in [\![m]\!]$. Given that $\mathbf{f}_j, j \in [\![m]\!]$ are co-primes, $\mathbf{u} \bmod \mathbf{f} = 0$, that is, $\mathbf{a} \bmod \mathbf{f} = 0$ and $\mathbf{a} = 0$ based on $\|\mathbf{a}\| < \|\mathbf{f}\|$. As such, we derive the following expression:

$$\begin{aligned}|v| &= \left|\left\lfloor \mathbf{s}^*\cdot\mathbf{D}\cdot\mathbf{P}_{zt}\cdot\mathbf{U}\cdot\mathbf{t}^*\right\rceil_q\right|\\ &= \left|\mathbf{s}^T Rot(\sum_{j=1}^{m}\mathbf{dh}_j\mathbf{r}\cdot\mathbf{f}/\mathbf{f}_j)\mathbf{t}\right|\\ &\le n^2\|\mathbf{s}\|\left\|Rot(\sum_{j=1}^{m}\mathbf{dh}_j\mathbf{r}\cdot\mathbf{f}/\mathbf{f}_j)\right\|\|\mathbf{t}\|\\ &\le n^2\|\mathbf{s}\|\left\|\sum_{j=1}^{m}\mathbf{dh}_j\mathbf{r}\cdot\mathbf{f}/\mathbf{f}_j\right\|\|\mathbf{t}\|\\ &\le mn^5\cdot\|\mathbf{d}\|\cdot\|\mathbf{h}_j\|\cdot\|\mathbf{rf}\|\cdot\|(\mathbf{f}_j)^{-1}\|\\ &\le mn^5\cdot n^6\xi\cdot n^2\cdot n^6\xi\cdot n^2\\ &\le q/2^{\eta}\end{aligned}.$$

(2) If $\mathbf{U}$ is not an encoding of zero, then $\mathbf{u} \bmod \mathbf{f} \ne 0$, that is, $\exists j \in [\![m]\!], \mathbf{a} \bmod \mathbf{f}_j \ne 0$. As such, we derive the following expression:

$$
\begin{aligned}
|v| &= \left|\left[\mathbf{s}^* \cdot \mathbf{D} \cdot \mathbf{P}_{zt} \cdot \mathbf{U} \cdot \mathbf{t}^*\right]_q\right| \\
&= \left|\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \cdot \left[\mathbf{f}_j^{-1}\right]_q \cdot (\mathbf{a}+\mathbf{rf}))\mathbf{t}\right]_q\right| \\
&= \left\|\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \cdot \left[\mathbf{f}_j^{-1}\right]_q \cdot (\mathbf{a}+\mathbf{rf}))\mathbf{t}\right]_q\right\| \\
&= \left\|\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{a} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t} + \mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{r} \cdot \mathbf{f}/\mathbf{f}_j)\mathbf{t}\right]_q\right\| \\
&\geq \left\|\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{a} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q\right\| - n^2 \left\|\mathbf{s}^T\right\| \left\|Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{r} \cdot \mathbf{f}/\mathbf{f}_j)\right\| \|\mathbf{t}\| \\
&\geq q^{1-\varepsilon} - q/2^{\eta} \\
&\geq q^{1-\varepsilon'}
\end{aligned}
$$

where $\varepsilon, \varepsilon'$ are small positive constants.

As such, $\text{isZero}_3(\text{par}_3, \mathbf{D}, \mathbf{U})$ correctly decides whether $\mathbf{U}$ is an encoding of zero. ■

Lemma 5.9 Suppose that $\mathbf{D} = \mathbf{S}Rot(\mathbf{d})\mathbf{S}^{-1}$ is a level- 0 encoding, and $\mathbf{U}_t = \mathbf{T}^{-1}Rot(\mathbf{u}_t)\mathbf{T}$, $t \in [\![2]\!]$ two level- k encodings. If $\mathbf{U}_t, t \in [\![2]\!]$ encode the same level- 0 elements, namely, $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{a}_j(\mathbf{g}_j)^k \bmod \mathbf{f}_j, \forall j \in [\![m]\!]$, then we derive the following expression:

$$\text{Ext}_3(\text{par}_3, \mathbf{D}, \mathbf{U}_1) = \text{Ext}_3(\text{par}_3, \mathbf{D}, \mathbf{U}_2).$$

Proof. Given that $\mathbf{u}_1 = \mathbf{u}_2 = \mathbf{a}_j(\mathbf{g}_j)^k \bmod \mathbf{f}_j, \forall j \in [\![m]\!]$ and $\mathbf{f}_j, j \in [\![m]\!]$ are co-primes, we obtain $\mathbf{u}_1 = \mathbf{a} + \mathbf{r}_1 \cdot \mathbf{f}, \mathbf{u}_2 = \mathbf{a} + \mathbf{r}_2 \cdot \mathbf{f}$. As such, we derive the following expressions:

$$
\begin{aligned}
v_1 &= \left[\mathbf{s}^* \cdot \mathbf{D} \cdot \mathbf{P}_{zt} \cdot \mathbf{U}_1 \cdot \mathbf{t}^*\right]_q \\
&= \left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \cdot \left[\mathbf{f}_j^{-1}\right]_q \cdot (\mathbf{a}+\mathbf{r}_1\mathbf{f}))\mathbf{t}\right]_q \\
&= \left(\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{a} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q + \left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{r}_1\mathbf{f} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q\right) \bmod q \\
v_2 &= \left[\mathbf{s}^* \cdot \mathbf{D} \cdot \mathbf{P}_{zt} \cdot \mathbf{U}_2 \cdot \mathbf{t}^*\right]_q \\
&= \left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \cdot \left[\mathbf{f}_j^{-1}\right]_q \cdot (\mathbf{a}+\mathbf{r}_2\mathbf{f}))\mathbf{t}\right]_q \\
&= \left(\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{a} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q + \left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{r}_2\mathbf{f} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q\right) \bmod q
\end{aligned}
$$

On the basis of Lemma 3.8, we obtain $\left|\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{r}_t\mathbf{f} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q\right| \leq q/2^{\eta}, t \in [\![2]\!]$. Thus, the $\gamma = \eta - \lambda$ most significant bits of v_1, v_2, which are the same with high probability, are decided on the basis of the first term $\left[\mathbf{s}^T Rot(\textstyle\sum_{j=1}^{m} \mathbf{dh}_j \mathbf{a} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q$. ■

5.3 Security

The security of our constructions depends on new hardness assumptions and cannot be

reduced to classic hard problems, such as lattice hard problem or LWE.

We adaptively extend the definition of ext-GCDH/ext-GDDH in [LSS14] to our construction. Consider the following process:

(1) $(\text{par}_3) \leftarrow \text{InstGen}_3(1^\lambda)$.

(2) Select an arbitrary positive integer k.

(3) For $t = 0$ to k:

Sample $w_{t,i} \leftarrow D_{\mathbb{Z},\sigma'}, i \in [\![\tau]\!]$

Generate level-1 encoding of $\mathbf{D}_t = (\sum_{i=1}^{\tau} w_{t,i} \cdot \mathbf{X}_i) \bmod \mathbf{M}$:

$\mathbf{U}_t = (\sum_{i=1}^{\tau} w_{t,i} \cdot \mathbf{Y}_i) \bmod \mathbf{N}$.

(4) Sample $r_{0,i} \leftarrow D_{\mathbb{Z},\sigma'}, i \in [\![\tau]\!]$ and generate $\mathbf{R}_0 = (\sum_{i=1}^{\tau} r_{0,i} \cdot \mathbf{X}_i) \bmod \mathbf{M}$.

(5) Compute $\mathbf{U} = \prod_{t=1}^{k} \mathbf{U}_t \bmod \mathbf{N}$.

(6) Set $v_C = v_D = \text{Ext}_3(\text{par}_3, \mathbf{D}_0, \mathbf{U})$.

(7) Set $v_R = \text{Ext}_3(\text{par}_3, \mathbf{R}_0, \mathbf{U})$.

Definition 5.10 (ideal-ext-GCDH/ideal-ext-GDDH). The extraction k-graded computational Diffie-Hellman problem (ideal-ext-GCDH) is on input $\{\text{par}, \mathbf{U}_0, \cdots, \mathbf{U}_k\}$ to output an extraction encoding $v_C \in \mathbb{Z}$. The extraction k-graded decisional Diffie-Hellman problem (ideal-ext-GDDH) distinguishes between v_D and v_R, that is, between the distributions $D_{GDDH} = \{\text{par}_3, \mathbf{U}_0, \cdots, \mathbf{U}_k, v_D\}$ and $D_{RAND} = \{\text{par}_3, \mathbf{U}_0, \cdots, \mathbf{U}_k, v_R\}$.

In this study, we assume that the ideal-ext-GCDH/ideal-ext-GDDH is hard.

5.4 Cryptanalysis

5.4.1 Easily Computable Quantities

Given that $\mathbf{M}_t = \mathbf{S}Rot(\mathbf{m}_t\mathbf{q})\mathbf{S}^{-1}$, $\mathbf{N}_t = \mathbf{T}^{-1}Rot(\mathbf{n}_t\mathbf{q})\mathbf{T}$ and $\mathbf{S}$, $\mathbf{T}$ are unimodular matrices, we derive the following expressions:

$$\det(\mathbf{M}_t) = \det(\mathbf{S}Rot(\mathbf{m}_t\mathbf{q})\mathbf{S}^{-1}) = \det(Rot(\mathbf{m}_t\mathbf{q})) = \det(Rot(\mathbf{m}_t))\det(Rot(\mathbf{q})),$$

$$\det(\mathbf{N}_t) = \det(\mathbf{S}Rot(\mathbf{n}_t\mathbf{q})\mathbf{S}^{-1}) = \det(Rot(\mathbf{n}_t\mathbf{q})) = \det(Rot(\mathbf{n}_t))\det(Rot(\mathbf{q})).$$

As such, the determinants $\det(Rot(\mathbf{q}))$, $\det(Rot(\mathbf{m}_t))$, and $\det(Rot(\mathbf{n}_t))$ can be computed by using the GCD algorithm.

Given that $\mathbf{N}_t, t \in [\![n]\!]$ are also encodings of zero, we derive the following expression:

$$\begin{aligned}
\mu_t &= \left[\mathbf{s}^* \mathbf{P}_{zt} \mathbf{N}_t \mathbf{t}^*\right]_q \\
&= \left[\mathbf{s}^T Rot(\mathbf{p}_{zt}\mathbf{n}_t\mathbf{q})\mathbf{t}\right]_q \\
&= \left[\mathbf{s}^T Rot(\sum_{j=1}^{m} \mathbf{h}_j \mathbf{n}_t \mathbf{p}_0 \cdot \mathbf{f} \cdot \left[\mathbf{f}_j^{-1}\right]_q)\mathbf{t}\right]_q \\
&= \left[\mathbf{s}^T Rot(\sum_{j=1}^{m} \mathbf{h}_j \mathbf{n}_t \mathbf{p}_0 \cdot \mathbf{f} / \mathbf{f}_j)\mathbf{t}\right]_q
\end{aligned}.$$

By using our parameter settings, μ_t is not reduced modulo q, that is, $\mu_t = \mathbf{s}^T Rot(\sum_{j=1}^{m} \mathbf{h}_j \mathbf{n}_t \mathbf{p}_0 \cdot \mathbf{f} / \mathbf{f}_j)\mathbf{t}$.

For $\mathbf{M}_t, t \in [\![n]\!]$, we derive $\nu_t = \left[\mathbf{s}^* \mathbf{M}_t \mathbf{P}_{zt} \mathbf{t}^*\right]_q = \mathbf{s}^T Rot(\sum_{j=1}^{m} \mathbf{h}_j \mathbf{m}_t \mathbf{p}_0 \cdot \mathbf{f} / \mathbf{f}_j)\mathbf{t}$.

Similarly, $\theta_{s,t} = \left[\mathbf{s}^* \mathbf{M}_s \mathbf{P}_{zt} \mathbf{N}_t \mathbf{t}^*\right]_q = \mathbf{s}^T Rot(\sum_{j=1}^{m} \mathbf{h}_j \mathbf{m}_s \mathbf{n}_t \cdot \mathbf{q}^2 / \mathbf{f}_j)\mathbf{t}$, where $s,t \in [\![n]\!]$.

By using the cross-multiplication of $\mathbf{X}_i, \mathbf{Y}_i$, we can also obtain the following quantities:

$$\begin{aligned}\pi_{i_1,i_2} &= \left[\mathbf{s}^* (\mathbf{X}_{i_1} \mathbf{P}_{zt} \mathbf{Y}_{i_2} - \mathbf{X}_{i_2} \mathbf{P}_{zt} \mathbf{Y}_{i_1})\mathbf{t}^*\right]_q \\ &= \left[\mathbf{s}^T Rot(\mathbf{p}_{zt}((\mathbf{a}_{i_1} + \mathbf{e}_{i_1}\mathbf{f})(\mathbf{a}_{i_2}\mathbf{g} + \mathbf{t}_{i_2}\mathbf{f}) - (\mathbf{a}_{i_2} + \mathbf{e}_{i_2}\mathbf{f})(\mathbf{a}_{i_1}\mathbf{g} + \mathbf{t}_{i_1}\mathbf{f})))\mathbf{t}\right]_q . \\ &= \left[\mathbf{s}^T Rot(\mathbf{p}_{zt}\mathbf{f}(\mathbf{a}_{i_1}\mathbf{t}_{i_2} + \mathbf{e}_{i_1}\mathbf{a}_{i_2}\mathbf{g} + \mathbf{e}_{i_1}\mathbf{t}_{i_2}\mathbf{f} - \mathbf{a}_{i_2}\mathbf{t}_{i_1} - \mathbf{e}_{i_2}\mathbf{a}_{i_1}\mathbf{g} - \mathbf{e}_{i_2}\mathbf{t}_{i_1}\mathbf{f}))\mathbf{t}\right]_q\end{aligned}$$

For these quantities generated from encodings of zero, building a system of equations is possible. In fact, if we define a function $f_{\mathbf{s},\mathbf{t}}(\mathbf{w}_i) = \mathbf{s}^T \cdot Rot(\mathbf{w}_i) \cdot \mathbf{t}$, then integers, such as u_t, ν_t, $\theta_{s,t}$, and π_{i_1,i_2}, are values of $f_{\mathbf{s},\mathbf{t}}$. Given only several values of $f_{\mathbf{s},\mathbf{t}}$, we cannot solve $\mathbf{s},\mathbf{t}$. The integers in our construction seemingly have the form of the CLT construction [CLT13]. However, the attacks proposed by Cheon et al. [CHL+15] cannot be applied to our construction.

5.4.2 Lattice Reduction Attack

Given that $\mathbf{M}_t = \mathbf{S}Rot(\mathbf{m}_t\mathbf{q})\mathbf{S}^{-1} \in \mathbb{Z}^{n\times n}$ and $\mathbf{N}_t = \mathbf{T}^{-1}Rot(\mathbf{n}_t\mathbf{q})\mathbf{T} \in \mathbb{Z}^{n\times n}$, we can attempt to use the lattice reduction algorithm to determine the secret elements in our construction. By using $\mathbf{M}_t$ and $\mathbf{N}_t$, we can generate the following lattices:

$$L_1(t_1,t_2) = \begin{pmatrix} \mathbf{M}_{t_1} \\ \mathbf{M}_{t_2} \end{pmatrix}, \quad L_2(t_1,t_2) = \begin{pmatrix} \mathbf{N}_{t_1} \\ \mathbf{N}_{t_2} \end{pmatrix}.$$

By applying the lattice reduction algorithm, we obtain $\mathbf{E}_1 Rot(\mathbf{q})\mathbf{S}^{-1}$ and $\mathbf{E}_2 Rot(\mathbf{q})\mathbf{T}$. However, according to the current lattice reduction algorithm, $\mathbf{E}_1$ and $\mathbf{E}_2$ are not identity matrices. If $\mathbf{E}_1 = \mathbf{E}_2 = \mathbf{I}$, we can further derive $Rot(\mathbf{q})$ by using $Rot(\mathbf{q})\mathbf{T}$ and $Rot(\mathbf{q})\mathbf{S}^{-1}$ and finally solve $\mathbf{T}$ and $\mathbf{S}$. When $\mathbf{T}$ and $\mathbf{S}$ are known, our construction can be broken, as will be described in the following subsection. Thus, we must set a sufficiently large dimension for our construction to prevent lattice reduction attack.

5.4.3 Attack of Known T and S

When $\mathbf{T}$ and $\mathbf{S}$ are known, we can compute $\mathbf{N}_t' = Rot(\mathbf{n}_t\mathbf{q}) = \mathbf{T}\mathbf{N}_t\mathbf{T}^{-1}$ and $\mathbf{M}_t' = Rot(\mathbf{m}_t\mathbf{q}) = \mathbf{S}^{-1}\mathbf{M}_t\mathbf{S}$. By using $\mathbf{N}_t'$ and $\mathbf{M}_t'$, we can solve the basis $\mathbf{Q} = \mathbf{E}_1 Rot(\mathbf{q})$ of $\mathbf{q}$, where $\mathbf{E}_1$ is a unimodular matrix.

On the basis of $\mathbf{X}_i = \mathbf{S}Rot(\mathbf{x}_i)\mathbf{S}^{-1}$ and $\mathbf{Y}_i = \mathbf{T}^{-1}Rot(\mathbf{y}_i)\mathbf{T}$, we derive the following expressions:

$$Rot(\mathbf{x}_i) = \mathbf{S}^{-1}\mathbf{X}_i\mathbf{S},$$
$$Rot(\mathbf{y}_i) = \mathbf{T}\mathbf{Y}_i\mathbf{T}^{-1}.$$

Notably, we obtain $\mathbf{x}_i = (\mathbf{a}_i + \mathbf{e}_i\mathbf{f}) \bmod \mathbf{q}$, $\mathbf{y}_i = (\mathbf{a}_i\mathbf{g} + \mathbf{t}_i\mathbf{f}) \bmod \mathbf{q}$ by using the cross-multiplication of $\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{z}_{i_1,i_2} = \mathbf{x}_{i_1}\mathbf{y}_{i_2} - \mathbf{x}_{i_2}\mathbf{y}_{i_1} = \mathbf{r}_{i_1,i_2}\mathbf{f}$. By using $\mathbf{z}_{i_1,i_2}$, we can compute the basis $\mathbf{B} = \mathbf{E} \cdot Rot(\mathbf{f})$ of $\mathbf{f}$, where $\mathbf{E}$ is a unimodular matrix.

In the subsequent sections, we remove matrices $\mathbf{T}$ and $\mathbf{S}$ in the public parameters for simplicity.

Without loss of generality, we assume that $Rot(\mathbf{x}_1)$ and $\mathbf{B}$ are co-primes, that is, $\gcd(\det(Rot(\mathbf{x}_1)), \det(\mathbf{B})) = 1$. Otherwise, we select other pair of encodings $\mathbf{x}_i$ and $\mathbf{y}_i$.

By using the LLL algorithm, we can compute $\mathbf{U}_1, \mathbf{U}_2$, such that $\mathbf{U}_1 \cdot Rot(\mathbf{x}_1) + \mathbf{U}_2 \cdot \mathbf{B} = \mathbf{U}_3$ and $\mathbf{U}_3$ is a unimodular matrix. As such, $\mathbf{U}_3^{-1} \cdot \mathbf{U}_1 \cdot Rot(\mathbf{x}_1) + \mathbf{U}_3^{-1} \cdot \mathbf{U}_2 \cdot \mathbf{B} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix.

On the basis of $\mathbf{x}_1 = (\mathbf{a}_1 + \mathbf{s}_1 \cdot \mathbf{f}) \bmod \mathbf{q} = \mathbf{a}_1 + \mathbf{s} \cdot \mathbf{f}$, $\mathbf{y}_1 = (\mathbf{a}_1 \cdot \mathbf{g} + \mathbf{t}_1 \cdot \mathbf{f}) \bmod \mathbf{q} = \mathbf{a}_1 \cdot \mathbf{g} + \mathbf{t} \cdot \mathbf{f}$, we derive the following expressions:

$$\begin{aligned}
&\mathbf{U}_3^{-1} \cdot \mathbf{U}_1 \cdot Rot(\mathbf{x}_1) + \mathbf{U}_3^{-1} \cdot \mathbf{U}_2 \cdot \mathbf{B} \\
&= \mathbf{U}_3^{-1}\mathbf{U}_1 \cdot Rot(\mathbf{a}_1) + (\mathbf{U}_3^{-1}\mathbf{U}_1 Rot(\mathbf{s}) + \mathbf{U}_3^{-1}\mathbf{U}_2\mathbf{E}) \cdot Rot(\mathbf{f}), \\
&= \mathbf{U}_4 \cdot Rot(\mathbf{a}_1) + \mathbf{U}_5 \cdot Rot(\mathbf{f}) \\
&= \mathbf{I}
\end{aligned}$$

$$\begin{aligned}
\mathbf{B}_1 &= \mathbf{U}_4 \cdot Rot(\mathbf{y}_1) \\
&= \mathbf{U}_4 \cdot Rot(\mathbf{a}_1 \cdot \mathbf{g} + \mathbf{t} \cdot \mathbf{f}) \\
&= \mathbf{U}_4 \cdot Rot(\mathbf{a}_1) \cdot Rot(\mathbf{g}) + \mathbf{U}_4 Rot(\mathbf{t}) \cdot Rot(\mathbf{f}) \\
&= (\mathbf{I} - \mathbf{U}_5 \cdot Rot(\mathbf{f})) \cdot Rot(\mathbf{g}) + \mathbf{U}_4 Rot(\mathbf{t}) \cdot Rot(\mathbf{f}), \\
&= Rot(\mathbf{g}) - (\mathbf{U}_5 Rot(\mathbf{g}) + \mathbf{U}_4 Rot(\mathbf{t})) \cdot Rot(\mathbf{f}) \\
&= Rot(\mathbf{g}) - \mathbf{U}_6 \cdot Rot(\mathbf{f})
\end{aligned}$$

where $\mathbf{U}_4 = \mathbf{U}_3^{-1}\mathbf{U}_1$, $\mathbf{U}_5 = \mathbf{U}_3^{-1}\mathbf{U}_1 Rot(\mathbf{s}) + \mathbf{U}_3^{-1}\mathbf{U}_2\mathbf{E}$, and $\mathbf{U}_6 = \mathbf{U}_5 Rot(\mathbf{g}) + \mathbf{U}_4 Rot(\mathbf{t})$.

By using the LLL algorithm, we can compute $\mathbf{U}_7, \mathbf{U}_8$, such that $\mathbf{U}_7 \cdot \mathbf{B}_1 + \mathbf{U}_8 \cdot \mathbf{B} = \mathbf{U}_9$ and $\mathbf{U}_9$ is a unimodular matrix. As such, $\mathbf{U}_9^{-1}\mathbf{U}_7 \cdot \mathbf{B}_1 + \mathbf{U}_9^{-1}\mathbf{U}_8 \cdot \mathbf{B} = \mathbf{I}$.

That is,

$$\begin{aligned}
&\mathbf{U}_9^{-1}\mathbf{U}_7 \cdot \mathbf{B}_1 + \mathbf{U}_9^{-1}\mathbf{U}_8 \cdot \mathbf{B} \\
&= \mathbf{U}_9^{-1}\mathbf{U}_7 \cdot (Rot(\mathbf{g}) - \mathbf{U}_6 \cdot Rot(\mathbf{f})) + \mathbf{U}_9^{-1}\mathbf{U}_8 \cdot \mathbf{E} \cdot Rot(\mathbf{f}) \\
&= \mathbf{U}_9^{-1}\mathbf{U}_7 \cdot Rot(\mathbf{g}) - (\mathbf{U}_9^{-1}\mathbf{U}_7 \cdot \mathbf{U}_6 + \mathbf{U}_9^{-1}\mathbf{U}_8 \cdot \mathbf{E}) \cdot Rot(\mathbf{f}), \\
&= \mathbf{U}_{10} \cdot Rot(\mathbf{g}) - \mathbf{U}_{11} Rot(\mathbf{f}) \\
&= \mathbf{I}
\end{aligned}$$

where $\mathbf{U}_{10} = \mathbf{U}_9^{-1}\mathbf{U}_7$ and $\mathbf{U}_{11} = \mathbf{U}_9^{-1}\mathbf{U}_7 \cdot \mathbf{U}_6 + \mathbf{U}_9^{-1}\mathbf{U}_8 \cdot \mathbf{E}$.

Thereafter, given a level-1 encoding $\mathbf{u} = \sum_{i=1}^{\tau} w_i \cdot \mathbf{y}_i \bmod \mathbf{q}$, we derive the following expression:

$$
\begin{aligned}
\mathbf{X}' &= \mathbf{U}_{10} Rot(\mathbf{u}) \\
&= \mathbf{U}_{10} \cdot Rot(\textstyle\sum_{i=1}^{\tau} w_i \cdot \mathbf{y}_i) \\
&= \mathbf{U}_{10} \cdot Rot(\textstyle\sum_{i=1}^{\tau} w_i \cdot (\mathbf{a}_i \cdot \mathbf{g} + \mathbf{t}_i \cdot \mathbf{f})) \\
&= \mathbf{U}_{10} \cdot \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i \cdot \mathbf{g}) + \mathbf{U}_{10} \cdot \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{t}_i \cdot \mathbf{f}) \\
&= \mathbf{U}_{10} \cdot Rot(\mathbf{g}) \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + \mathbf{U}_{10} \cdot \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{t}_i) \cdot Rot(\mathbf{f}) \\
&= (\mathbf{I} + \mathbf{U}_{11} Rot(\mathbf{f})) \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + \mathbf{U}_{10} \cdot \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{t}_i) \cdot Rot(\mathbf{f}) \\
&= \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + (\mathbf{U}_{11} \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + \mathbf{U}_{10} \cdot \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{t}_i)) \cdot Rot(\mathbf{f}) \\
&= \textstyle\sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + \mathbf{U}_{12} \cdot Rot(\mathbf{f})
\end{aligned},
$$

where $\mathbf{U}_{12} = \mathbf{U}_{11} \sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{a}_i) + \mathbf{U}_{10} \cdot \sum_{i=1}^{\tau} Rot(w_i \cdot \mathbf{t}_i)$.

Thus, we can obtain $\mathbf{a} = (\sum_{i=1}^{\tau} w_i \cdot \mathbf{a}_i + \mathbf{r} \cdot \mathbf{f}) \bmod (\mathbf{n}_1 \mathbf{q})$ by arranging the first row of $\mathbf{X}'$ and considering modulus $(\mathbf{n}_1 \mathbf{q})$. Notably, we can solve a level-0 encoding $\mathbf{a}$ corresponding to $\mathbf{u}$.

5.4.4 Attack of the Principal Ideal Generator

Given that $\mathbf{X}_i = \mathbf{S} Rot(\mathbf{x}_i) \mathbf{S}^{-1} \in \mathbb{Z}^{n \times n}$ and $\mathbf{Y}_i = \mathbf{T}^{-1} Rot(\mathbf{y}_i) \mathbf{T} \in \mathbb{Z}^{n \times n}$, we can compute $p_i = \det(Rot(\mathbf{x}_i))$ and $q_i = \det(Rot(\mathbf{y}_i))$.

Without loss of generality, we assume that p_1, q_1 are primes. We first factor $(x^n + 1) \bmod p_1$ and derive the n roots of α_j to obtain $\mathbf{x}_1$. We then solve the principal ideal generator of $(p_1, \alpha_j), j \in [\![n]\!]$ by using an arbitrary efficient algorithm (if existing). We assume that the principal ideal generator generated by (p_1, α_1) is $\mathbf{x}_1$. Finally, we derive $\mathbf{S}$ by using $\mathbf{x}_1$. Similarly, we can also obtain $\mathbf{T}$.

If there exists an efficient algorithm that solves the short principal ideal generator, can we prevent this attack? In fact, previous algorithms computing the principal ideal generator do not guarantee that we can derive the original $\mathbf{x}_1$ in $\mathbf{X}_i = \mathbf{S} Rot(\mathbf{x}_i) \mathbf{S}^{-1}$ [CDPR15]. If we can derive the short principal ideal generator $\mathbf{x}$ of $\mathbf{x}_1$ but $\mathbf{x} \neq \mathbf{x}_1$, we can obtain a nontrivial unit element $\boldsymbol{\delta} = \mathbf{x}_1 / \mathbf{x}$, that is, $\det(Rot(\boldsymbol{\delta})) = 1$. Once we obtain the short nontrivial unit element $\boldsymbol{\delta}$, we can modify our construction, as follows:

$$
\mathbf{Y}_i = \mathbf{T}^{-1} Rot(\boldsymbol{\delta} \mathbf{y}_i) \mathbf{T} \quad \text{and} \quad \mathbf{X}_i = \mathbf{S} Rot(\boldsymbol{\delta} \mathbf{x}_i) \mathbf{S}^{-1},
$$

$$
\mathbf{M}_t = \mathbf{S} Rot(\boldsymbol{\delta} \mathbf{m}_t \mathbf{q}) \mathbf{S}^{-1} \quad \text{and} \quad \mathbf{N}_t = \mathbf{T}^{-1} Rot(\boldsymbol{\delta} \mathbf{n}_t \mathbf{q}) \mathbf{T}.
$$

In this case, we must exactly solve $\boldsymbol{\delta} \mathbf{y}_i$ and $\boldsymbol{\delta} \mathbf{x}_i$ to obtain $\mathbf{T}$ and $\mathbf{S}$. Otherwise, we cannot derive $\mathbf{T}$ and $\mathbf{S}$. So, this countermeasure improves the security of our construction.

6 Asymmetric ideal multilinear maps

Asymmetric ideal multilinear maps with different group are required in some applications. Similar to [GGH13], we briefly describe asymmetric variant as follow.

In the asymmetric construction, we sample different $\mathbf{g}_{j,t} \leftarrow D_{\mathbb{Z}^n, \sigma}, t = 1, ..., \beta$. The

element of the form $(\mathbf{a}_{j,t}\cdot\mathbf{g}_{j,t})\bmod\mathbf{f}_j$ is a level-1 encoding relative to the t-th generator $\mathbf{g}_{j,t}$. We denote by vectors the different levels of encoding. For a level-0 encoding $\mathbf{a}$, the encoding $\mathbf{c}$ with an index vector $\mathbf{w}=(w_1,...,w_\beta)\in\mathbb{N}^\beta$ is satisfied to $\mathbf{c}\bmod\mathbf{f}_j=(\mathbf{a}\prod_{t=1}^{\beta}(\mathbf{g}_{j,t})^{w_t})\bmod\mathbf{f}_j$. So, we give the public parameters $\mathbf{x}_{i,t}=(\mathbf{a}_{i,t}+\mathbf{s}_{i,t}\cdot\mathbf{f})\bmod\mathbf{q}$ and $\mathbf{y}_{i,t}=(\mathbf{a}_{i,t}\mathbf{g}_t+\mathbf{t}_{i,t}\cdot\mathbf{f})\bmod\mathbf{q}$, $i\in[\![\tau]\!]$, $t\in[\![\beta]\!]$, where $\mathbf{a}_{i,t}=\mathbf{a}_{i,j,t}\bmod\mathbf{f}_j$, $\mathbf{g}_{j,t}=\mathbf{g}_t\bmod\mathbf{f}_j$. Finally, we choose unimodular matrices $\mathbf{T},\mathbf{S}$ and generate the public parameters $\mathrm{par}_4=\left\{q,\mathbf{Q}_x,\mathbf{Q}_y,\left\{\mathbf{X}_{i,t},\mathbf{Y}_{i,t}\right\}_{i\in[\![\tau]\!],t\in[\![\beta]\!]},\mathbf{P}_{zt}\right\}$. For the variant in Section 4, we can similarly obtain its asymmetric variant.

7 Commutative variant

In our ideal construction using unimodular matrix, the dimension n must be large enough to guarantee security and $\tau>n^2+\lambda$ is the lowest requirement to avoid algebraic equation attack. As a result, the public parameter size of our construction is too large to be practical. To decrease the public parameter size, we use polynomial ring instead of the ring of integers.

We use $R_1=\mathbb{Z}[y]/<y^{n_1}+1>$ and $R_{1,q}=\mathbb{Z}_q[y]/<y^{n_1}+1>$ instead of $\mathbb{Z}$ and $\mathbb{Z}_q$ for our ideal construction using unimodular matrices. It is easy to verify that our ideal construction is still correct under this case.

8 Applications

Using our construction of ideal multilinear maps, we describe two applications: the one-round multipartite Diffie–Hellman key exchange protocol (MPKE) and witness encryption (WE).

8.1 MPKE

In this subsection, we present basic construction using our basic ideal scheme and construction using unimodular matrix.

8.1.1 Basic construction

We describe the construction of one-round multipartite Diffie-Hellman key exchange protocol using our ideal multilinear maps in Section 3 and 4. As in [GGH13], protocol security relies on the hardness assumption of the ext-GDDH.

Setup(1^λ). For $\forall t\in[\![2]\!]$, output $(\mathrm{par}_t)\leftarrow\mathrm{InstGen}_t(1^\lambda)$ as the public parameter.

Publish(par_t,j). Let N be the number of participants, $j\in[\![N]\!]$. Each party j samples random elements $\mathbf{w}_{j,i}\leftarrow D_{\mathbb{Z}^n,\sigma'},i\in[\![\tau]\!]$ which is used as secret key, computes and publishes level-1 encoding $\mathbf{u}_j\leftarrow\mathrm{Enc}_t\left(\mathrm{par}_t,1,\left\{\mathbf{w}_{j,i}\right\}_{i\in[\![\tau]\!]}\right)$ as a public key.

KeyGen$(\mathrm{par}_t,j,\left\{\mathbf{w}_{j,i}\right\}_{i\in[\![\tau]\!]},\left\{\mathbf{u}_j\right\}_{j\neq i})$. Each party j computes $\mathbf{c}_j=\prod_{k\neq j}\mathbf{u}_k$ and extracts the common secret key $sk=\mathrm{Ext}_t(\mathrm{par}_t,\mathbf{c}_j,\left\{\mathbf{w}_{j,i}\right\}_{i\in[\![\tau]\!]})$.

Theorem 8.1 Suppose that ext-GDDH is hard, then our construction above is one-round

multipartite Diffie-Hellman key exchange protocol.
Proof. The proof is similar as Theorem 2 in [GGH13]. ■

8.1.2 Construction using unimodular matrix

$\boldsymbol{Setup}(1^\lambda)$. The output $(\text{par}_3) \leftarrow \text{InstGen}_3(1^\lambda)$ is used as the public parameter.

$\boldsymbol{Publish}(\text{par}_3, j)$. We let N be the number of participants. For $j \in [\![N]\!]$, each party j samples random elements $w_{j,i} \leftarrow D_{\mathbb{Z},\sigma'}, i \in [\![\tau]\!]$, which are used as secret keys. Thereafter, level-1 encoding $\mathbf{U}_j \leftarrow \text{Enc}_3\left(\text{par}_3, 1, \{w_{j,i}\}_{i\in[\![\tau]\!]}\right)$ is computed and published as a public key.

$\boldsymbol{KeyGen}(\text{par}_3, j, \{w_{j,i}\}_{i\in[\![\tau]\!]}, \{\mathbf{U}_j\}_{j\neq i})$. Each party j computes $\mathbf{C}_j = \prod_{k\neq j} \mathbf{U}_k$ and extracts the common secret key $sk = \text{Ext}_3(\text{par}_3, \mathbf{C}_j, \{w_{j,i}\}_{i\in[\![\tau]\!]})$.

Remark 8.2 Given that each party merely requires the level-1 encoding be generated in the MPKE protocol, the parameters $\mathbf{s}^*$, $\mathbf{X}_i$, $\mathbf{P}_{zt}$ in par can be combined into a vector $\mathbf{p}_{zt,i} = \left[\mathbf{s}^* \cdot \mathbf{X}_i \cdot \mathbf{P}_{zt}\right]_q$. As such, the public parameters can be $\text{par}_3' = \left\{q, \mathbf{N}, \{\mathbf{p}_{zt,i}, \mathbf{Y}_i\}_{i\in[\![\tau]\!]}, \mathbf{t}^*\right\}$.

Theorem 8.3 Suppose that the ideal-ext-GDDH is hard. Then our protocol is a one-round multipartite Diffie–Hellman key exchange protocol.
Proof. The proof is similar to the proof presented in Theorem 2 in [GGH13]. ■

8.2 Witness Encryption

For $\alpha = 3\theta$, an instance $inst$ of 3-exact cover problem consists of a number α and a collection Set of subsets $S_1, S_2, ..., S_\beta \subset [\![\alpha]\!]$, find a 3-exact cover of $[\![\alpha]\!]$. For an instance of witness encryption, the public key is a collection Set of $inst$ and the public parameters par_3 in our ideal construction, the secret key is a hidden 3-exact cover of $[\![\alpha]\!]$ for $inst$.

$\boldsymbol{Encrypt}(1^\lambda, inst, \text{par}_3, M)$**:**

(1) Generate α level-1 encodings $\mathbf{U}_k = \sum_{i=1}^{\tau} d_{k,i}\mathbf{Y}_i \bmod \mathbf{N}$, $k \in [\![\alpha]\!]$, where $\mathbf{d}_k \leftarrow D_{Z^\tau,\sigma}$.

(2) Generate an encryption key $sk = \text{Ext}_3(\text{par}_3, \mathbf{I}, \mathbf{U})$, where $\mathbf{U} = (\prod_{k=1}^{\alpha} \mathbf{U}_k) \bmod \mathbf{N}$, and encrypt M into ciphertext C using encryption algorithm (such as AES), where $\mathbf{I}$ is an identity matrix.

(3) For each element $S_i = \{i_1, i_2, i_3\}$, generate a level-3 encoding $\mathbf{U}_{S_i} = (\mathbf{U}_{i_1}\mathbf{U}_{i_2}\mathbf{U}_{i_3}) \bmod \mathbf{N}$.

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

$\boldsymbol{Decrypt}(inst, W, C, E)$:

(1) Given C, E and a witness set W, compute $\mathbf{U} = (\prod_{S_i\in W} \mathbf{U}_{S_i}) \bmod \mathbf{N}$.

(2) Generate $sk = \mathrm{Ext}_3(\mathrm{par}_3, \mathbf{I}, \mathbf{U})$, and decrypt C to get the plaintext M.

Correctness. the correctness of witness encryption directly follows that of our ideal construction.

Security. Similar to [GGSW13], the security of our construction depends on the assumption of the Decision Graded Encoding No-Exact-Cover.

Theorem 8.4 Suppose that the Decision Graded Encoding No-Exact-Cover is hard. Then our construction is a witness encryption scheme.

Proof. The proof is identical to one presented in Theorem 5.2 in [GGSW13]. ■

Possible Attacks. (1) For $\mathbf{U}_{S_i} = (\mathbf{U}_{i_1}\mathbf{U}_{i_2}\mathbf{U}_{i_3}) \bmod \mathbf{N}$, one cannot get $(\mathbf{U}_{S_i})^{-1} \bmod \mathbf{N}$ since $\mathbf{N}$ consists of encodings of zero. On the one hand, $\mathbf{U}_{S_i}$ is a vector, whereas $\mathbf{N}$ is a matrix. On the other hand, given a matrix $\mathbf{N}_t$ that is a column vector of $\mathbf{N}$, one can generate two matrices $\mathbf{E}_1, \mathbf{E}_2$ such that $\mathbf{E}_1\mathbf{U}_{S_i} + \mathbf{E}_2\mathbf{N}_t = \mathbf{I}$. However, one cannot find the matrices $\mathbf{E}_1, \mathbf{E}_2$ of the form $\mathbf{T}^{-1}Rot(\mathbf{r})\mathbf{T}$ for large dimension n, where $\mathbf{r} \in R$. So, the attack in [HJ15b] does not work for our construction.

(2) For the attack in [HJ15a], we adapt their notation to our construction to obtain the following equation

$$(\prod\nolimits_{pf \in CPF} \mathbf{U}_{(pf)}) \bmod \mathbf{N} = (\prod\nolimits_{k=1}^{\alpha} \mathbf{U}_k) \bmod \mathbf{N} \times (\prod\nolimits_{nf \in CPF} \mathbf{U}_{(nf)}) \bmod \mathbf{N},$$

where CPF is the collection of positive factors, and CNF the collection of negative factors (see [HJ15a]).

Given $PPF = (\prod_{pf \in CPF} \mathbf{U}_{(pf)}) \bmod \mathbf{N}$ and $PNF = (\prod_{nf \in CPF} \mathbf{U}_{(nf)}) \bmod \mathbf{N}$, find $PTS = (\prod_{k=1}^{\alpha} \mathbf{U}_k) \bmod \mathbf{N}$. Similarly, one cannot solve the inverse of PNF modulo $\mathbf{N}$. Thus, the attack in [HJ15a] does not hold for our construction.

Remark 8.5 We select an element $\varphi \in [\![\alpha]\!]$ such that $\left|S^{(\varphi)}\right| = \max\left\{\left|S^{(\pi)}\right|, \pi \in [\![\alpha]\!]\right\}$, where $S^{(\pi)} = \{S_i \mid (\pi \in S_i) \bigcap (S_i \in Set)\}$. For $S_i \in S^{(\varphi)}$, we modify $\mathbf{U}_{S_i} = (\mathbf{U}_{i_1}\mathbf{U}_{i_2}\mathbf{U}_{i_3}) \bmod \mathbf{N}$ into $\mathbf{u}_{S_i} = \left[\mathbf{s}^* \cdot \mathbf{I} \cdot \mathbf{P}_{zt} \cdot \mathbf{U}_{S_i}\right]_q$. In this case, we do not increase the size of q since the level-0 encoding $\mathbf{I}$ (identity matrix) compensates an increase multiplied by $\mathbf{U}_{S_i}$. When decrypting, we compute the secret key as follows:

$$sk = \mathrm{Extract}_s(\mathrm{msbs}_\gamma(\left[\mathbf{u}_{S_i \in W \cap S_i \in S^{(\varphi)}} \cdot (\prod\nolimits_{S_i \in W \cap S_i \notin S^{(\varphi)}} \mathbf{U}_{S_i} \bmod \mathbf{N}) \cdot \mathbf{t}^*\right]_q)).$$

Using this countermeasure, we merely increase the difficulty that adversary attacks our witness encryption using a combined 3-exact cover.

References

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

[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, Part I. LNCS 8616, pp. 480–499.

[CDPR15] R. Cramer, L. Ducas, C. Peikertz, and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings. http://eprint.iacr.org/2015/313.

[CHL+14] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehle. Cryptanalysis of the Multilinear Map over the Integers. EUROCRYPT 2015, Part I, LNCS 9056, pp. 3–12.

[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.

[CLT15] J. S. Coron, T. Lepoint, and M. Tibouchi. New Multilinear Maps over the Integers. http://eprint.iacr.org/2015/162.

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

[GGS+13] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. STOC 2013, pp. 467–476.

[GGH+13a] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. Cryptology ePrint Archive, Report 2013/128 (2013)

[GGH+13b] 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.

[GGHZ14] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. http://eprint.iacr.org/2014/666.

[GGSW13] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. STOC 2013, pp. 467–476.

[GLSW14] C. Gentry, A. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. http://eprint.iacr.org/2014/309.

[GLW14] C. Gentry, A. Lewko, and B. Waters. Witness Encryption from Instance Independent Assumptions. CRYPTO 2014. LNCS 8616, pp. 426 - 443.

[Gu15] Gu Chunsheng. Multilinear Maps Using Ideal Lattices without Encodings of Zero. http://eprint.iacr.org/2015/023.

[HJ15a] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH Map. http://eprint.iacr.org/2015/301.

[HJ15b] Yupu Hu and Huiwen Jia. A Comment on Gu Map-1. http://eprint.iacr.org/2015/448.

[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.

[LPSS14] S. Ling, D. Phan, D. Stehlé, and R. Steinfeld. Hardness of k-lwe and applications in traitor tracing. CRYPTO 2014, LNCS 8616, pp. 315-334.

[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.

[Sma03] N.P Smart. 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.

[Zim15] J. Zimmerman. How to obfuscate programs directly. EUROCRYPT 2015, Part II, LNCS 9057, pp. 439–467.