Multilinear Maps Using a Variant of Ring-LWE

Gu Chunsheng

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

Abstract. GGH13, CLT13 and GGH15 of multilinear maps suffer from zeroizing attacks. In this paper, we present a new construction of multilinear maps using a variant of ring-LWE (vRLWE). Furthermore, we also present two new variants of vRLWE, which respectively support the applications of multipartite key exchange and witness encryption. The security of our construction depends upon new hardness assumptions.

Keywords: Multilinear maps, ring-LWE, multipartite key exchange, zeroizing attack, approximate GCD

1 Introduction

Multilinear maps have plenty of applications including multipartite key exchange (MPKE), program obfuscation and efficient broadcast encryption [9,20,21,37]. However, current constructions GGH13, CLT13 and GGH15 [19,15,23] do not depend upon classic hardness assumptions, and recent are proved to be insecure [19,12,11,30,18], especially for MPKE using these constructions.

The GGH13 construction has the weak discrete logarithm attack (or called zeroizing attack) presented by authors themselves [19]. By using the weak-DL attack, one can get related information of some secret parameters of GGH13 such as basis of secret element. As a result, some problems including the subgroup membership problem and the decisional linear problem are become easy. Very recently, Hu and Jia [30] presented an efficient weak-DL-based attack on the GGH13 map, which breaks the GGH13-based applications on multipartite key exchange and witness encryption (WE) based on the hardness of 3-exact cover problem. To fix GGH13, Gentry, Halevi and Lepoint [28] recently described a variant of the GGH13 scheme [19], in which the linear zero-testing procedure from [19] is replaced by a quadratic (or higher-degree) procedure. However, Brakerski et al. [5] showed that this variant of GGH13 fails to thwart zeroizing attacks. On the other hand, Halevi [29] described a variant of GGH13. But, Coron et al [18] proved that this variant is also insecure.

The CLT13 construction also has the problem of zeroizing attack. Cheon et al. [12] broke CLT13 using an extension of zeroizing attack. To fix CLT13, Garg, Gentry, Halevi and Zhandry [22], and Boneh, Wu and Zimmerman [8] respectively described two variants of multilinear maps over the integers. However, Coron et al. [11] extended Cheon et al.'s attack [12] to setting where no encoding of zero below top level are available. Consequently, these variants [22,8] can also

be defeated using an extension of Cheon et al.'s zeroizing attack [11]. Recently, Coron, Lepoint and Tibouchi [17] (CLT15) presented a new improvement of CLT13 by modifying zero-testing parameter. However, the CLT15 construction is also insecure [14,32].

Recently, the GGH15-based MPKE is attacked by using a variant of the Cheon et al.'s attack [18]. While the public parameters in GGH15 does not include encodings of zero, each plaintext appears in the encoding of each path of the directed graph of GGH15. Coron et al. [18] described an attack of GGH15, which broke GGH15-based MPKE in polynomial time by generating an equivalent user private key.

Currently, it is still an open problem how to construct a secure multilinear maps, in particular supporting MPKE.

1.1 Our contribution

Our main contribution is to present a new multilinear map using a variant of ring-LWE. The security of our construction is still dependent on the new hardness assumption. However, we observe that if a construction of multilinear maps supports MPKE, it is impossible to completely avoid zeroizing attacks.

Our starting point is a new variant of LWE [35]. In the LWE problem, given q a prime integer, and a list of samples $(\mathbf{a}_l, b_l = [< \mathbf{a}_l, \mathbf{s} > + e_l]_q)$, where $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{a}_l \in \mathbb{Z}_q^n$ are chosen independently and uniformly from $\mathbb{Z}_q^n$, and e_l is chosen independently according to the probability distribution $\chi = D_{\mathbb{Z},\sigma}$, find $\mathbf{s}$. In the first variant of LWE, $\mathbf{s}$ is chosen from the error distribution χ^n rather than uniformly at random, the choice of other parameters remains unchanged. This variant becomes no easier to solve than the decisional LWE [34,2].

In this paper, we introduce a new variant of LWE. Namely, we draw many samples $(\mathbf{a}_l, b_l = [< \mathbf{a}_l, \mathbf{s} > + e_l]_q)$, where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{a}_l \leftarrow D_{\mathbb{Z}^n,\sigma}$, $e_l \leftarrow D_{\mathbb{Z},\sigma}$. To directly support multiplication, we write samples in the matrix form. That is, given many samples $(\mathbf{A}_l, \mathbf{B}_l = [\mathbf{A}_l\mathbf{S} + \mathbf{E}_l]_q)$, where $\mathbf{S} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{A}_l \leftarrow D_{\mathbb{Z}^{n\times n},\sigma}$, $\mathbf{E}_l \leftarrow D_{\mathbb{Z}^{n\times m},\sigma}$, the problem is to find $\mathbf{S}$. Since this new variant is also a special form of LWE, its decisional version is equivalent to the search version. However, at present this variant can not be reduced to LWE or other classical hardness problems.

It is easy to see that this variant supports addition and multiplication. For example, given samples $(\mathbf{A}_1, \mathbf{B}_1), (\mathbf{A}_2, \mathbf{B}_2)$, for addition we have:

$$\begin{cases} \mathbf{A} & = \mathbf{A}_1 + \mathbf{A}_2, \\ \mathbf{B} & = \mathbf{B}_1 + \mathbf{B}_2 = (\mathbf{A}_1 + \mathbf{A}_2)\mathbf{S} + (\mathbf{E}_1 + \mathbf{E}_2) = \mathbf{AS} + \mathbf{E} \mod q. \end{cases}$$

Similarly, for multiplication we have:

$$\begin{cases} \mathbf{A} & = \mathbf{A}_1\mathbf{A}_2, \\ \mathbf{B} & = \mathbf{A}_1\mathbf{B}_2 = \mathbf{A}_1\mathbf{A}_2\mathbf{S} + \mathbf{A}_1\mathbf{E}_2 = \mathbf{AS} + \mathbf{E} \mod q, \end{cases}$$

where $\mathbf{E} = \mathbf{A}_1\mathbf{E}_2$.

Since $\mathbf{A}_1, \mathbf{A}_2, \mathbf{E}_1, \mathbf{E}_2$ are all "small", as a consequence, $\mathbf{A}$ and $\mathbf{E}$ generated by addition or multiplication are also "small".

1.2 Applications

Similar to GGH15 [23], here we also describe two applications using our new constructions. In order to improve efficiency, we switch to a ring version of the variant LWE. That is, we sample scalars from a large polynomial ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, rather than the ring of integers $\mathbb{Z}$.

MPKE. Let κ be the number of parties. Since the matrix product generally does not satisfy the commutative law, we choose $\mathbf{S} = \mathbf{sI}$ to construct MPKE. Given many samples $(\mathbf{A}_l, \mathbf{B}_l = \mathbf{A}_l\mathbf{S} + \mathbf{E}_l)$ as the public parameters, each party i generates a random linear combination $(\mathbf{U}_i, \mathbf{V}_i)$ of these samples, publicly publishes $\mathbf{U}_i$, and remains $\mathbf{V}_i$ secret. Applying an ordering of all parties generated by $\mathbf{U}_i$, (e.g. $1, 2, \cdots, \kappa$), then the i-th party computes an extracting encoding

$$\mathbf{C}_i = \prod_{j=1}^{i-1} \mathbf{U}_j \times \mathbf{V}_i \times \prod_{j=i+1}^{\kappa} \mathbf{U}_j = \prod_{j=1}^{\kappa} \mathbf{U}_j \times \mathbf{S} + \mathbf{E}_i^{'}$$

Finally, the i-th party extracts the shared secret key from the most significant bits of each element of $\mathbf{C}_i$.

The main difference between our scheme and their scheme is that in our scheme, the product of the plaintexts $\mathbf{U}_j$ is not commutative, the matrix $\mathbf{S}$ is commutative, whereas in their scheme, the situation is just the opposite.

Branching-program obfuscation. Branching program (BP) obfuscation can be constructed by applying the variant of LWE. Given a length-κ matrix BP $\{\mathbf{A}_{j,b}, j \in [\kappa], b \in [2]\}$, we first use Kilian's randomization to generate a matrix encode BP as follow:

$$\begin{aligned}
\overline{\mathbf{A}}_{j,b} &= \{\mathbf{T}_{j-1}^{-1}\mathbf{A}_{j,b}\mathbf{T}_j, j \in [\kappa - 1], b \in [2]\},\\
\overline{\mathbf{B}}_{\kappa,b} &= \{\mathbf{T}_{\kappa-1}^{-1}(\mathbf{A}_{\kappa,b}\mathbf{S} + \mathbf{E}_{\kappa,b})\mathbf{T}_\kappa, b \in [2]\},\\
\overline{\mathbf{u}} &= \mathbf{u}^T\mathbf{T}_0, \overline{\mathbf{v}} = \mathbf{T}_\kappa^{-1}\mathbf{v}.
\end{aligned}$$

Now, we can compute an encoding of a product of matrices corresponding to an input $\mathbf{x}$. If $\prod_{j\in[\kappa]} \mathbf{A}_{j,x_{ind_j}} = \mathbf{I}$, then we get $\mathbf{u}^T\mathbf{S}\mathbf{v} + \mathbf{u}^T\mathbf{E}'\mathbf{v}$. Hence, given $\mathbf{u}^T\mathbf{S}\mathbf{v} + e'$ in the public parameters, we can compare them to obtain the result of the computation.

Organization. Section 2 recalls some background. Section 3 describes our new construction using a variant of the ring LWE. Section 4 and 5 describe an asymmetric commutative variant and a symmetric commutative variant, respectively. Section 6 presents some applications using our construction and its variants.

2 Preliminaries

2.1 Notations

Let $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ denote the ring of integers, the field of rational numbers, and the field of real numbers. Let n be a positive integer and power of 2. Notation $[n]$ denotes the set $\{1, 2, ..., n\}$. Let $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, and $\mathbb{K} = \mathbb{Q}[x]/\langle x^n + 1\rangle$. Vectors are denoted in bold lowercase (e.g. $\mathbf{a}$), and matrices in bold uppercase (e.g. $\mathbf{A}$). We denote by a_j the j-th entry of a vector $\mathbf{a}$, and $a_{i,j}$ the element of the i-th row and j-th column of $\mathbf{A}$. We denote by $\|\mathbf{a}\|_2$ (abbreviated as $\|\mathbf{a}\|$) the Euclidian norm of $\mathbf{a}$. For $\mathbf{A} \in R^{d\times d}$, we define $\|\mathbf{A}\| = \max\{\|a_{i,j}\|, i, j \in [d]\}$, where $\|a_{i,j}\|$ is the Euclidian norm corresponding to the coefficient vector of $a_{i,j}$.

Let $[a]_q$ denote the absolute minimum residual system, namely $[a]_q = a \bmod q \in (-q/2, q/2]$. Similarly, for $\mathbf{a} \in \mathbb{Z}^n$ (or $a \in R$), $[\mathbf{a}]_q$ denotes each entry (or each coefficient) $[a_j]_q \in (-q/2, q/2]$ of $\mathbf{a}$ (or a).

2.2 Lattices and Ideal Lattices

An n-dimensional 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. Given a basis $\mathbf{B}$ of L, we define $P(\mathbf{B}) = \{\mathbf{By}|\mathbf{y} \in \mathbb{R}^n \text{ and } y_i \in [-1/2, 1/2)\}$ as the parallelization corresponding to $\mathbf{B}$. We let $\det(\mathbf{B})$ be the determinant of $\mathbf{B}$.

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

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

Micciancio and Regev [33] introduced the smoothing parameter of lattices. For an n-dimensional lattice L, and positive real $\epsilon > 0$, we define its smoothing parameter $\eta_\epsilon(L)$ to be the smallest s such that $\rho_{1/s}(L^*\backslash\{0\}) \le \epsilon$, where L^* is the dual lattice of L.

Lemma 2.1 (Lemma 3.3 [33]). For any n-dimensional lattice L and positive real $\epsilon > 0$, $\eta_\epsilon(L) \le \sqrt{\ln(2n(1 + 1/\epsilon))/\pi} \cdot \lambda_n(L)$.

Lemma 2.2 (Lemma 4.4 [33]). For any n-dimensional lattice L, vector $\mathbf{c} \in \mathbb{R}^n$ and reals $0 < \epsilon < 1$, $s \ge \eta_\epsilon(L)$, we have

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

2.3 Multilinear Maps

In the following, we give the definition of multilinear maps.

Definition 2.3 (Multilinear Map [7]). For κ+1 cyclic groups $G_1, ..., G_\kappa, G_T$ of the same order q, a κ-multilinear map $e : G_1 \times ... \times G_\kappa \to G_T$ has the following properties:

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

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

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

Definition 2.4 (κ-Graded Encoding System [19]). A κ-graded encoding system over R is a set system of $S = \{S_j^{(a)} \in R : a \in R, j \in [\kappa]\}$ with the following properties:

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

(2) Binary operations '+' and '-' exist, such that every a_1, a_2 , every index $j \in [\kappa]$, and every $u_1 \in S_j^{(a_1)}$ and $u_2 \in S_j^{(a_2)}$ hold that $u_1 + u_2 \in S_j^{(a_1+a_2)}$ and $u_1 - u_2 \in S_j^{(a_1-a_2)}$, where $a_1 + a_2$ and $a_1 - a_2$ are the addition and subtraction operations in R respectively.

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

3 Our construction

In this section, we first describe our construction using a variant of ring LWE (vRLWE), then show its correctness, and finally give its hardness assumption.

3.1 Construction

Setting the parameters. Let λ be the security parameter, κ the multilinearity level (or the number of times to support multiplication). For simplicity, concrete parameters are set as $n = O(\lambda)$, $\sigma = O(n)$, $O(n^{6.4\kappa+11.2}) < q < O(n^{8\kappa+14})$, $d \geq 2$, $\tau = \lambda d^2$.

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

(1) Choose a prime $O(n^{6.4\kappa+11.2}) < q < O(n^{8\kappa+14})$.

(2) Choose a random matrix $\mathbf{S} \leftarrow R_q^{d\times d}$.

(3) For $l \in [\tau]$, sample $\mathbf{A}_l, \mathbf{E}_l \in R^{d\times d}$ such that $a_{l,i,j}, e_{l,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma}, i, j \in [d]$.

(4) Sample $\mathbf{u}, \mathbf{v} \in R^d$ such that $u_i, v_i \leftarrow D_{\mathbb{Z}^n,\sigma}, i \in [d]$.

(5) For $l \in [\tau]$, set $\mathbf{B}_l = [\mathbf{A}_l\mathbf{S} + \mathbf{E}_l]_q$.

(6) Output the public parameters $par = \big\{q, (\mathbf{A}_l, \mathbf{B}_l)_{l\in[\tau]}, \mathbf{u}, \mathbf{v}\big\}$.

Generating a random encoding: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Enc}(par)$.

Given $r_l \leftarrow D_{\mathbb{Z}^n,\sigma}$, $l \in [\tau]$, generate

$$\mathbf{U} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{A}_l\big]_q, \mathbf{V} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{B}_l\big]_q.$$

Adding encodings: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Add}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$.
Given two encodings $(\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2)$, compute

$$\mathbf{U} = [\mathbf{U}_1 + \mathbf{U}_2]_q, \mathbf{V} = [\mathbf{V}_1 + \mathbf{V}_2]_q.$$

Multiplying encodings: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Mul}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$.
Given two encodings $(\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2)$, compute

$$\mathbf{U} = [\mathbf{U}_1\mathbf{U}_2]_q, \mathbf{V} = [\mathbf{U}_1\mathbf{V}_2]_q.$$

Zero-testing: $\mathrm{isZero}(par, (\mathbf{U}, \mathbf{V}))$:
Given an encoding $(\mathbf{U}, \mathbf{V})$, we check whether $\|\mathbf{u}^T\mathbf{V}\mathbf{v}\|$ is short:

$$\mathrm{isZero}(par, (\mathbf{U}, \mathbf{V})) = \begin{cases} 1, & \text{if } \|\mathbf{u}^T\mathbf{V}\mathbf{v}\| < q^{7/8}; \\ 0, & \text{otherwise.} \end{cases}$$

Extract: $sk \leftarrow \mathrm{Ext}(par, (\mathbf{U}, \mathbf{V}))$.
Given an encoding $(\mathbf{U}, \mathbf{V})$, we extract the $\eta = (\mathtt{log}q)/8 - \lambda$ most-significant bits from each of the n coefficients of $\mathbf{u}^T\mathbf{V}\mathbf{v}$:

$$\mathrm{Ext}(par, (\mathbf{U}, \mathbf{V})) = \mathrm{msbs}_\eta(\mathbf{u}^T\mathbf{V}\mathbf{v}),$$

where msbs_η extracts the η most significant bits from each coefficient of $\mathbf{u}^T\mathbf{V}\mathbf{v}$.

Remark 3.1 (1) Similar to GGH13 [20], we can construct graded encoding scheme by introducing a random ring element $z \in R_q$. That is, we generate the following parameters:

$$\overline{\mathbf{A}}_l = [\mathbf{A}_l/z]_q,$$
$$\overline{\mathbf{B}}_l = [(\mathbf{A}_l\mathbf{S} + \mathbf{E}_l)z^\kappa]_q$$

(2) Using Kilian randomization, we can also construct an asymmetric version as follows: For $l \in [\tau], k \in [\kappa + 1]$,

$$\overline{\mathbf{A}}_{l,k} = [\mathbf{T}_{k-1}^{-1}\mathbf{A}_{l,k}\mathbf{T}_k]_q,$$
$$\overline{\mathbf{B}}_{l,k} = [\mathbf{T}_{k-1}^{-1}(\mathbf{A}_{l,k}\mathbf{S} + \mathbf{E}_{l,k})\mathbf{T}_k]_q,$$
$$\overline{\mathbf{u}}^T = \mathbf{u}^T\mathbf{T}_0, \overline{\mathbf{v}} = \mathbf{T}_\kappa\mathbf{v},$$

where $\mathbf{S}, \mathbf{T}_k \in R_q^{d\times d}$, and $\mathbf{A}_{l,k}, \mathbf{E}_{l,k} \in R^{d\times d}$ such that $e_{l,k,i,j}, a_{l,k,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma}$.

3.2 Correctness

Correctness of our construction follows from a fact, which says that the encodings returned by Enc, Add, Mul are all legal samples of vRLWE. For completeness, we give the brief proof of correctness in the following.

Lemma 3.2 The algorithm InstGen$(1^\lambda, 1^\kappa)$ runs in polynomial time.

Proof. Since each step in InstGen runs in polynomial time, the result is directly obtained. ∎

Lemma 3.3 The encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \text{Enc}(par)$ is a sample of vRLWE.

Proof. By $\mathbf{U} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{A}_l\big]_q$ with $r_l \leftarrow D_{\mathbb{Z}^n,\sigma}$, $l \in [\tau]$, we have

$$\mathbf{V} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{B}_l\big]_q = \mathbf{US} + \mathbf{E},$$

where $\mathbf{E} = \sum_{l=1}^{\tau} r_l \mathbf{E}_l$. ∎

Lemma 3.4 The encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \text{Add}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$ is a sample of vRLWE.

Proof. By $\mathbf{V}_i = \mathbf{U}_i\mathbf{S} + \mathbf{E}_i, i \in [2]$, we have

$$\mathbf{U} = [\mathbf{U}_1 + \mathbf{U}_2]_q, \mathbf{V} = [\mathbf{V}_1 + \mathbf{V}_2]_q = [\mathbf{US} + \mathbf{E}]_q,$$

where $\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2$. ∎

Lemma 3.5 The encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \text{Mul}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$ is a sample of vRLWE.

Proof. By $\mathbf{V}_i = \mathbf{U}_i\mathbf{S} + \mathbf{E}_i, i \in [2]$, we have

$$\mathbf{U} = [\mathbf{U}_1\mathbf{U}_2]_q, \mathbf{V} = [\mathbf{U}_1\mathbf{V}_2]_q = [\mathbf{US} + \mathbf{E}]_q,$$

where $\mathbf{E} = \mathbf{U}_1\mathbf{E}_2$. ∎

Lemma 3.6 The procedure isZero$(par, (\mathbf{U}, \mathbf{V}))$ can correctly determine whether $(\mathbf{U}, \mathbf{V})$ is an encoding of zero.

Proof. By Lemma 2.1, $\sigma = O(n) > \eta_\epsilon(\mathbb{Z}^n)$. Using Lemma 2.2, we have $\|\mathbf{A}_l\| = O(\sigma\sqrt{n}) = O(n^{1.5})$ with overwhelming probability.

Given an encoding $(\mathbf{U}_1, \mathbf{V}_1)$ returned by Enc, we have

$$\|\mathbf{U}_1\| = \|\sum_{l=1}^{\tau} r_l \mathbf{A}_l\| = O(\tau \cdot n \cdot \|r_l\| \cdot \|\mathbf{A}_l\|) = O(n^4),$$

where $\|r_l\| = O(\sigma\sqrt{n}) = O(n^{1.5})$ for $r_l \leftarrow D_{\mathbb{Z}^n,\sigma}$.

Since $\mathbf{V}_1 = \sum_{l=1}^{\tau} r_l \mathbf{B}_l = \mathbf{U}_1\mathbf{S} + \sum_{l=1}^{\tau} r_l \mathbf{E}_l = \mathbf{U}_1\mathbf{S} + \widetilde{\mathbf{E}}_1 \mod q$, we have

$$\|\widetilde{\mathbf{E}}_1\| = \|\sum_{l=1}^{\tau} r_l \mathbf{E}_l\| = O(\tau \cdot n \cdot O(n^{1.5}) \cdot O(n^{1.5})) = O(n^4).$$

Since the above construction supports κ multiplications, without loss of generality, we assume that $(\mathbf{U}_i, \mathbf{V}_i = \mathbf{U}_i\mathbf{S} + \widetilde{\mathbf{E}}_i), i \in [\kappa+1]$ is the encodings returned by Enc, and $(\mathbf{U}, \mathbf{V})$ is their product generated by using Mul.

By $\|\mathbf{U}_i\| = O(n^4), i \in [\kappa+1]$, we get

$$\|\mathbf{U}\| = \|\prod_{i=1}^{\kappa+1} \mathbf{U}_i\| = O(n^{\kappa} O(n^{4(\kappa+1)})) = O(n^{5\kappa+4}).$$

For simplicity, let $\mathbf{V} = \prod_{i=1}^{\kappa} \mathbf{U}_i \times \mathbf{V}_{\kappa+1} = \mathbf{US} + \mathbf{E}$. Hence, we have

$$\|\mathbf{E}\| = \|\prod_{i=1}^{\kappa} \mathbf{U}_i \times \widetilde{\mathbf{E}}_{\kappa+1}\| = n \times O(n^{5(\kappa-1)+4}) \times O(n^4) = O(n^{5\kappa+4}).$$

On one hand, if $\mathbf{U} = \mathbf{0}$, then $\mathbf{V} = [\mathbf{E}]_q = \mathbf{E}$. Namely, $\mathbf{V}$ is not reduced modulo q. Hence, we obtain

$$\begin{aligned}\|\mathbf{u}^T\mathbf{V}\mathbf{v}\| &= \|\mathbf{u}^T\mathbf{E}\mathbf{v}\| \\ &< n^2\|\mathbf{u}\|\|\mathbf{E}\|\|\mathbf{v}\| \\ &< n^2 O(n^{1.5}) O(n^{5\kappa+4}) O(n^{1.5}) \\ &= O(n^{5\kappa+9}) \\ &< q^{7/8}.\end{aligned}$$

In the setting of parameters, we set $O(n^{5.7\kappa+10.3}) < q < O(n^{6.7\kappa+4})$ to satisfy the condition $q^{3/4} < O(n^{5\kappa+9}) < q^{7/8}$.

On the other hand, if $\mathbf{U} \neq \mathbf{0}$, then $\|\mathbf{u}^T\mathbf{V}\mathbf{v}\| \approx \|\mathbf{u}^T\mathbf{U}\mathbf{S}\mathbf{v}\| \approx q > q^{7/8}$ with overwhelming probability, since $\|\mathbf{S}\| \approx q$.

Thus, the zero-test procedure isZero$(par, (\mathbf{U}, \mathbf{V}))$ is correct. ∎

Lemma 3.7 For two encodings $(\mathbf{U}_1, \mathbf{V}_1)$, $(\mathbf{U}_2, \mathbf{V}_2)$, if $\mathbf{U}_1 = \mathbf{U}_2$, then $\mathrm{Ext}(par, (\mathbf{U}_1, \mathbf{V}_1)) = \mathrm{Ext}(par, (\mathbf{U}_2, \mathbf{V}_2))$.

Proof. Since $\mathbf{V}_i = \mathbf{U}_i\mathbf{S} + \mathbf{E}_i, i \in [2]$, we have

$$\begin{aligned}\mathbf{u}^T\mathbf{V}_1\mathbf{v} &= \mathbf{u}^T\mathbf{U}_1\mathbf{S}\mathbf{v} + \mathbf{u}^T\mathbf{E}_1\mathbf{v}, \\ \mathbf{u}^T\mathbf{V}_2\mathbf{v} &= \mathbf{u}^T\mathbf{U}_2\mathbf{S}\mathbf{v} + \mathbf{u}^T\mathbf{E}_2\mathbf{v}.\end{aligned}$$

By $\mathbf{U}_1 = \mathbf{U}_2$, we obtain $\mathbf{u}^T\mathbf{V}_1\mathbf{v} - \mathbf{u}^T\mathbf{V}_2\mathbf{v} = \mathbf{u}^T\mathbf{E}_1\mathbf{v} - \mathbf{u}^T\mathbf{E}_2\mathbf{v}$.

Again since $\|\mathbf{u}^T\mathbf{E}_i\mathbf{v}\| < q^{7/8}, i \in [2]$, we have $\|\mathbf{u}^T\mathbf{E}_1\mathbf{v} - \mathbf{u}^T\mathbf{E}_2\mathbf{v}\| \leq 2q^{7/8}$.

Furthermore, when $\mathbf{U}_i \neq \mathbf{0}, i \in [2]$, by Lemma 3.6, $\|\mathbf{u}^T\mathbf{U}_i\mathbf{S}\mathbf{v}\| \approx q$ with overwhelming probability.

Hence, the $\eta = (\mathtt{log}q)/8 - \lambda$ most-significant bits from each of the n coefficients of $\mathbf{u}^T\mathbf{V}_i\mathbf{v}$ is determined by the term $\mathbf{u}^T\mathbf{U}_i\mathbf{S}\mathbf{v}$. That is, $\mathrm{Ext}(par, (\mathbf{U}_1, \mathbf{V}_1)) = \mathrm{Ext}(par, (\mathbf{U}_2, \mathbf{V}_2))$ with overwhelming probability. ∎

3.3 Hardness Assumptions

The security of our construction depends on the hardness of vRLWE, and cannot be reduced to classic hard problems, such as hard lattice problems or Ring-LWE/LWE. Similarly [31], we define the extraction version of GCDH/GDDH for our construction. Consider the following security experiment:

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

(2) For $j = 1$ to $k \leq \kappa + 1$:
(2.1) Sample $d_{l,j} \leftarrow D_{\mathbb{Z}^n,\sigma}, l \in [\tau]$,
(2.2) Generate an encoding $\mathbf{U}_j = \sum_{l=1}^{\tau} d_{l,j}\mathbf{A}_l$, $\mathbf{V}_j = \big[\sum_{l=1}^{\tau} d_{l,j}\mathbf{B}_l\big]_q$.
(3) Set $\mathbf{U} = \prod_{j=1}^{k} \mathbf{U}_j, \mathbf{V} = \big[\prod_{j=1}^{k-1} \mathbf{U}_j\mathbf{V}_k\big]_q$.
(4) Set $w_C = w_D = \mathrm{Ext}(par, (\mathbf{U}, \mathbf{V}))$.
(5) Sample $r_l \leftarrow D_{\mathbb{Z}^n,\sigma}, l \in [\tau]$, and set

$$\mathbf{U}_r = \sum_{l=1}^{\tau} r_l\mathbf{A}_l, \mathbf{V}_r = \big[\sum_{l=1}^{\tau} r_l\mathbf{B}_l\big]_q,$$
$$\widetilde{\mathbf{U}}_r = \prod_{j=1}^{k-1} \mathbf{U}_j \times \mathbf{U}_r, \widetilde{\mathbf{V}}_r = \big[\prod_{j=1}^{k-1} \mathbf{U}_j \times \mathbf{V}_r\big]_q,$$
$$w_R = \mathrm{Ext}(par, (\widetilde{\mathbf{U}}_r, \widetilde{\mathbf{V}}_r)).$$

Definition 3.8 (Ext-GCDH/Ext-GDDH). According to the security experiment, the Ext-GCDH and Ext-GDDH are defined as follows:

Extraction CDH (Ext-GCDH): Given $\{par, \mathbf{U}_j, j \in [k]\}$, output a level-k extraction bits string w such that $w = v_C$.

Extraction DDH (Ext-GDDH): Given $\{par, \mathbf{U}_j, j \in [k], w\}$, distinguish between $D_{\text{Ext-GDDH}}$, and $D_{\text{Ext-RAND}}$:

$$D_{\text{Ext-GDDH}} = \{par, \mathbf{U}_j, j \in [k], w_D\}, D_{\text{Ext-RAND}} = \{par, \mathbf{U}_j, j \in [k], w_R\}.$$

For the above construction, we assume that the Ext-GCDH/Ext-GDDH is hard.

3.4 Cryptanalysis

This variant can be transform into $\mathbf{S} = \mathbf{A}^{-1}$, that is $\mathbf{A}_1^{-1}\mathbf{B}_1 = [\mathbf{S} + \mathbf{A}_1^{-1}\mathbf{E}_1]_q$, $\mathbf{A}_2^{-1}\mathbf{B}_2 = [\mathbf{S} + \mathbf{A}_2^{-1}\mathbf{E}_2]_q$, then we have $\mathbf{A}_1(\mathbf{A}_1^{-1}\mathbf{B}_1 - \mathbf{A}_2^{-1}\mathbf{B}_2) = \mathbf{A}_1(\mathbf{A}_1^{-1}\mathbf{E}_1 - \mathbf{A}_2^{-1}\mathbf{E}_2) = \mathbf{E}_1 - \mathbf{A}_1\mathbf{A}_2^{-1}\mathbf{E}_2$. Solving this problem is not easier than LWE.

4 Asymmetric Commutative Variant

In the above construction, the plaintext matrices $\mathbf{A}_l$ and the secret matrix $\mathbf{S}$ of encoding do not support the commutative law of multiplication. However, there exist some applications that require at least one of them to satisfy the commutative law of multiplication. To meet these applications, we construct an asymmetric variant by using a secret commutative matrix $\mathbf{S} = s\mathbf{I}$. But, when $\mathbf{S}$ is chosen to be a ring element s, it is necessary to deploy some additional safeguards in order to guarantee the security of this variant.

4.1 Construction

Setting the parameters. We let λ be the security parameter, κ the length of multilinearity edges, and take $n = O(\lambda)$, $\mu = \kappa + 1$, $\sigma_0 = O(n^3)$, $\sigma_1 = O(n)$, $O(n^{8\kappa+13}) < q < O(n^{9\kappa+17})$, $\sigma_2 = O(n^3)$, $d \geq 2$, $\tau = \lambda d^2$.

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

(1) Choose a prime $O(n^{8\kappa+13}) < q < O(n^{9\kappa+17})$.

(2) Choose random matrices $\mathbf{T}_k \leftarrow R_q^{d\times d}, k = 0, \cdots, \mu$ such that $\mathbf{T}_k^{-1} \in R_q^{d\times d}$.

(3) Choose random elements $s, z \in R_q$.

(4) For $l \in [\tau], k \in [\mu]$, sample $\mathbf{A}_{l,k}, \mathbf{E}_{l,k} \in R^{d\times d}$ such that $a_{l,k,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma_0}, e_{l,k,i,j} \leftarrow D_{\mathbb{Z}^n,\sigma_2}, i, j \in [d]$.

(5) Sample $\mathbf{u}, \mathbf{v} \in R^d$ such that $u_i, v_i \leftarrow D_{\mathbb{Z}^n,\sigma_1}, i \in [d]$.

(6) For $l \in [\tau], k \in [\mu]$, set

$$\begin{aligned}\overline{\mathbf{A}}_{l,k} &= [\mathbf{T}_{k-1}^{-1}\mathbf{A}_{l,k}\mathbf{T}_k/z]_q,\\ \overline{\mathbf{B}}_{l,k} &= [\mathbf{T}_{k-1}^{-1}(\mathbf{A}_{l,k}\cdot s + \mathbf{E}_{l,k})\mathbf{T}_k \cdot z^\kappa]_q,\\ \overline{\mathbf{u}}^T &= \mathbf{u}^T\mathbf{T}_0, \overline{\mathbf{v}} = \mathbf{T}_\mu^{-1}\mathbf{v}.\end{aligned}$$

(7) Output the public parameters $par = \big\{q, (\overline{\mathbf{A}}_{l,k}, \overline{\mathbf{B}}_{l,k})_{l\in[\tau],k\in[\mu]}, \overline{\mathbf{u}}, \overline{\mathbf{v}}\big\}$.

Generating a k-edge encoding: $(\mathbf{U}_k, \mathbf{V}_k) \leftarrow \mathrm{Enc}(par, k)$.

Given $r_{l,k} \leftarrow D_{\mathbb{Z}^n,\sigma_1}$, $l \in [\tau]$, generate

$$\mathbf{U}_k = \big[\sum\nolimits_{l=1}^{\tau} r_{l,k}\cdot\overline{\mathbf{A}}_{l,k}\big]_q, \mathbf{V}_k = \big[\sum\nolimits_{l=1}^{\tau} r_{l,k}\cdot\overline{\mathbf{B}}_{l,k}\big]_q.$$

Adding same edge encodings:

$(\mathbf{U}_k, \mathbf{V}_k) \leftarrow \mathrm{Add}(par, (\mathbf{U}_{1,k}, \mathbf{V}_{1,k}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k}))$.

Given two k-edge encodings $(\mathbf{U}_{1,k}, \mathbf{V}_{1,k}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k})$, compute

$$\mathbf{U}_k = [\mathbf{U}_{1,k} + \mathbf{U}_{2,k}]_q, \mathbf{V}_k = [\mathbf{V}_{1,k} + \mathbf{V}_{2,k}]_q.$$

In the following, we use the subscript $k_1 \to k_2$ to denote a connected path encoding from k_1-edge to k_2-edge.

Multiplying adjacent edge encodings:

$(\mathbf{U}_{k-1\to k}, \mathbf{V}_{k-1\to k}) \leftarrow \mathrm{Mul}(par, (\mathbf{U}_{1,k-1}, \mathbf{V}_{1,k-1}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k}))$.

Given two adjacent edge encodings $(\mathbf{U}_{1,k-1}, \mathbf{V}_{1,k-1}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k})$, compute

$$\mathbf{U}_{k-1\to k} = [\mathbf{U}_{1,k-1}\mathbf{U}_{2,k}]_q, \mathbf{V}_{k-1\to k} = [\mathbf{U}_{1,k-1}\mathbf{V}_{2,k}]_q.$$

Given the connected edge encodings $(\mathbf{U}_{k_1\to k_2}, \mathbf{V}_{k_1\to k_2}), (\mathbf{U}_{k_2+1\to k_3}, \mathbf{V}_{k_2+1\to k_3})$, we can similarly multiply them to generate a $(k_1 \to k_3)$ encoding as follows:

$$\mathbf{U}_{k_1\to k_3} = [\mathbf{U}_{k_1\to k_2}\mathbf{U}_{k_2+1\to k_3}]_q, \mathbf{V}_{k_1\to k_3} = [\mathbf{U}_{k_1\to k_2}\mathbf{V}_{k_2+1\to k_3}]_q.$$

Zero-testing: $\mathrm{isZero}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu}))$:

Given an encoding $(\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})$, we check whether $\|\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}\|$ is short:

$$\mathrm{isZero}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})) = \begin{cases}1, & \text{if } \|\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}\| < q^{7/8};\\ 0, & \text{otherwise.}\end{cases}$$

Extract: $sk \leftarrow \mathrm{Ext}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu}))$.

Given an encoding $(\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})$, we extract the $\eta = (\mathtt{log}q)/8 - \lambda$ most-significant bits from each of the n coefficients of $\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}$:

$$\mathrm{Ext}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})) = \mathrm{msbs}_\eta(\overline{\mathbf{u}}^T \mathbf{V}_{1\to\mu} \overline{\mathbf{v}}),$$

where msbs_η extracts the η most significant bits from each coefficient of $\overline{\mathbf{u}}^T \mathbf{V}_{1\to\mu} \overline{\mathbf{v}}$.

Remark 4.1 (1) For the asymmetric variant, we can generate a new form by modifying $\overline{\mathbf{B}}_{l,k} = [\mathbf{T}_{k-1}^{-1}(\mathbf{A}_{l,k} \cdot s + \mathbf{E}_{l,k} \cdot z^\kappa)\mathbf{T}_k]_q$. In essence, this form is equivalent to the above asymmetric variant, since $\overline{\mathbf{B}}_{l,k} = [\mathbf{T}_{k-1}^{-1}(\mathbf{A}_{l,k} \cdot (s/z^\kappa) + \mathbf{E}_{l,k})\mathbf{T}_k \cdot z^\kappa]_q$.

(2) In the asymmetric variant, we can also choose $\mathbf{u}, \mathbf{v}$ such that $u_i, v_i \leftarrow D_{\mathbb{Z}^n, q^{1/4}}, i \in [d]$. In this case, we set $\sigma_0 = O(n)$ to reduce the size of modulus q and improve efficiency.

(3) We can construct a symmetric variant. To immune the possible subfield lattice attack [1], we choose $\mathbf{u}, \mathbf{v}$ such that $u_i, v_i \leftarrow D_{\mathbb{Z}^n, q^{1/4}}, i \in [d]$. That is, we generate the following public parameters:

$$\begin{aligned}
\overline{\mathbf{A}}_l &= [\mathbf{T}\mathbf{A}_l\mathbf{T}^{-1}/z]_q, \\
\overline{\mathbf{B}}_l &= [\mathbf{T}(\mathbf{A}_l \cdot s + \mathbf{E}_l)\mathbf{T}^{-1} z^\kappa z_1]_q, \\
\overline{\mathbf{u}}^T &= \mathbf{u}^T\mathbf{T}^{-1} z_0, \overline{\mathbf{v}} = \mathbf{T}\mathbf{v}/(z_0 z_1),
\end{aligned}$$

where $z_0, z_1, z \in R_q$ are the random ring elements.

(4) To improve the security of the symmetric variant, we can also choose a ring that can immune the possible subfield lattice attack [1], e.g. $R = \mathbb{Z}[x]/\langle f(x)\rangle$, where $f(x) = x^p - 1$ with p a safe prime (or $f(x) = x^p - x - 1$ with p a prime).

(5) If we set $s = h/g$ with $g \leftarrow D_{\mathbb{Z}^n, \sigma_0}, h \leftarrow D_{\mathbb{Z}^n, \sigma_0}$, then our construction becomes a new variant of GGH13 [19].

4.2 Correctness

In the asymmetric variant, we replace $\mathbf{S}$ with a commutative matrix $s\mathbf{I}$, use Kilian randomization method, and remain others unchanged. So, its correctness directly follows that of the above construction. For completeness, we give the brief proof of correctness in the following.

Lemma 4.2 The algorithm $\mathrm{InstGen}(1^\lambda, 1^\kappa)$ runs in polynomial time.

Proof. Since each step in InstGen takes in polynomial time, the result directly follows. ∎

Lemma 4.3 $(\mathbf{U}_k, \mathbf{V}_k) \leftarrow \mathrm{Enc}(par, k)$ is a k-layer encoding.

Proof. Given $r_{l,k} \leftarrow D_{\mathbb{Z}^n, \sigma_1}$, $l \in [\tau]$, we have

$$\begin{aligned}
\mathbf{U}_k &= [\sum\nolimits_{l=1}^{\tau} r_{l,k} \cdot \overline{\mathbf{A}}_{l,k}]_q = [\mathbf{T}_{k-1}^{-1}\mathbf{C}_k\mathbf{T}_k/z]_q, \\
\mathbf{V}_k &= [\sum\nolimits_{l=1}^{\tau} r_{l,k} \cdot \overline{\mathbf{B}}_{l,k}]_q = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}_k \cdot s + \mathbf{D}_k)\mathbf{T}_k \cdot z^\kappa]_q,
\end{aligned}$$

where $\mathbf{C}_k = \sum_{l=1}^{\tau} r_{l,k}\mathbf{A}_{l,k}$, $\mathbf{D}_k = \sum_{l=1}^{\tau} r_{l,k}\mathbf{E}_{l,k}$. ∎

Lemma 4.4 The encoding $(\mathbf{U}_k, \mathbf{V}_k) \leftarrow \mathrm{Add}(par, (\mathbf{U}_{1,k}, \mathbf{V}_{1,k}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k}))$ is a level-k random encoding.

Proof. Given two level-k encodings $(\mathbf{U}_{1,k}, \mathbf{V}_{1,k}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k})$, we have

$$\begin{aligned}
\mathbf{U}_k &= [\mathbf{U}_{1,k} + \mathbf{U}_{2,k}]_q = [\mathbf{T}_{k-1}^{-1}\mathbf{C}_k\mathbf{T}_k/z]_q, \\
\mathbf{V}_k &= [\mathbf{V}_{1,k} + \mathbf{V}_{2,k}]_q = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}_k \cdot s + \mathbf{D}_k)\mathbf{T}_k \cdot z^\kappa]_q,
\end{aligned}$$

where $\mathbf{U}_{i,k} = [\mathbf{T}_{k-1}^{-1}\mathbf{C}_{i,k}\mathbf{T}_k/z]_q$, $\mathbf{V}_{i,k} = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}_{i,k}\cdot s + \mathbf{D}_{i,k})\mathbf{T}_k \cdot z^{\kappa}]_q$, and $\mathbf{C}_k = \mathbf{C}_{1,k} + \mathbf{C}_{2,k}$, $\mathbf{D}_k = \mathbf{D}_{1,k} + \mathbf{D}_{2,k}$. ∎

Lemma 4.5 $(\mathbf{U}_{k-1\to k}, \mathbf{V}_{k-1\to k}) \leftarrow \mathrm{Mul}(par, (\mathbf{U}_{1,k-1}, \mathbf{V}_{1,k-1}), (\mathbf{U}_{2,k}, \mathbf{V}_{2,k}))$ a level-$(k-1) \to k$ random encoding.

Proof. Given a level-$(k-1)$ encoding $(\mathbf{U}_{1,k-1}, \mathbf{V}_{1,k-1})$ and a level-k encoding $(\mathbf{U}_{2,k}, \mathbf{V}_{2,k})$, we have

$$\mathbf{U}_{k-1\to k} = [\mathbf{U}_{1,k-1}\mathbf{U}_{2,k}]_q = [\mathbf{T}_{k-2}^{-1}\mathbf{C}_{k-1\to k}\mathbf{T}_k/z^2]_q,$$
$$\mathbf{V}_{k-1\to k} = [\mathbf{U}_{1,k-1}\mathbf{V}_{2,k}]_q = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}_{k-1\to k}\cdot s + \mathbf{D}_{k-1\to k})\mathbf{T}_k \cdot z^{\kappa-1}]_q,$$

where $\mathbf{U}_{i,j} = [\mathbf{T}_{j-1}^{-1}\mathbf{C}_{i,j}\mathbf{T}_j/z]_q$, $\mathbf{V}_{i,j} = [\mathbf{T}_{j-1}^{-1}(\mathbf{C}_{i,j}\cdot s + \mathbf{D}_{i,j})\mathbf{T}_j \cdot z^{\kappa}]_q$, and $\mathbf{C}_{k-1\to k} = \mathbf{C}_{1,k-1}\mathbf{C}_{2,k}$, $\mathbf{D}_{k-1\to k} = \mathbf{C}_{1,k-1}\mathbf{D}_{2,k}$. ∎

Furthermore, we can also perform multiplication as follows:

$$\mathbf{V}'_{k-1\to k} = [\mathbf{V}_{1,k-1}\mathbf{U}_{2,k}]_q = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}'_{k-1\to k}\cdot s + \mathbf{D}'_{k-1\to k})\mathbf{T}_k \cdot z^{\kappa-1}]_q,$$

where $\mathbf{C}'_{k-1\to k} = \mathbf{C}_{1,k-1}\mathbf{C}_{2,k}$, $\mathbf{D}'_{k-1\to k} = \mathbf{D}_{1,k-1}\mathbf{C}_{2,k}$.

It is easy to see that the plaintexts $\mathbf{C}'_{k-1\to k}\cdot s = \mathbf{C}_{k-1\to k}\cdot s$, but the noise terms $\mathbf{D}_{k-1\to k} \neq \mathbf{D}'_{k-1\to k}$.

For two connected level encodings $(\mathbf{U}_{k_1\to k_2}, \mathbf{V}_{k_1\to k_2})$, $(\mathbf{U}_{k_2+1\to k_3}, \mathbf{V}_{k_2+1\to k_3})$, we can similarly show that their product is a $(k_1 \to k_3)$ encoding.

Lemma 4.6 The procedure $\mathrm{isZero}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu}))$ can correctly determine whether $(\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})$ is an encoding of zero.

Proof. Given a level-1 $\to \mu$ encoding $(\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu})$, we assume that it is the product of encodings $(\mathbf{U}_k, \mathbf{V}_k), k \in [\mu]$, which are encodings generated by $\mathrm{Enc}(par, k)$, respectively.

Without loss of generality, we further assume

$$\mathbf{U}_{1\to\mu} = \big[\prod_{k=1}^{\mu} \mathbf{U}_k\big]_q,$$
$$\mathbf{V}_{1\to\mu} = \big[\prod_{k=1}^{j-1} \mathbf{U}_k \times \mathbf{V}_j \times \prod_{k=j+1}^{\mu} \mathbf{U}_k\big]_q, j \in [\mu].$$

By $\mathbf{U}_k = [\mathbf{T}_{k-1}^{-1}\mathbf{C}_k\mathbf{T}_k/z]_q$, and $\mathbf{V}_k = [\mathbf{T}_{k-1}^{-1}(\mathbf{C}_k\cdot s + \mathbf{D}_k)\mathbf{T}_k \cdot z^{\kappa}]_q$, we have

$$\mathbf{U}_{1\to\mu} = \big[\mathbf{T}_0 \prod_{k=1}^{\mu} \mathbf{C}_k\mathbf{T}_\mu/z^{\mu}\big]_q,$$
$$\mathbf{V}_{1\to\mu} = \big[\mathbf{T}_0(\prod_{k=1}^{\mu} \mathbf{C}_k\cdot s + \prod_{k=1}^{j-1} \mathbf{C}_k \times \mathbf{D}_j \times \prod_{k=j+1}^{\mu} \mathbf{C}_k)\mathbf{T}_\mu\big]_q$$
$$= \big[\mathbf{T}_0(\mathbf{C}_{1\to\mu}\cdot s + \mathbf{D}_{1\to\mu})\mathbf{T}_\mu\big]_q,$$

where $\mathbf{C}_{1\to\mu} = \prod_{k=1}^{\mu} \mathbf{C}_k$, $\mathbf{D}_{1\to\mu} = \prod_{k=1}^{j-1} \mathbf{C}_k \times \mathbf{D}_j \times \prod_{k=j+1}^{\mu} \mathbf{C}_k$.

Hence, we obtain $[\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}]_q = [\mathbf{u}^T(\mathbf{C}_{1\to\mu}\cdot s + \mathbf{D}_{1\to\mu})\mathbf{v}]_q$.

Now, we evaluate the value $\|[\mathbf{u}^T(\mathbf{C}_{1\to\mu}\cdot s + \mathbf{D}_{1\to\mu})\mathbf{v}]_q\|$.

By Lemma 4.3, we know $\mathbf{C}_k = \sum_{l=1}^{\tau} r_{l,k}\mathbf{A}_{l,k}$. Using Lemma 2.2, we have $\|\mathbf{A}_{l,k}\| = O(\sigma_0\sqrt{n}) = O(n^{3.5})$, $\|r_{l,k}\| = O(\sigma_1\sqrt{n}) = O(n^{1.5})$.

So, we get $\|\mathbf{C}_k\| = \tau \cdot n \cdot O(n^{1.5}) \cdot O(n^{3.5}) = O(n^6)$.

Again by Lemma 2.2 and Lemma 4.3, we have $\mathbf{D}_k = \sum_{l=1}^{\tau} r_{l,k}\mathbf{E}_{l,k}$, and $\|r_{l,k}\| = O(n^{1.5})$, $\|\mathbf{E}_{l,k}\| = O(\sigma_2\sqrt{n}) = O(n^{3.5})$. So, $\|\mathbf{D}_k\| = O(n^6)$.

On one hand, if $\mathbf{C}_{1\to\mu} = \mathbf{0}$, then we have

$$[\mathbf{u}^T(\mathbf{C}_{1\to\mu} \cdot s + \mathbf{D}_{1\to\mu})\mathbf{v}]_q = [\mathbf{u}^T\mathbf{D}_{1\to\mu}\mathbf{v}]_q = \mathbf{u}^T\mathbf{D}_{1\to\mu}\mathbf{v}.$$

Namely, $\mathbf{u}^T\mathbf{D}_{1\to\mu}\mathbf{v}$ is not reduced modulo q. Hence, we obtain

$$\begin{aligned}\|[\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}]_q\| &= \|\mathbf{u}^T\mathbf{D}_{1\to\mu}\mathbf{v}\| \\ &< n^2\|\mathbf{u}\|\|\mathbf{D}_{1\to\mu}\|\|\mathbf{v}\| \\ &< n^2O(n^{1.5})O(n^{\kappa}\cdot(n^6)^{\kappa}\cdot O(n^6))O(n^{1.5}) \\ &= O(n^{7\kappa+11}) \\ &< q^{7/8}.\end{aligned}$$

Furthermore, to immune the possible subfield lattice [1], we require

$$\begin{aligned}\|\mathbf{u}^T\textstyle\prod_{k=1}^{\mu}\mathbf{A}_{l,k}\mathbf{v}\| &= O(n^2\cdot O(n^{1.5})\cdot O(n^{\kappa}\cdot(n^{3.5})^{\kappa+1})\cdot O(n^{1.5})) \\ &= O(n^{4.5\kappa+8.5}) \\ &> q^{1/2}.\end{aligned} \tag{1}$$

So, we set $O(n^{8\kappa+13}) < q < O(n^{9\kappa+17})$, the above conditions are satisfied.

On the other hand, if $\mathbf{C}_{1\to\mu} \neq \mathbf{0}$, then

$$\begin{aligned}\|[\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}\overline{\mathbf{v}}]_q\| &= \|[\mathbf{u}^T(\mathbf{C}_{1\to\mu}\cdot s + \mathbf{D}_{1\to\mu})\mathbf{v}]_q\| \\ &> \|[\mathbf{u}^T(\mathbf{C}_{1\to\mu}\cdot s)\mathbf{v}]_q\| - \|[\mathbf{u}^T\mathbf{D}_{1\to\mu}\mathbf{v}]_q\| \\ &> q - q^{7/8} \\ &> q^{7/8},\end{aligned}$$

where $\|s\| \approx q$.

Thus, the zero-test procedure isZero$(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu}))$ is correct. ∎

Lemma 4.7 Given two encodings $(\mathbf{U}^{(t)}_{1\to\mu}, \mathbf{V}^{(t)}_{1\to\mu}), t \in [2]$, if $\mathbf{U}^{(1)}_{1\to\mu} = \mathbf{U}^{(2)}_{1\to\mu}$, then $\mathrm{Ext}(par, (\mathbf{U}^{(1)}_{1\to\mu}, \mathbf{V}^{(1)}_{1\to\mu})) = \mathrm{Ext}(par, (\mathbf{U}^{(2)}_{1\to\mu}, \mathbf{V}^{(2)}_{1\to\mu}))$.

Proof. For the encodings $(\mathbf{U}^{(t)}_{1\to\mu}, \mathbf{V}^{(t)}_{1\to\mu}), t \in [2]$, we have

$$\begin{aligned}\mathbf{U}^{(t)}_{1\to\mu} &= \left[\mathbf{T}_0\mathbf{C}^{(t)}_{1\to\mu}\mathbf{T}_{\mu}/z^{\mu}\right]_q, \\ \mathbf{V}^{(t)}_{1\to\mu} &= \left[\mathbf{T}_0(\mathbf{C}^{(t)}_{1\to\mu}\cdot s + \mathbf{D}^{(t)}_{1\to\mu})\mathbf{T}_{\mu}\right]_q.\end{aligned}$$

So, $[\overline{\mathbf{u}}^T\mathbf{V}^{(t)}_{1\to\mu}\overline{\mathbf{v}}]_q = [\mathbf{u}^T(\mathbf{C}^{(t)}_{1\to\mu}\cdot s)\mathbf{v} + \mathbf{u}^T\mathbf{D}^{(t)}_{1\to\mu}\mathbf{v}]_q, t \in [2]$.

By $\mathbf{U}^{(1)}_{1\to\mu} = \mathbf{U}^{(2)}_{1\to\mu}$, we get $\mathbf{C}^{(1)}_{1\to\mu} = \mathbf{C}^{(2)}_{1\to\mu}$. Hence, we obtain

$$[\overline{\mathbf{u}}^T\mathbf{V}^{(1)}_{1\to\mu}\overline{\mathbf{v}} - \overline{\mathbf{u}}^T\mathbf{V}^{(2)}_{1\to\mu}\overline{\mathbf{v}}]_q = [\mathbf{u}^T\mathbf{D}^{(1)}_{1\to\mu}\mathbf{v} - \mathbf{u}^T\mathbf{D}^{(2)}_{1\to\mu}\mathbf{v}]_q.$$

Again by Lemma 4.6, $\|\mathbf{u}^T\mathbf{D}^{(t)}_{1\to\mu}\mathbf{v}\| < q^{7/8}, t \in [2]$. Namely, $\|\mathbf{u}^T\mathbf{D}^{(1)}_{1\to\mu}\mathbf{v} - \mathbf{u}^T\mathbf{D}^{(2)}_{1\to\mu}\mathbf{v}\| < 2q^{7/8}$.

Futhermore, when $\mathbf{C}^{(t)}_{1\to\mu} \neq \mathbf{0}, t \in [2]$, we have $\|[\mathbf{u}^T(\mathbf{C}^{(t)}_{1\to\mu} \cdot s)\mathbf{v}]_q\| \approx q$ with overwhelming probability.

Hence, the $\eta = (\log q)/8 - \lambda$ most-significant bits from each of the n coefficients of $[\overline{\mathbf{u}}^T\mathbf{V}^{(t)}_{1\to\mu}\overline{\mathbf{v}}]_q$ is determined by the term $\|[\mathbf{u}^T(\mathbf{C}^{(t)}_{1\to\mu} \cdot s)\mathbf{v}]_q\|$. That is, $\mathrm{Ext}(par, (\mathbf{U}^{(1)}_{1\to\mu}, \mathbf{V}^{(1)}_{1\to\mu})) = \mathrm{Ext}(par, (\mathbf{U}^{(2)}_{1\to\mu}, \mathbf{V}^{(2)}_{1\to\mu}))$ with overwhelming probability. ∎

4.3 Hardness Assumptions

The security of our asymmetric variant depends on new hardness assumption, which is similar to the hardness assumption of vRLWE. For completeness, we also define the extraction version of GCDH/GDDH for this variant. Consider the following security experiment:

(1) $(par) \leftarrow \mathrm{InstGen}(1^\lambda, 1^\kappa)$.
(2) For $k = 1$ to μ:
(2.1) Sample $d_{l,k} \leftarrow D_{\mathbb{Z}^n,\sigma_1}, l \in [\tau]$,
(2.2) Generate an encoding $\mathbf{U}_k = \sum_{l=1}^{\tau} d_{l,k}\overline{\mathbf{A}}_{l,k}$, $\mathbf{V}_k = \big[\sum_{l=1}^{\tau} d_{l,k}\overline{\mathbf{B}}_{l,k}\big]_q$.
(3) Set $\mathbf{U}_{1\to\mu} = \prod_{k=1}^{\mu}\mathbf{U}_k, \mathbf{V}_{1\to\mu} = \big[\prod_{k=1}^{\kappa}\mathbf{U}_k \times \mathbf{V}_\mu\big]_q$.
(4) Set $w_C = w_D = \mathrm{Ext}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}_{1\to\mu}))$.
(5) Sample $r_l \leftarrow D_{\mathbb{Z}^n,\sigma}, l \in [\tau]$, and set

$$\begin{aligned}
\mathbf{U}_r &= \sum_{l=1}^{\tau} r_l\overline{\mathbf{A}}_{l,k}, \mathbf{V}_r = \big[\sum_{l=1}^{\tau} r_l\overline{\mathbf{B}}_{l,k}\big]_q,\\
\widetilde{\mathbf{U}}_{1\to\mu} &= \prod_{k=1}^{\kappa}\mathbf{U}_k \times \mathbf{U}_r, \widetilde{\mathbf{V}}_{1\to\mu} = \big[\prod_{k=1}^{\kappa}\mathbf{U}_k \times \mathbf{V}_r\big]_q,\\
w_R &= \mathrm{Ext}(par, (\widetilde{\mathbf{U}}_{1\to\mu}, \widetilde{\mathbf{V}}_{1\to\mu})).
\end{aligned}$$

Definition 4.8 (Ext-GCDH/Ext-GDDH). According to the security experiment, the Ext-GCDH and Ext-GDDH are defined as follows:

Extraction CDH (Ext-GCDH): Given $\{par, \mathbf{U}_k, k \in [\mu]\}$, output a level-κ extraction bits string w such that $w = v_C$.

Extraction DDH (Ext-GDDH): Given $\{par, \mathbf{U}_k, k \in [\mu], w\}$, distinguish between $D_{\text{Ext-GDDH}}$, and $D_{\text{Ext-RAND}}$:

$$D_{\text{Ext-GDDH}} = \{par, \mathbf{U}_k, k \in [\mu], w_D\}, D_{\text{Ext-RAND}} = \{par, \mathbf{U}_k, k \in [\mu], w_R\}.$$

Note that in the above security experiment, if assuming $\mathbf{V}^{(j)}_{1\to\mu} = \big[\prod_{k=1}^{j-1}\mathbf{U}_k \times \mathbf{V}_j \times \prod_{k=j+1}^{\mu}\mathbf{U}_k\big]_q, j \in [\mu]$, then $w_C = \mathrm{Ext}(par, (\mathbf{U}_{1\to\mu}, \mathbf{V}^{(j)}_{1\to\mu}))$. Namely, the extraction bit strings of $\mathbf{V}^{(j)}_{1\to\mu}, j \in [\mu]$ are all same with overwhelming probability.

For this asymmetric variant, we also assume that its Ext-GCDH/Ext-GDDH is hard.

4.4 Cryptanalysis

In this section, we give easily computable some quantities in our asymmetric variant, and analyze possible attacks using these quantities. According to our analysis, currently known attacks does not seem to work for our asymmetric construction.

1. Easily computable quantities

Given the public parameters par, we can perform cross-multiplication between encodings of the asymmetric variant to obtain some non-reduced quantities.

For notational simplicity, we denote

$$\mathbf{C}^{(j)}_{l,r,s,t} = [\prod_{k=1}^{j-1} \overline{\mathbf{A}}_{l,k} \cdot \overline{\mathbf{B}}_{r,j} \overline{\mathbf{A}}_{s,j+1} \cdot \prod_{k=j+2}^{\mu} \overline{\mathbf{A}}_{t,k}]_q,$$
$$\mathbf{D}^{(j)}_{l,r,s,t} = [\prod_{k=1}^{j-1} \overline{\mathbf{A}}_{l,k} \cdot \overline{\mathbf{A}}_{r,j} \overline{\mathbf{B}}_{s,j+1} \cdot \prod_{k=j+2}^{\mu} \overline{\mathbf{A}}_{t,k}]_q.$$

So, we have

$$\begin{aligned} c^{(j)}_{l,r,s,t} &= [\overline{\mathbf{u}}^T (\mathbf{C}^{(j)}_{l,r,s,t} - \mathbf{D}^{(j)}_{l,r,s,t}) \overline{\mathbf{v}}]_q \\ &= [\mathbf{u}^T \prod_{k=1}^{j-1} \mathbf{A}_{l,k} (\mathbf{E}_{r,j}\mathbf{A}_{s,j+1} - \mathbf{A}_{r,j}\mathbf{E}_{s,j+1}) \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k} \mathbf{v}]_q, \\ &= [\mathbf{u}^T \prod_{k=1}^{j-1} \mathbf{A}_{l,k} (\mathbf{E}^{(j)}_{r,s}) \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k} \mathbf{v}]_q \end{aligned}$$

where $\mathbf{E}^{(j)}_{r,s} = \mathbf{E}_{r,j}\mathbf{A}_{s,j+1} - \mathbf{A}_{r,j}\mathbf{E}_{s,j+1}$.

According to the setting of parameters, $c^{(j)}_{l,r,s,t}$ is not reduced modulus q.

Now, we can write $c^{(j)}_{l,r,s,t}$ in the following matrix forms:

$$\begin{aligned} c^{(j)}_{l,r,s,t} &= \mathbf{u}^T \prod_{k=1}^{j-1} \mathbf{A}_{l,k} (\mathbf{E}_{r,j}\mathbf{A}_{s,j+1} - \mathbf{A}_{r,j}\mathbf{E}_{s,j+1}) \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k} \mathbf{v} \\ &= \begin{pmatrix} \mathbf{u}^T_{(l)}\mathbf{E}_{r,j} & -\mathbf{u}^T_{(l)}\mathbf{A}_{r,j} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{s,j+1}\mathbf{v}_{(t)} \\ \mathbf{E}_{s,j+1}\mathbf{v}_{(t)} \end{pmatrix} \qquad (2) \\ &= \left(\mathbf{u}^T_{(l)}\right) \left(\mathbf{E}^{(j)}_{r,s}\right) \left(\mathbf{v}_{(t)}\right) \end{aligned}$$

where $\mathbf{u}^T_{(l)} = \mathbf{u}^T \prod_{k=1}^{j-1} \mathbf{A}_{l,k}$, $\mathbf{v}_{(t)} = \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k}\mathbf{v}$.

2. Cheon et al.'s attack [12]

To break CLT13 [15], Cheon et al. [12] first generated two matrices $\mathbf{C}_i = \mathbf{U}\mathbf{E}_i\mathbf{V}, i \in [2]$ from some non-reduced quantities obtained by the public parameters of CLT13, then computed the secret parameters from the eigenvalues of the quotient $\mathbf{C}_1\mathbf{C}_2^{-1}$. The successful key of the Cheon et al.'s attack is that $\mathbf{E}_i, i \in [2]$ are the diagonal matrices. Consequently, Cheon et al. [12] completely broke CLT13.

To perform the Cheon et al.'s attack [12], we generate similar matrices by using the above form (1) of $c^{(j)}_{l,r,s,t}$. Without loss of generality, for $l, t \in [d]$, we

generate $\mathbf{C}_{r,s}^{(j)}$ as follows:

$$\mathbf{C}_{r,s}^{(j)} = \begin{pmatrix} \mathbf{u}_{(1)}^T \\ \vdots \\ \mathbf{u}_{(d)}^T \end{pmatrix} \left(\mathbf{E}_{r,s}^{(j)}\right) \left(\mathbf{v}_{(1)} \cdots \mathbf{v}_{(d)}\right) = \left(\mathbf{U}\right) \left(\mathbf{E}_{r,s}^{(j)}\right) \left(\mathbf{V}\right).$$

So, given $\mathbf{C}_{r_1,s_1}^{(j)}$, $\mathbf{C}_{r_2,s_2}^{(j)}$, we have

$$\mathbf{C}_{r_1,s_1}^{(j)}(\mathbf{C}_{r_2,s_2}^{(j)})^{-1} = \mathbf{U} \times \mathbf{E}_{r_1,s_1}^{(j)}(\mathbf{E}_{r_2,s_2}^{(j)})^{-1} \times \mathbf{U}^{-1}.$$

According to sampling $\mathbf{E}_{r,j}, \mathbf{A}_{s,j+1}, \mathbf{A}_{r,j}, \mathbf{E}_{s,j+1}$ in the procedure InstGen, $\mathbf{E}_{r,s}^{(j)} \in R^{d\times d}$ is not a diagonal matrix with overwhelming probability.

Using the Cheon et al.'s attack [12], we can not find useful information from $\mathbf{C}_{r_1,s_1}^{(j)}(\mathbf{C}_{r_2,s_2}^{(j)})^{-1}$ since the matrices $\mathbf{E}_{r_1,s_1}^{(j)}, \mathbf{E}_{r_2,s_2}^{(j)}$ are not the diagonal matrices.

On the other hand, for $r, s \in [2d]$, we can generate the following matrix $\mathbf{C}_{l,t}^{(j)}$.

$$\begin{aligned} \mathbf{C}_{l,t}^{(j)} &= \begin{pmatrix} \mathbf{u}_{(l)}^T\mathbf{E}_{1,j} & -\mathbf{u}_{(l)}^T\mathbf{A}_{1,j} \\ \vdots & \vdots \\ \mathbf{u}_{(l)}^T\mathbf{E}_{2d,j} & -\mathbf{u}_{(l)}^T\mathbf{A}_{2d,j} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{1,j+1}\mathbf{v}_{(t)} & \cdots & \mathbf{A}_{2d,j+1}\mathbf{v}_{(t)} \\ \mathbf{E}_{1,j+1}\mathbf{v}_{(t)} & \cdots & \mathbf{E}_{2d,j+1}\mathbf{v}_{(t)} \end{pmatrix}, \\ &= \mathbf{U}_{(l)}\mathbf{V}_{(t)}. \end{aligned}$$

Similarly, given $\mathbf{C}_{l,t_1}^{(j)}$, $\mathbf{C}_{l,t_2}^{(j)}$, we can compute

$$\begin{aligned} (\mathbf{C}_{l,t_1}^{(j)})^{-1}\mathbf{C}_{l,t_2}^{(j)} &= (\mathbf{V}_{(t_1)})^{-1}\mathbf{V}_{(t_2)}, \\ \mathbf{C}_{l,t_1}^{(j)}(\mathbf{C}_{l,t_2}^{(j)})^{-1} &= \mathbf{U}_{(l)} \cdot \mathbf{V}_{(t_1)}(\mathbf{V}_{(t_2)})^{-1} \cdot (\mathbf{U}_{(l)})^{-1}. \end{aligned}$$

Since $\mathbf{U}_{(l)}, \mathbf{V}_{(t_1)}, \mathbf{V}_{(t_2)} \in R^{d\times d}$ are not the diagonal matrices, we can not obtain useful information from this form matrix of $\mathbf{C}_{l,t}^{(j)}$.

In summary, our asymmetric variant avoids the Cheon et al.'s attack [12].

3. Coron et al.'s attack [18]

To break GGH15-based MPKE [23], the Coron et al.'s attack [18] includes two steps. First, they express one secret exponent s_1 of User 1 as a linear combination of the other exponents $t_{l,1}$ using a variant of the Cheon et al.'s attack. Second, they generate an equivalent private encoding corresponding to s_1 by correcting the large noise term.

To perform the Coron et al.'s attack [18], we express $c_{l,r,s,t}^{(j)}$ as the following form

$$\begin{aligned} c_{l,r,s,t}^{(j)} &= \mathbf{u}^T \prod_{k=1}^{j-1} \mathbf{A}_{l,k}(\mathbf{E}_{r,s}^{(j)}) \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k}\mathbf{v} \\ &= \mathbf{u}^T\mathbf{A}_{l,1} \prod_{k=2}^{j-1} \mathbf{A}_{l,k}(\mathbf{E}_{r,s}^{(j)}) \prod_{k=j+2}^{\mu} \mathbf{A}_{t,k}\mathbf{v} \\ &= \mathbf{u}_{(l,1)}^T \mathbf{v}_{t,2\to\mu}^{(j)}, \end{aligned} \tag{3}$$

where $\mathbf{u}_{(l,1)}^T = \mathbf{u}^T\mathbf{A}_{l,1}$, $\mathbf{v}_{t,2\to\mu}^{(j)} = \prod_{k=2}^{j-1}\mathbf{A}_{l,k}(\mathbf{E}_{r,s}^{(j)})\prod_{k=j+2}^{\mu}\mathbf{A}_{t,k}\mathbf{v}$.

For, given $l \in [d+1], t \in [d]$, we generate a matrix $\mathbf{C}_{1,2\to\mu}^{(j)}$ as follows:

$$\mathbf{C}_{1,2\to\mu}^{(j)} = \begin{pmatrix} \mathbf{u}_{(1,1)}^T \\ \vdots \\ \mathbf{u}_{(d+1,1)}^T \end{pmatrix} \left(\mathbf{v}_{1,2\to\mu}^{(j)} \cdots \mathbf{v}_{d,2\to\mu}^{(j)}\right) = \mathbf{U}_1\mathbf{V}_{2\to\mu} \in R^{(d+1)\times d}.$$

Then, we can compute a vector $\mathbf{k} \in R^{d+1}$ such that $\mathbf{k}^T\mathbf{C}_{1,2\to\mu}^{(j)} = \mathbf{0}$. Since $\mathbf{V}_{2\to\mu}$ is invertible with overwhelming probability, hence $\mathbf{k}^T\mathbf{U}_1 = \mathbf{0}$.

Hence, given the encoding $\overline{\mathbf{A}}_1$ of a sample $(\overline{\mathbf{A}}_1, \overline{\mathbf{B}}_1)$, we assume $\overline{\mathbf{A}}_1 = \mathbf{T}_0\mathbf{A}_1\mathbf{T}_1$. Note that except with the public parameters of the asymmetric variant, for a sample $(\overline{\mathbf{A}}_1, \overline{\mathbf{B}}_1)$, one only publicly publishes $\overline{\mathbf{A}}_1$, remains $\overline{\mathbf{B}}_1$ secret.

Then, we write $\mathbf{C}_{1',2\to\mu}^{(j)}$ as follows:

$$\mathbf{C}_{1',2\to\mu}^{(j)} = \begin{pmatrix} \mathbf{u}^T\mathbf{A}_1 \\ \mathbf{u}_{(2,1)}^T \\ \vdots \\ \mathbf{u}_{(d+1,1)}^T \end{pmatrix} \left(\mathbf{v}_{1,2\to\mu}^{(j)} \cdots \mathbf{v}_{d,2\to\mu}^{(j)}\right) = \mathbf{U}_{1'}\mathbf{V}_{2\to\mu} \in R^{(d+1)\times d}.$$

Now, we can generate a linear combination for such that $\alpha\mathbf{u}^T\mathbf{A}_1 = \sum_{l=2}^{d+1}\alpha_l\mathbf{u}^T\mathbf{A}_{l,1}$.

Similarly, we can choose a new group $\mathbf{u}_{(1,1)}^T, \cdots, \mathbf{u}_{(d,1)}^T$ to generate another linear combination $\beta\mathbf{u}^T\mathbf{A}_1 = \sum_{l=1}^{d}\beta_l\mathbf{u}^T\mathbf{A}_{l,1}$.

By the Coron et al.'s result [18], the elements α, β are relatively prime, which happens with significant probability. So, we can obtain $\mathbf{u}^T\mathbf{A}_1 = \sum_{l=1}^{d+1}\lambda_l\mathbf{u}^T\mathbf{A}_{l,1}$. Namely, $\mathbf{A}_1 = \sum_{l=1}^{d+1}\lambda_l\mathbf{A}_{l,1}$.

However, since $\|\lambda_l\|, l \in [d+1]$ are not small, we cannot perform the second step in the Coron et al.'s attack [18]. Consequently, we can not remove the noise term generated by λ_l for our asymmetric construction.

Hence, the Coron et al.'s attack [18] cannot break the hardness assumption Ext-GCDH/Ext-GDDH in the asymmetric variant.

4. Subfield lattice attack [1]

To break the overstretched NTRU assumption, Albrecht et al. [1] presented a subfield lattice attack. Given an overstretched instantiation of NTRU $h = [f/g]_q$, Albrecht et al.'s attack consists of three steps: mapping the secret key from full field to a subfield, running lattice reduction in subfield to solve smaller lattice problem, and lifting the solution back to the full field.

Using the subfield lattice attack, Albrecht et al. provided a quantum polynomial-time [10] and classical subexponential attack [3,4] for the GGH13 construction [20], regardless of whether the public parameter contains encodings of zero. However, this attack can be completely thwarted by setting Gaussian sample parameter $\sigma \approx q^{1/4}$, that is, $f, g \leftarrow D_{\mathbb{Z}^n,\sigma}$.

Cryptanalysis of the asymmetric variant. Given the public parameters, we compute

$$\begin{aligned}\overline{c}_l &= [\overline{\mathbf{u}}^T \prod_{k=1}^{\mu} \overline{\mathbf{A}}_{l,k} \overline{\mathbf{v}}]_q \\ &= [\mathbf{u}^T \prod_{k=1}^{\mu} \mathbf{A}_{l,k} \mathbf{v}/z^{\mu}]_q \\ &= [c_l/z^{\mu}]_q,\end{aligned}$$

where $c_l = \mathbf{u}^T \prod_{k=1}^{\mu} \mathbf{A}_{l,k} \mathbf{v}$.

So, we can obtain an instantiation of NTRU: $\frac{\overline{c}_{l_1}}{\overline{c}_{l_2}} = \frac{c_{l_1}}{c_{l_2}}$. By the equation (1), we know $\|c_{l_i}\| = \|\mathbf{u}^T \prod_{k=1}^{\mu} \mathbf{A}_{l_i,k} \mathbf{v}\| > q^{1/2}, i \in [2]$. Stehlé and Steinfeld [36] proved that $\frac{c_{l_1}}{c_{l_2}}$ is statistically close to uniform when $\|\mathbf{u}^T \prod_{k=1}^{\mu} \mathbf{A}_{l_i,k} \mathbf{v}\| > q^{1/2}, i \in [2]$. Hence, using the subfield lattice attack [1], we can not obtain $c_{l_i}, i \in [2]$, and z^{μ}.

Thus, our asymmetric variant can resist the subfield lattice attack in [1].

Cryptanalysis of the symmetric variant. For the symmetric variant in Remark 4.1 (2), by the public parameters we can compute

$$\begin{aligned}\overline{c}_0 &= [\overline{\mathbf{u}}^T \overline{\mathbf{v}}]_q = [\mathbf{u}^T \mathbf{v}/z_1]_q = [c_0/z_1]_q \\ \overline{c}_l &= [\overline{\mathbf{u}}^T \overline{\mathbf{A}}_l \overline{\mathbf{v}}]_q = [\mathbf{u}^T \mathbf{A}_l \mathbf{v}/z_1 z]_q = [c_l/z_1 z]_q,\end{aligned}$$

where $c_0 = \mathbf{u}^T \mathbf{v}$, $c_l = \mathbf{u}^T \mathbf{A}_l \mathbf{v}$.

Similarly, we generate an instantiation of NTRU: $\frac{\overline{c}_{l_1}}{\overline{c}_{l_2}} = \frac{c_{l_1}}{c_{l_2}}$. According to the setting of parameters, we have $\|c_{l_i}\| = \|\mathbf{u}^T \mathbf{A}_{l_i} \mathbf{v}\| > q^{1/2}, i \in [2]$. As a result, $\frac{c_{l_1}}{c_{l_2}}$ is statistically close to uniform. Hence, the symmetric variant can immune the subfield lattice attack.

Remark 4.9 (1) The immune ring proposed by Albrecht et al. [1] can further enhance the security of the symmetric variant. However, this countermeasure may not be essential for the security of construction.

(2) We observe that our asymmetric variant seems to avoid the principal ideal lattice problem [19,10].

4.5 Impossible results

In this section, we prove that it is impossible to completely avoid zeroizing attack. All previous constructions based on noise have the zeroizing attack problem. Why does this happen? Here, we will prove that the zeroizing attack problem is inherent in a noise-based construction of multilinear maps, as long as the construction supports the application of MPKE.

Lemma 4.10 Suppose that e is an ideal noise-based κ-level construction of multilinear maps and supports the application of MPKE. Then, there exists a zeroizing attack problem in the construction e.

proof. Without loss of generality, assume that $d_i \in R, i \in [\kappa+1]$ are plaintext encodings and $r_i \in R, i \in [\kappa+1]$ are random noise terms. Using $d_i, r_i, i \in [\kappa+1]$, we generate 1-level encodings $u_i = e(d_i, \{r_i\})$. Since e supports the application of MPKE, we can obtain a κ-level encoding as follows:

$$\begin{aligned} u &= d_1 \prod_{i=2}^{\kappa+1} u_i - d_{\kappa+1} \prod_{i=1}^{\kappa} u_i \\ &= e(\prod_{i=1}^{\kappa+1} d_i, \{r_2, \cdots, r_{\kappa+1}\}) - e(\prod_{i=1}^{\kappa+1} d_i, \{r_1, \cdots, r_{\kappa}\}) \\ &= e(0, \{d_1, \cdots, d_{\kappa+1}; r_1, \cdots, r_{\kappa+1}\}), \end{aligned}$$

where the final equation follows the definition 2.4.

Namely, u is a κ-level encoding of plaintext "0". Furthermore, since e is a noise-based κ-level construction, the probability $u = 0$ is negligible. Thus, given e, an attacker can generate a top-level encoding of zero. ∎

According to Lemma 4.10, it is easy to see that it is impossible to construct a graded encoding scheme without zeroizing attack. Because the MPKE application itself contains the top-level encodings of zero. Consequently, the best possible multilinear map we can expect is to construct a graded encoding scheme, whose security is not compromised by the top-level encodings of zero.

5 Symmetric Commutative Variant

Since the encodings in the asymmetric variant above are not commutative, as a result, it cannot be applied to some applications with commutative encodings, such as witness encryption (WE) based on 3-exact cover. To support these applications, we present a new symmetric variant, which uses the matrix form of approximate GCD.

5.1 Construction

Setting the parameters. Let λ be the security parameter, κ the multilinearity level. For simplicity, concrete parameters are set as $n = O(\lambda)$, $\sigma = O(n)$, $\rho = 2^{\lambda}$, $q = O(\kappa^2 \lambda^{15\kappa+20} 2^{\lambda(\kappa+1)})$, $\beta = O(q^{1/4})$, $d = O(\lambda)$, $\tau = \lambda d^2$. Let c be a small positive constant. We note that σ, d, τ can set small constants.

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

(1) Choose a prime $q = O(\kappa^2 \lambda^{15\kappa+20} 2^{\lambda(\kappa+1)})$.

(2) Choose a monic irreducible polynomial $f(y) \in \mathbb{Z}[y]$ with degree d and $\|f(y)\| = c$. Let $R_{(y)} = R[y]/\langle f(y) \rangle$, $R_{(y,q)} = R_q[y]/\langle f(y) \rangle$.

(3) Choose an invertible element $z \leftarrow R_q$, and two randomly invertible matrices $\mathbf{T}, \mathbf{T}_1 \leftarrow R_q^{d \times d}$.

(4) Sample $\mathbf{S} \leftarrow R_{\sigma}^{d \times d}$.

(5) For $l \in [\tau]$,

Sample $a_l \leftarrow R_{(y,\rho)}$, and set $\mathbf{A}_l = Rot_{(y)}(a_l)$.

Sample $\mathbf{E}_l, \mathbf{R}_l \leftarrow R_{\sigma}^{d \times d}$.

Set $\mathbf{X}_l = [\mathbf{T}(\mathbf{A}_l + \mathbf{E}_l)\mathbf{T}^{-1}/z]_q$, $\mathbf{Y}_l = [z^{\kappa} \cdot \mathbf{T}(\mathbf{A}_l \mathbf{S} + \mathbf{R}_l)\mathbf{T}_1^{-1}]_q$.

(6) Sample $\mathbf{u}, \mathbf{v} \leftarrow R_\beta^d$, and set $\overline{\mathbf{u}}^T = \mathbf{u}^T\mathbf{T}^{-1}, \overline{\mathbf{v}} = \mathbf{T}_1\mathbf{v}$.

(7) Output the public parameters $par = \{q, (\mathbf{X}_l, \mathbf{Y}_l)_{l\in[\tau]}, \overline{\mathbf{u}}, \overline{\mathbf{v}}\}$.

Generating a 1-level encoding: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Enc}(par)$.

Given $r_l \leftarrow R_\sigma$, $l \in [\tau]$, generate

$$\mathbf{U} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{X}_l\big]_q, \mathbf{V} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{Y}_l\big]_q.$$

Adding encodings: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Add}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$.

Given two k-level encodings $(\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2)$, compute

$$\mathbf{U} = [\mathbf{U}_1 + \mathbf{U}_2]_q, \mathbf{V} = [\mathbf{V}_1 + \mathbf{V}_2]_q.$$

Multiplying encodings: $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Mul}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$.

Given a i-level encoding $(\mathbf{U}_1, \mathbf{V}_1)$ and a j-level encoding $(\mathbf{U}_2, \mathbf{V}_2)$, compute a $i + j$-level encoding $(\mathbf{U}, \mathbf{V})$ as follows

$$\mathbf{U} = [\mathbf{U}_1\mathbf{U}_2]_q, \mathbf{V} = [\mathbf{U}_1\mathbf{V}_2]_q.$$

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

Given a $(\kappa + 1)$-level encoding $(\mathbf{U}, \mathbf{V})$, we check whether $\|\overline{\mathbf{u}}^T\mathbf{V}\overline{\mathbf{v}}\|$ is short:

$$\mathrm{isZero}(par, (\mathbf{U}, \mathbf{V})) = \begin{cases} 1, & \text{if } \|\overline{\mathbf{u}}^T\mathbf{V}\overline{\mathbf{v}}\| < \lambda^6 q/2^\lambda; \\ 0, & \text{otherwise.} \end{cases}$$

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

Given a $(\kappa + 1)$-level encoding $(\mathbf{U}, \mathbf{V})$, we extract the $\eta = O(\lambda - 6\log\lambda)$ most-significant bits from each of the n coefficients of $\overline{\mathbf{u}}^T\mathbf{V}\overline{\mathbf{v}}$:

$$\mathrm{Ext}(par, (\mathbf{U}, \mathbf{V})) = \mathrm{msbs}_\eta(\overline{\mathbf{u}}^T\mathbf{V}\overline{\mathbf{v}}).$$

Remark 5.1 (1) We remain $f(y)$ secret to improve the security of the symmetric variant. In fact, currently we do not find any possible attack when $f(y)$ is public. For efficiency, we can set d to constant.

(2) In the variant, we set $\mathbf{A}_l = Rot_{(y)}(a_l)$ to enable commutative plaintexts. Namely, given any plaintexts $\mathbf{A}_i, \mathbf{A}_j$, we have $\mathbf{A}_i\mathbf{A}_j = \mathbf{A}_j\mathbf{A}_i = Rot_{(y)}(a_ia_j)$. To correctly extract the most significant bits, we set $\rho = 2^\lambda$ and sample $a_l \leftarrow R_{(y,\rho)}$.

5.2 Correctness

Correctness of our construction follows from the commutative plaintexts of encodings. In the following we give the brief proof of correctness.

Lemma 5.2 The algorithm $\mathrm{InstGen}(1^\lambda, 1^\kappa)$ runs in polynomial time.

Proof. Since each step in InstGen runs in polynomial time, the result is directly obtained. ∎

Lemma 5.3 The encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Enc}(par)$ is a 1-level encoding.

Proof. By $r_l \leftarrow R_\sigma$, $l \in [\tau]$, we have

$$\mathbf{U} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{X}_l\big]_q = \big[\mathbf{T}(\mathbf{A}^{'} + \mathbf{E}^{'})\mathbf{T}^{-1}/z\big]_q$$
$$\mathbf{V} = \big[\sum_{l=1}^{\tau} r_l \cdot \mathbf{Y}_l\big]_q = \big[z^{\kappa} \cdot \mathbf{T}(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{T}_1^{-1}\big]_q,$$

where $\mathbf{A}^{'} = \sum_{l=1}^{\tau} r_l \cdot \mathbf{A}_l = Rot_{(y)}(\sum_{l=1}^{\tau} r_l \cdot a_l)$, $\mathbf{E}^{'} = \sum_{l=1}^{\tau} r_l \cdot \mathbf{E}_l$, $\mathbf{R}^{'} = \sum_{l=1}^{\tau} r_l \cdot \mathbf{R}_l$. ∎

Lemma 5.4 Given two k-level encodings $(\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2)$, the encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Add}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$ is a k-level encoding.

Proof. Since $(\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2)$ are k-level encodings. That is, for $i = 1, 2$

$$\mathbf{U}_i = \big[\mathbf{T}(\mathbf{A}_i^{'} + \mathbf{E}_i^{'})\mathbf{T}^{-1}/z^k\big]_q,$$
$$\mathbf{V}_i = \big[z^{\kappa-k+1} \cdot \mathbf{T}(\mathbf{A}_i^{'}\mathbf{S} + \mathbf{R}_i^{'})\mathbf{T}_1^{-1}\big]_q.$$

Thus, we have

$$\mathbf{U}_1 + \mathbf{U}_2 = \big[\mathbf{T}(\mathbf{A}^{'} + \mathbf{E}^{'})\mathbf{T}^{-1}/z^k\big]_q$$
$$\mathbf{V}_1 + \mathbf{V}_2 = \big[z^{\kappa-k+1} \cdot \mathbf{T}(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{T}_1^{-1}\big]_q,$$

where $\mathbf{A}^{'} = \mathbf{A}_1^{'} + \mathbf{A}_2^{'}$, $\mathbf{E}^{'} = \mathbf{E}_1^{'} + \mathbf{E}_2^{'}$, and $\mathbf{R}^{'} = \mathbf{R}_1^{'} + \mathbf{R}_2^{'}$. ∎

Lemma 5.5 The encoding $(\mathbf{U}, \mathbf{V}) \leftarrow \mathrm{Mul}(par, (\mathbf{U}_1, \mathbf{V}_1), (\mathbf{U}_2, \mathbf{V}_2))$ is a $i+j$-level encoding.

Proof. Since a i-level encoding $(\mathbf{U}_1, \mathbf{V}_1)$ has the following form:

$$\mathbf{U}_1 = \big[\mathbf{T}(\mathbf{A}_1^{'} + \mathbf{E}_1^{'})\mathbf{T}^{-1}/z^i\big]_q,$$
$$\mathbf{V}_1 = \big[z^{\kappa-i+1} \cdot \mathbf{T}(\mathbf{A}_1^{'}\mathbf{S} + \mathbf{R}_1^{'})\mathbf{T}_1^{-1}\big]_q.$$

Similarly, the j-level encoding $(\mathbf{U}_1, \mathbf{V}_1)$ also has the above form.

So, we get

$$\mathbf{U}_1\mathbf{U}_2 = \big[\mathbf{T}(\mathbf{A}^{'} + \mathbf{E}^{'})\mathbf{T}^{-1}/z^{i+j}\big]_q,$$
$$\mathbf{U}_1\mathbf{V}_2 = \big[z^{\kappa-(i+j)+1} \cdot \mathbf{T}(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{T}_1^{-1}\big]_q.$$

where $\mathbf{A}^{'} = \mathbf{A}_1^{'}\mathbf{A}_2^{'}$, $\mathbf{E}^{'} = \mathbf{A}_1^{'}\mathbf{E}_2^{'} + \mathbf{E}_1^{'}(\mathbf{A}_2^{'} + \mathbf{E}_2^{'})$, $\mathbf{R}^{'} = (\mathbf{A}_1^{'} + \mathbf{E}_1^{'})\mathbf{R}_2^{'}$. ∎

Lemma 5.6 Given a $(\kappa+1)$-level encoding $(\mathbf{U}, \mathbf{V})$, the procedure isZero$(par, (\mathbf{U}, \mathbf{V}))$ can correctly determine whether $(\mathbf{U}, \mathbf{V})$ is an encoding of zero.

Proof. Since a $(\kappa+1)$-level encoding $(\mathbf{U}, \mathbf{V})$ has the following form:

$$\mathbf{U} = \big[\mathbf{T}(\mathbf{A}^{'} + \mathbf{E}^{'})\mathbf{T}^{-1}/z^{(\kappa+1)}\big]_q,$$
$$\mathbf{V} = \big[\mathbf{T}(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{T}_1^{-1}\big]_q.$$

So, $\overline{\mathbf{u}}^T\mathbf{V}\overline{\mathbf{v}} = \big[\mathbf{u}^T(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{v}\big]_q$.

Given an encoding $(\mathbf{U}_1, \mathbf{V}_1)$ returned by Enc, we have

$$\mathbf{U}_1 = \big[\mathbf{T}(\mathbf{A}_1^{'} + \mathbf{E}_1^{'})\mathbf{T}^{-1}/z\big]_q,$$
$$\mathbf{V}_1 = \big[z^{\kappa} \cdot \mathbf{T}(\mathbf{A}_1^{'}\mathbf{S} + \mathbf{R}_1^{'})\mathbf{T}_1^{-1}\big]_q.$$

So, we obtain

$$\|\mathbf{A}_1^{'}\| = \|\sum\nolimits_{l=1}^{\tau} r_l\mathbf{A}_l\| = O(\tau \cdot n \cdot \|r_l\| \cdot \|\mathbf{A}_l\|) = O(\lambda^{5.5}2^{\lambda}),$$
$$\|\mathbf{E}_1^{'}\| = \|\sum\nolimits_{l=1}^{\tau} r_l\mathbf{E}_l\| = O(\lambda^7),$$
$$\|\mathbf{R}_1^{'}\| = \|\sum\nolimits_{l=1}^{\tau} r_l\mathbf{R}_l\| = O(\lambda^7),$$

where $\|r_l\| = O(\sigma\sqrt{n}) = O(n^{1.5})$ for $r_l \leftarrow R_{\sigma}$.

Since the above construction supports κ multiplications, without loss of generality, we assume that $(\mathbf{U}_i, \mathbf{V}_i), i \in [\kappa + 1]$ is the encodings returned by Enc, and $(\mathbf{U}, \mathbf{V})$ is their product generated by using Mul.

For simplicity, we let $\mathbf{U} = \prod_{i=1}^{\kappa+1} \mathbf{U}_i$, $\mathbf{V} = \prod_{i=1}^{\kappa} \mathbf{U}_i \times \mathbf{V}_{\kappa+1}$.

Hence, we have

$$\begin{aligned}\mathbf{V} &= \prod\nolimits_{i=1}^{\kappa} \mathbf{U}_i \times \mathbf{V}_{\kappa+1} \\ &= \big[\mathbf{T} \cdot \prod\nolimits_{i=1}^{\kappa}(\mathbf{A}_i^{'} + \mathbf{E}_i^{'}) \cdot (\mathbf{A}_{\kappa+1}^{'}\mathbf{S} + \mathbf{R}_{\kappa+1}^{'}) \cdot \mathbf{T}_1^{-1}\big]_q \\ &= \big[\mathbf{T} \cdot (\prod\nolimits_{i=1}^{\kappa+1} \mathbf{A}_i^{'} \cdot \mathbf{S} + \mathbf{R}^{'}) \cdot \mathbf{T}_1^{-1}\big]_q \\ &= \big[\mathbf{T} \cdot (\mathbf{A}^{'} \cdot \mathbf{S} + \mathbf{R}^{'}) \cdot \mathbf{T}_1^{-1}\big]_q,\end{aligned}$$

where $\mathbf{R}^{'} = \mathbf{R}_1^{''} + \mathbf{R}_2^{''}$ such that

$$\mathbf{R}_1^{''} = \sum\nolimits_{j=1}^{\kappa} \prod\nolimits_{i=1}^{j-1}(\mathbf{A}_i^{'} + \mathbf{E}_i^{'}) \cdot \mathbf{E}_j^{'} \cdot \prod\nolimits_{i=j+1}^{\kappa}(\mathbf{A}_i^{'} + \mathbf{E}_i^{'}) \cdot \mathbf{A}_{\kappa+1}^{'}\mathbf{S},$$
$$\mathbf{R}_2^{''} = \prod\nolimits_{i=1}^{\kappa}(\mathbf{A}_i^{'} + \mathbf{E}_i^{'}) \cdot \mathbf{R}_{\kappa+1}^{'}.$$

According to the setting of parameters, it is not difficult to get

$$\|\mathbf{R}_1^{''}\| \approx \kappa \cdot O(\lambda^{5.5}2^{\lambda})^{\kappa} \cdot (nd)^{\kappa+1} \cdot O(\lambda^7) \cdot O(\lambda^{1.5}) = O(\kappa\lambda^{7.5\kappa+10.5}2^{\lambda\kappa}),$$
$$\|\mathbf{R}_2^{''}\| \approx \cdot O(\lambda^{5.5}2^{\lambda})^{\kappa} \cdot (nd)^{\kappa} \cdot O(\lambda^7)) = O(\lambda^{7.5\kappa+7}2^{\lambda\kappa}),$$
$$\|\mathbf{R}^{'}\| \approx \|\mathbf{R}_1^{''}\| + \|\mathbf{R}_2^{''}\| = O(\kappa\lambda^{7.5\kappa+10.5}2^{\lambda\kappa}),$$
$$\|\mathbf{A}^{'}\mathbf{S}\| \approx O(\lambda^{5.5}2^{\lambda})^{\kappa+1} \cdot (nd)^{\kappa+1} \cdot O(\lambda^{1.5}) = O(\lambda^{7.5\kappa+9}2^{\lambda(\kappa+1)}).$$

On one hand, if there exists some $\mathbf{A}_i^{'} = \mathbf{0}$ for $i \in [\kappa + 1]$, then we obtain

$$\begin{aligned}\|\overline{\mathbf{u}}^T \mathbf{V} \overline{\mathbf{v}}\| &= \|\left[\mathbf{u}^T(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{v}\right]_q\| \\ &= \|\left[\mathbf{u}^T \mathbf{R}^{'}\mathbf{v}\right]_q\| \\ &= \|\mathbf{u}^T \mathbf{R}^{'}\mathbf{v}\| \\ &= (nd)^2 \cdot n \cdot q^{1/2} \cdot O(\kappa\lambda^{7.5\kappa+10.5}2^{\lambda\kappa}) \\ &= O(\kappa\lambda^{7.5\kappa+15.5}2^{\lambda\kappa})q^{1/2} \\ &< \lambda^6 q/2^{\lambda}.\end{aligned}$$

On the other hand, if $\mathbf{A}^{'} \neq \mathbf{0}$, then with overwhelming probability,

$$\begin{aligned}\|\overline{\mathbf{u}}^T \mathbf{V} \overline{\mathbf{v}}\| &= \|\left[\mathbf{u}^T(\mathbf{A}^{'}\mathbf{S} + \mathbf{R}^{'})\mathbf{v}\right]_q\| \\ &\approx \|\mathbf{A}^{'}\mathbf{S}\mathbf{v}\| \\ &\approx (nd)^2 \cdot n \cdot q^{1/2} \cdot O(\lambda^{7.5\kappa+9}2^{\lambda(\kappa+1)}) \\ &\approx q.\end{aligned}$$

Thus, the zero-test procedure isZero$(par, (\mathbf{U}, \mathbf{V}))$ is correct. ∎

Lemma 5.7 If the plaintexts of two encodings $(\mathbf{U}_1, \mathbf{V}_1)$, $(\mathbf{U}_2, \mathbf{V}_2)$ are same, then $\mathrm{Ext}(par, (\mathbf{U}_1, \mathbf{V}_1)) = \mathrm{Ext}(par, (\mathbf{U}_2, \mathbf{V}_2))$.

Proof. By $\mathbf{V}_i = \left[\mathbf{T}(\mathbf{A}_i^{'}\mathbf{S} + \mathbf{R}_i^{'})\mathbf{T}_1^{-1}\right]_q, i \in [2]$, we have

$$\overline{\mathbf{u}}^T \mathbf{V}_i \overline{\mathbf{v}} = \left[\mathbf{u}^T(\mathbf{A}_i^{'}\mathbf{S} + \mathbf{R}_i^{'})\mathbf{v}\right]_q.$$

Since $\mathbf{A}_1^{'} = \mathbf{A}_2^{'}$, we obtain $\overline{\mathbf{u}}^T \mathbf{V}_1 \overline{\mathbf{v}} - \overline{\mathbf{u}}^T \mathbf{V}_2 \overline{\mathbf{v}} = \mathbf{u}^T \mathbf{E}_1 \mathbf{v} - \mathbf{u}^T \mathbf{E}_2 \mathbf{v}$.

Again since $\|\mathbf{u}^T \mathbf{R}_i^{'}\mathbf{v}\| < \lambda^6 q/2^{\lambda}, i \in [2]$, we have $\|\mathbf{u}^T \mathbf{R}_1^{'}\mathbf{v} - \mathbf{u}^T \mathbf{R}_2^{'}\mathbf{v}\| \leq 2\lambda^6 q/2^{\lambda}$.

Furthermore, when $\mathbf{A}_i^{'} \neq \mathbf{0}, i \in [2]$, by Lemma 5.6, $\|\mathbf{u}^T \mathbf{A}_i^{'}\mathbf{S}\mathbf{v}\| \approx q$ with overwhelming probability.

Hence, the $\eta = O(\lambda - 6\log\lambda)$ most-significant bits from each of the n coefficients of $\overline{\mathbf{u}}^T \mathbf{V}_i \overline{\mathbf{v}}$ is determined by the term $\mathbf{u}^T \mathbf{A}_i^{'}\mathbf{S}\mathbf{v}$. That is, $\mathrm{Ext}(par, (\mathbf{U}_1, \mathbf{V}_1)) = \mathrm{Ext}(par, (\mathbf{U}_2, \mathbf{V}_2))$ with overwhelming probability. ∎

5.3 Cryptanalysis

Since the plaintexts of encodings is commutative in the above symmetric variant, as a result, one can obtain the encoding of zero by computing $\mathbf{X}_i\mathbf{X}_j - \mathbf{X}_j\mathbf{X}_i$ from par. So, there also exists the zeroizing attack problem for this variant. However, our variant use new noise method to immunize the zeroizing attack. In the following work, we will try to rigorously prove that this variant is resistant to zeroizing attack.

6 Applications

In this section, we describe MPKE based on our asymmetric variant and branching-program obfuscation using vRLWE.

6.1 MPKE

Using the asymmetric variant, the MPKE protocol consists of three algorithms: Setup, Publish, and KeyGen.

Setup$(1^\lambda, 1^\mu)$:
Let $\kappa = \mu - 1$. Output the public parameter $(par) \leftarrow \mathrm{InstGen}(1^\lambda, 1^\kappa)$.

Publish(par, k):
For $k \in [\mu]$, the k-th party samples elements $d_{l,k} \leftarrow D_{\mathbb{Z}^n,\sigma_1}, l \in [\tau]$, publishes the public key $\mathbf{U}_k = \big[\sum_{l=1}^{\tau} d_{l,k}\overline{\mathbf{A}}_{l,k}\big]_q$, and remains $d_{l,k}$ as the secret key.

KeyGen$(par, k, d_{l,k}, \{\mathbf{U}_j\}_{j=1}^{\mu})$:
(1) Using the secret key $\{d_{l,k}, l \in [\tau]\}$, the k-th party computes $\mathbf{V}_k = \big[\sum_{l=1}^{\tau} d_{l,k}\overline{\mathbf{B}}_{l,k}\big]_q$.
(2) The k-th party computes and extracts the common secret key as follows:

$$sk_k = \mathrm{msbs}_\eta\left(\left[\overline{\mathbf{u}}^T\big(\prod_{j=1}^{k-1}\mathbf{U}_j \times \mathbf{V}_k \times \prod_{j=k+1}^{\mu}\mathbf{U}_j\big)\overline{\mathbf{v}}\right]_q\right).$$

Correctness. The following lemma shows the correctness of MPKE.

Lemma 6.1 Suppose that $sk_k, k \in [\mu]$ are generated by the above MPKE protocol. Then all $sk_k, k \in [\mu]$ are equal with overwhelming probability.

Proof. Let $\mathbf{V}_{1\to\mu}^{(k)} = \big[\prod_{j=1}^{k-1}\mathbf{U}_j \times \mathbf{V}_k \times \prod_{j=k+1}^{\mu}\mathbf{U}_j\big]_q$.
According to the asymmetric variant, we have

$$\begin{aligned}\overline{\mathbf{u}}^T\mathbf{V}_{1\to\mu}^{(k)}\overline{\mathbf{v}} &= \big[\overline{\mathbf{u}}^T\big(\prod_{j=1}^{k-1}\mathbf{U}_j \times \mathbf{V}_k \times \prod_{j=k+1}^{\mu}\mathbf{U}_j\big)\overline{\mathbf{v}}\big]_q \\ &= \big[\mathbf{u}^T\big(\prod_{j=1}^{k-1}\mathbf{C}_j \times (\mathbf{C}_k \cdot s + \mathbf{D}_k) \times \prod_{j=k+1}^{\mu}\mathbf{C}_j\big)\mathbf{v}\big]_q \\ &= \big[\mathbf{u}^T\big(\prod_{j=1}^{\mu}\mathbf{C}_j \cdot s\big)\mathbf{v} + \mathbf{u}^T\mathbf{D}_{1\to\mu}^{(k)}\mathbf{v}\big]_q,\end{aligned}$$

where $\mathbf{C}_j = \sum_{l=1}^{\tau} d_{l,j}\mathbf{A}_{l,j}$, $\mathbf{D}_k = \sum_{l=1}^{\tau} d_{l,k}\mathbf{E}_{l,k}$, $\mathbf{D}_{1\to\mu}^{(k)} = \prod_{j=1}^{k-1}\mathbf{C}_j \times \mathbf{D}_k \times \prod_{j=k+1}^{\mu}\mathbf{C}_j$.

By Lemma 4.6, we have $\|\mathbf{u}^T\mathbf{D}_{1\to\mu}^{(k)}\mathbf{v}\| < q^{7/8}$. By $\|s\| \approx q$, we get $\|\mathbf{u}^T\big(\prod_{j=1}^{\mu}\mathbf{C}_j \cdot s\big)\mathbf{v}\| \approx q$.

Furthermore, all plaintexts in $\mathbf{V}_{1\to\mu}^{(k)}, k \in [\mu]$ are $\prod_{j=1}^{\mu} \mathbf{C}_j$.

Hence, by Lemma 4.7, the result follows. ∎

Security. Unfortunately, the security of our scheme only depends on new hardness assumption Ext-GCDH/Ext-GDDH, and cannot be reduced to any classical hardness assumption. However, at present the attacks that we know against our scheme do not seem to apply to our MPKE protocol.

Remark 6.2 When constructing MPKE based on the symmetric variant, we first require to determine the computing order of the encodings $\{\mathbf{U}_k, k \in [\mu]\}$, then generate the common secret key using the above same method.

6.2 BP Obfuscation

Our BP obfuscation construction is a slight variant of the GGH13-based BP in [20] using a variant of vRLWE. Namely, in the vRLWE, we sample the secret key $\mathbf{S} \leftarrow R_q^{w\times w}$. For completeness, we concretely describe our construction in the following.

A branching program consists of a sequence of steps, where each step is defined by a pair of permutations. In each step, we choose one of the permutations according to one input bit. We then multiply all these permutations chosen in all steps, and output 1 if the resulting permutation is the identity.

Consider a dual-input branching program BP of width w and length κ over l-bit inputs:

$$\mathrm{BP} = (\mathrm{inp}_1(k), \mathrm{inp}_2(k), \{\mathbf{A}_{k,b_1,b_2}\}_{b_1,b_2\in\{0,1\}})_{k=1}^{\kappa},$$

where the $\mathbf{A}_{k,b_1,b_2}$'s are permutation matrices in $\{0,1\}^{w\times w}$, and $\mathrm{inp}_1(k), \mathrm{inp}_2(k) \in [l]$ are the input bit position examined in step k.

Without loss of generality, we assume that:

(1) Every step of BP examines two different input bits. Namely, for all $k \in [\kappa]$, we have $\mathrm{inp}_1(k) \neq \mathrm{inp}_2(k)$.

(2) Every pair of different input bits are examined in some step of BP. That is, for every pair $j_1, j_2 \in [l]$ such that $j_1 \neq j_2$ there exists a step $k \in [\kappa]$ such that $(\mathrm{inp}_1(k), \mathrm{inp}_2(k)) = (j_1, j_2)$.

(3) Every input bit is examined by BP exactly l' times. That is, for input bit position $j \in [l]$, if $\mathrm{ind}(j)$ denotes the set of steps that examine the j-th input bit:

$$\mathrm{ind}(j) = \{k \in [\kappa] : \mathrm{inp}_1(k) = j\} \cup \{k \in [\kappa] : \mathrm{inp}_2(k) = j\},$$

then we have $|\mathrm{ind}(j)| = l'$.

Now, we obfuscate BP as follows:

Step 1: Dummy branch. We introduce a "dummy branching program":

$$\mathrm{BP}' = (\mathrm{inp}_1(k), \mathrm{inp}_2(k), \{\mathbf{A}'_{k,b_1,b_2}\}_{b_1,b_2\in\{0,1\}})_{k=1}^{\kappa},$$

where every $\mathbf{A}'_{k,b_1,b_2} = \mathbf{I}$ is the identity matrix in $\{0,1\}^{w\times w}$.

Step 2: Bundle scalars. We first sample random scalars $\{\beta_{k,b}, \beta'_{k,b} \leftarrow R_\sigma : k \in [\kappa], b \in \{0,1\}\}$ such that

$$\alpha_{j,b} = \prod\nolimits_{k\in \mathrm{ind}(j)} \beta_{k,b} = \prod\nolimits_{k\in \mathrm{ind}(j)} \beta'_{k,b}.$$

Then, we bundle scalars to generate

$$\begin{aligned}\overline{\mathbf{A}}_{k,b_1,b_2} &= \beta_{k,b_1}\beta_{k,b_2}\mathbf{A}_{k,b_1,b_2},\\ \overline{\mathbf{A}}'_{k,b_1,b_2} &= \beta'_{k,b_1}\beta'_{k,b_2}\mathbf{A}'_{k,b_1,b_2}\end{aligned}$$

Step 3: Kilian randomization on the plaintexts. We choose random unimodular matrices $\mathbf{P}_k, \mathbf{P}'_k, k = 0, \cdots, \kappa$ such that the norms of $\mathbf{P}_k, \mathbf{P}'_k$ and their inverse matrices are small, and for $k \in [\kappa]$ set

$$\begin{aligned}\widetilde{\mathbf{A}}_{k,b_1,b_2} &= \mathbf{P}_{k-1}^{-1}\overline{\mathbf{A}}_{k,b_1,b_2}\mathbf{P}_k,\\ \widetilde{\mathbf{A}}'_{k,b_1,b_2} &= \mathbf{P}'^{-1}_{k-1}\overline{\mathbf{A}}'_{k,b_1,b_2}\mathbf{P}'_k.\end{aligned}$$

Step 4: Encoding using our vRLWE. We sample matrices $\mathbf{S} \leftarrow R_q^{w\times w}$, and $\mathbf{E}_{\kappa,b_1,b_2}, \mathbf{E}'_{\kappa,b_1,b_2} \leftarrow R_\sigma^{w\times w}$, and set

$$\begin{aligned}\widetilde{\mathbf{A}}_{\kappa,b_1,b_2} &= \widetilde{\mathbf{A}}_{\kappa,b_1,b_2}\mathbf{P}_\kappa^{-1}\mathbf{S} + \mathbf{E}_{\kappa,b_1,b_2} = \mathbf{P}_{\kappa-1}^{-1}\overline{\mathbf{A}}_{\kappa,b_1,b_2}\mathbf{S} + \mathbf{E}_{\kappa,b_1,b_2},\\ \widetilde{\mathbf{A}}'_{\kappa,b_1,b_2} &= \widetilde{\mathbf{A}}'_{\kappa,b_1,b_2}\mathbf{P}'^{-1}_\kappa\mathbf{S} + \mathbf{E}'_{\kappa,b_1,b_2} = \mathbf{P}_{\kappa-1}^{-1}\overline{\mathbf{A}}'_{\kappa,b_1,b_2}\mathbf{S} + \mathbf{E}_{\kappa,b_1,b_2},\end{aligned}$$

where we override the notations $\widetilde{\mathbf{A}}_{\kappa,b_1,b_2}, \widetilde{\mathbf{A}}'_{\kappa,b_1,b_2}$.

Step 5: Extend matrices. Let $d = w + s$ with $s \geq 2$.

For $k \in [\kappa - 1]$, we extend $w \times w$-dimensional matrices into $d \times d$-dimensional matrices

$$\begin{aligned}\widehat{\mathbf{A}}_{k,b_1,b_2} &= \begin{pmatrix}\widetilde{\mathbf{A}}_{k,b_1,b_2} & 0\\ 0 & \mathbf{R}_{k,b_1,b_2}\end{pmatrix},\\ \widehat{\mathbf{A}}'_{k,b_1,b_2} &= \begin{pmatrix}\widetilde{\mathbf{A}}'_{k,b_1,b_2} & 0\\ 0 & \mathbf{R}'_{k,b_1,b_2}\end{pmatrix},\end{aligned}$$

where $\mathbf{R}_{k,b_1,b_2}, \mathbf{R}'_{k,b_1,b_2} \leftarrow R_\sigma^{s\times s}$.

For $k = \kappa$, we modify the above extension into the following form:

$$\begin{aligned}\widehat{\mathbf{A}}_{\kappa,b_1,b_2} &= \begin{pmatrix}\widetilde{\mathbf{A}}_{\kappa,b_1,b_2} & \mathbf{R}^{(1,2)}_{\kappa,b_1,b_2}\\ \mathbf{R}^{(2,1)}_{\kappa,b_1,b_2} & \mathbf{R}_{\kappa,b_1,b_2}\end{pmatrix},\\ \widehat{\mathbf{A}}'_{\kappa,b_1,b_2} &= \begin{pmatrix}\widetilde{\mathbf{A}}'_{\kappa,b_1,b_2} & \mathbf{R}'^{(1,2)}_{\kappa,b_1,b_2}\\ \mathbf{R}'^{(2,1)}_{\kappa,b_1,b_2} & \mathbf{R}'_{\kappa,b_1,b_2}\end{pmatrix},\end{aligned}$$

where $\mathbf{R}^{(1,2)}_{\kappa,b_1,b_2}, \mathbf{R}'^{(1,2)}_{\kappa,b_1,b_2} \leftarrow R_\sigma^{w\times s}$, $\mathbf{R}^{(2,1)}_{\kappa,b_1,b_2}, \mathbf{R}'^{(2,1)}_{\kappa,b_1,b_2} \leftarrow R_\sigma^{s\times w}$.

Step 6: Kilian randomization on the encodings. We choose random invertible matrices $\mathbf{T}_k, \mathbf{T}'_k \in R_q^{d\times d}, k = 0, \cdots, \kappa$, and for $k \in [\kappa]$ set

$$\begin{aligned}
\widehat{\mathbf{A}_{k,b_1,b_2}} &= \mathbf{T}_{k-1}^{-1}\widehat{\mathbf{A}}_{k,b_1,b_2}\mathbf{T}_k,\\
\widehat{\mathbf{A}'_{k,b_1,b_2}} &= \mathbf{T}'^{-1}_{k-1}\widehat{\mathbf{A}}'_{k,b_1,b_2}\mathbf{T}'_k,\\
\widehat{\mathbf{u}}^T &= \mathbf{u}^T\mathbf{T}_0, \widehat{\mathbf{v}} = \mathbf{T}_\kappa\mathbf{v},\\
\widehat{\mathbf{u}'}^T &= \mathbf{u}'^T\mathbf{T}'_0, \widehat{\mathbf{v}'} = \mathbf{T}'_\kappa\mathbf{v}',
\end{aligned}$$

where $\mathbf{u}, \mathbf{v} \leftarrow R_\sigma^d$, $\mathbf{u}', \mathbf{v}' \leftarrow R_\sigma^d$ such that $u_j = u'_j, v_j = v'_j, j \in [w]$.

Step 7: Straddling sets. We can further apply the level structure by using the straddling sets defined in [6]. Because this level structure does not affect the result of each honest evaluation. For simplicity, in the following we do not concretely give the straddling set and still use the above notations.

Step 8: Output the obfuscation of BP. The obfuscation $\widehat{\mathrm{BP}}$ consists of the following matrices and vectors:

$$\begin{aligned}
&\left(\left\{\widehat{\mathbf{A}_{k,b_1,b_2}}, k \in [\kappa], b_1, b_2 \in \{0,1\}\right\}, \widehat{\mathbf{u}}^T, \widehat{\mathbf{v}}\right),\\
&\left(\left\{\widehat{\mathbf{A}'_{k,b_1,b_2}}, k \in [\kappa], b_1, b_2 \in \{0,1\}\right\}, \widehat{\mathbf{u}'}^T, \widehat{\mathbf{v}'}\right).
\end{aligned}$$

Evaluation. Given the obfuscation $\widehat{\mathrm{BP}}$ and an arbitrary input $\mathbf{x} \in \{0,1\}^l$, we compute an honest evaluation as follows:

$$\begin{aligned}
y &= \widehat{\mathbf{u}}^T \cdot \prod_{k=1}^{\kappa} \widehat{\mathbf{A}_{k,x_{inp_1(k)},x_{inp_2(k)}}} \cdot \widehat{\mathbf{v}}\\
&= \mathbf{u}^T \cdot \prod_{k=1}^{\kappa} \widehat{\mathbf{A}}_{k,x_{inp_1(k)},x_{inp_2(k)}} \cdot \mathbf{v}\\
&= \mathbf{u}^T \cdot \begin{pmatrix} \alpha \cdot \prod_{k=1}^{\kappa} \mathbf{A}_{k,x_{inp_1(k)},x_{inp_2(k)}}\mathbf{S} + \mathbf{E}^{(1,1)} & \mathbf{E}^{(1,2)}\\ \mathbf{E}^{(2,1)} & \mathbf{E}^{(2,2)} \end{pmatrix} \cdot \mathbf{v},
\end{aligned}$$

$$\begin{aligned}
y' &= \widehat{\mathbf{u}'}^T \cdot \prod_{k=1}^{\kappa} \widehat{\mathbf{A}'_{k,x_{inp_1(k)},x_{inp_2(k)}}} \cdot \widehat{\mathbf{v}'}\\
&= \mathbf{u}'^T \cdot \prod_{k=1}^{\kappa} \widehat{\mathbf{A}}'_{k,x_{inp_1(k)},x_{inp_2(k)}} \cdot \mathbf{v}'\\
&= \mathbf{u}'^T \cdot \begin{pmatrix} \alpha \cdot \prod_{k=1}^{\kappa} \mathbf{A}'_{k,x_{inp_1(k)},x_{inp_2(k)}}\mathbf{S} + \mathbf{E}'^{(1,1)} & \mathbf{E}'^{(1,2)}\\ \mathbf{E}'^{(2,1)} & \mathbf{E}'^{(2,2)} \end{pmatrix} \cdot \mathbf{v}'\\
&= \mathbf{u}'^T \cdot \begin{pmatrix} \alpha \cdot \mathbf{S} + \mathbf{E}'^{(1,1)} & \mathbf{E}'^{(1,2)}\\ \mathbf{E}'^{(2,1)} & \mathbf{E}'^{(2,2)} \end{pmatrix} \cdot \mathbf{v}',
\end{aligned}$$

where $\alpha = \prod_{j=1}^{l} \alpha_{j,x_j}$, the norms of $\mathbf{E}^{(i,j)}, \mathbf{E}'^{(i,j)}$ all are small.

Now, if $\prod_{k=1}^{\kappa} \mathbf{A}_{k,x_{inp_1(k)},x_{inp_2(k)}} = \mathbf{I}$, then it is easy to verify that $\|y - y'\| < q^{7/8}$ and $\widehat{\mathrm{BP}}(\mathbf{x}) = 1$. Otherwise, $\widehat{\mathrm{BP}}(\mathbf{x}) = 0$.

Security. Similarly, the security of this obfuscation only relies upon new hardness assumption Ext-GCDH/Ext-GDDH, and cannot be reduced to any classical hardness assumption. However, currently the attacks that we know against our construction do not seem to apply to this BP obfuscation.

References

1. M. Albrecht, S. Bai, and L. Ducas. A subfield lattice attack on overstretched NTRU assumptions Cryptanalysis of some FHE and Graded Encoding Schemes. CRYPTO 2016, LNCS 9814, pp.153-178. http://eprint.iacr.org/2016/127.
2. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular secure encryption based on hard learning problems. CRYPTO 2009, LNCS 5677, pp. 595-618.
3. J. Biasse and C. Fieker. Subexponential class group and unit group computation in large degree number fields. LMS Journal of Computation and Mathematics 17(A), pp.385-403, 2014.
4. J. Biasse. Subexponential time relations in the class group of large degree number fields. Advances in Mathematics of Communications, 8(4):407-425, 2014.
5. Z. Brakerskiy, C. Gentry, S. Halevi, T. Lepoint, A. Sahai, M. Tibouchi. Cryptanalysis of the Quadratic Zero-Testing of GGH. http://eprint.iacr.org/2015/845.
6. B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai. Protecting obfuscation against algebraic attacks. EUROCRYPT 2014, LNCS 8441, pp. 221-238.
7. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. Contemporary Mathematics, 324:71-90, 2003.
8. D. Boneh, D. J. Wu, and J. Zimmerman. Immunizing multilinear maps against zeroizing attacks. http://eprint.iacr.org/2014/930.
9. D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. CRYPTO 2014, LNCS 8616, pp. 480-499.
10. R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings, EUROCRYPT 2016, LNCS 9666, pp. 559-585.
11. J. S. Coron, C. Gentry, S. Halevi, T. Lepoint, H. K. Maji, E. Miles, M. Raykova, A. Sahai, M. Tibouchi. Zeroizing Without Low-Level Zeroes New MMAP Attacks and Their Limitations. http://eprint.iacr.org/2015/596.
12. 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.
13. J. H. Cheon, C. Lee. Cryptanalysis of the multilinear map on the ideal lattices. http://eprint.iacr.org/2015/461.
14. J. H. Cheon, C. Lee, H. Ryu. Cryptanalysis of the New CLT Multilinear Maps. http://eprint.iacr.org/2015/934.
15. J. S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. CRYPTO 2013, LNCS 8042, pp. 476-493.
16. 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.
17. J. S. Coron, T. Lepoint, and M. Tibouchi. New Multilinear Maps over the Integers. http://eprint.iacr.org/2015/162.
18. J. S. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi. Cryptanalysis of GGH15 Multilinear Maps. CRYPTO 2016, LNCS 9815, pp. 607-628.

19. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. EUROCRYPT 2013, LNCS 7881, pp. 1-17.
20. 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.
21. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps, CRYPTO (2) 2013, LNCS 8043, 479-499.
22. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. http://eprint.iacr.org/2014/666.
23. C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. TCC 2015, Part II, LNCS 9015, pp. 498-527.
24. C. Gentry, S. Halevi, H. K. Majiy, A. Sahaiz. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. http://eprint.iacr.org/2014/929.
25. O. Goldreich. Computational Complexity: a Conceptual Perspective. Cambridge University Press, New York, NY, USA, 1 edition, 2008.
26. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. STOC 2013, pp. 467-476.
27. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC 2008, pp. 197-206.
28. Shai Halevi. The state of cryptographic multilinear maps, 2015. Invited Talk of CRYPTO 2015.
29. Shai Halevi. Graded Encoding, Variations on a Scheme, http://eprint.iacr.org/2015/866.
30. Yupu Hu and Huiwen Jia. Cryptanalysis of GGH Map. http://eprint.iacr.org/2015/301.
31. A. Langlois, D. Stehlé, and R. Steinfeld, GGHLite: More Efficient Multilinear Maps from Ideal Lattices, EUROCRYPT 2014, LNCS 8441, 2014, pp. 239-256.
32. B. Minaud and P. Fouque. Cryptanalysis of the New Multilinear Map Over the integers, http://eprint.iacr.org/2015/941.
33. D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures, SIAM Journal on Computing, 37(1):267C302, 2007.
34. D. Micciancio and O. Regev. Lattice-based cryptography. Post Quantum Cryptography, pp. 147-191. Springer, February 2009.
35. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of ACM, 56(6), pp.1-40, 2009.
36. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. EUROCRYPT 2011, LNCS 6632, pp. 27-47.
37. J. Zimmerman. How to obfuscate programs directly. EUROCRYPT 2015, Part II, LNCS 9057, pp. 439-467.