

Multilinear Map via Scale-Invariant FHE: Enhancing Security and Efficiency

Jinsu Kim¹ Sungwook Kim¹ Jae Hong Seo^{2*}

¹ Frontier CS Lab., Software R&D Center
Samsung Electronics, Korea
{jinsu86.kim, sw14.kim}@samsung.com
² Myongji University, Korea
jaehongseo@mju.ac.kr

Abstract. Cryptographic multilinear map is a useful tool for constructing numerous secure protocols and Graded Encoding System (GES) is an *approximate* concept of multilinear map. In multilinear map context, there are several important issues, mainly about security and efficiency. All early stage candidate multilinear maps are recently broken by so-called zeroizing attack, so that it is highly required to develop reliable mechanisms to prevent zeroizing attacks. Moreover, the encoding size in all candidate multilinear maps grows quadratically in terms of multilinearity parameter κ and it makes them less attractive for applications requiring large κ .

In this paper, we propose a new integer-based multilinear map that has several advantages over previous schemes. In terms of security, we expect that our construction is resistant to the zeroizing attack. In terms of efficiency, the bit-size of an encoding grows sublinearly with κ , more precisely $O((\log_2 \kappa)^2)$.

To this end, we essentially utilize a technique of the multiplication procedure in *scale-invariant* fully homomorphic encryption (FHE), which enables to achieve sublinear complexity in terms of multilinearity and at the same time security against the zeroizing attacks (EUROCRYPT 2015, IACR-Eprint 2015/934, IACR-Eprint 2015/941), which totally broke Coron, Lepoint, and Tibouchi’s integer-based construction (CRYPTO 2013, CRYPTO2015). We find that the technique of scale-invariant FHE is not very well harmonized with previous approaches of making GES from (non-scale-invariant) FHE. Therefore, we first devise a new approach for approximate multilinear maps, called *Ring Encoding System (RES)*, and prove that a multilinear map built via RES is generically secure. Next, we propose a new efficient scale-invariant FHE with special properties, and then construct a candidate RES based on a newly proposed scale-invariant FHE.

It is worth noting that, contrary to the CLT multilinear map (CRYPTO 2015), multiplication procedure in our construction does not add hidden constants generated by ladders of zero encodings, but mixes randoms in encodings in non-linear ways without using ladders of zero encodings. This feature is obtained by using the scale-invariant FHE and essential to prevent the Cheon et al.’s zeroizing attack.

1 Introduction

Cryptographic multilinear map is a mathematical structure with cryptographic assumptions introduced by Boneh and Silverberg [6]. Garg, Gentry, and Halevi proposed the first *approximate* realization of the cryptographic multilinear map, called the *Graded Encoding Scheme (GES)* [25]. Since then, numerous secure protocols using multilinear map have been proposed. To name a few, the recent advances in indistinguishability obfuscation [26], broadcast encryption [7], multipartite Diffie-Hellman key exchange [25], attribute-based encryption [27, 28], witness encryption [30], programmable hashes [24], ID-based aggregate signatures [34], and many other secure protocols seriously depends on the invention of candidate multilinear maps [25, 19, 38, 31, 18].

Zeroizing Attack on Integer-based Multilinear Maps. To utilize randomized encodings for multilinear map, we need a way to check equality between two randomized encodings of the same message. The currently

* corresponding author

existing strategy is due to Garg, Gentry and Halevi [25]; A special parameter \mathbf{p}_{zt} is published. Once multiplied by \mathbf{p}_{zt} , all encodings of zeros become relatively smaller values than that of encodings of non-zeros. Then, due to homomorphic property of encodings, any encodings of the same message will have the same value in the most significant bits of the resulting values multiplied by \mathbf{p}_{zt} , so that one can test the equality of given encodings. Coron, Lepoint, and Tibouchi’s construction over the integers (CLT13) [19] also uses the same strategy. In CLT13 scheme, a maximum level, say κ , encoding is of the form C/z^κ , where $C := \text{CRT}_{p_i}(c_i)$, which means the result of Chinese remaindering of $c_i \pmod{p_i}$ for several secret primes p_i ’s. Let $\pi = \prod_{i=1}^n p_i$. Then, $C = \sum_{i=1, \dots, n} c_i u_i \pmod{\pi}$ for some u_i ’s (coefficients for Chinese remaindering). The zero-testing parameter \mathbf{p}_{zt} is of the form $\sum_{j=1, \dots, n} v_j \pmod{\pi}$, which is designed to satisfy that $u_i v_j = 0 \pmod{\pi}$ for $i \neq j$, so that $C/z^\kappa \cdot \mathbf{p}_{zt} \pmod{\pi}$ is congruent to $\sum_i c_i u_i v_i \pmod{\pi}$. Furthermore, if C is an encoding of 0, then v_i ’s are designed to satisfy $\sum_i c_i u_i v_i$ is sufficiently smaller than π , so that one can distinguish it from encodings of non-zero with overwhelming probability. The authors expected that one cannot extract useful information, in particular c_i , from $\sum_i c_i u_i v_i$ since it can be considered as a hidden subset sum problem with hidden coefficients $u_i v_i$ ’s and it can be intractable by sufficiently increasing n .

Recently, Cheon, Han, Lee, Ryu, and Stehlé totally broke the CLT13 scheme in the sense that all secrets can be recovered in polynomial time [12]. The basic goal of Cheon et al.’s attack is to find c_i from given a level-0 encoding $\text{CRT}_{p_i}(c_i)$. (All other secrets can be recovered from c_i ’s in polynomial time.) Using lower level encodings of 0, each of which is of the form $\text{CRT}_{p_i}(r_{ik})/z^t$, and arbitrary encodings $\text{CRT}_{p_i}(x_{ij})/z^{\kappa-t}$, one can construct a level- κ encoding of zero of the form $\text{CRT}_{p_i}(x_{ij} c_i r_{ik})/z^\kappa$. Once $\text{CRT}_{p_i}(x_{ij} c_i r_{ik})/z^\kappa$ is multiplied by \mathbf{p}_{zt} , one can obtain $\sum_i x_{ij} c_i r_{ik} u_i v_i$ as a small integer value since $\text{CRT}_{p_i}(x_{ij} c_i r_{ik})/z^\kappa$ is an encoding of zero. Let $r'_{ik} = r_{ik} u_i v_i$. One can consider the resulting value as a quadratic form $\sum_i x_{ij} c_i r'_{ik}$ over \mathbb{Z} with coefficients c_i ’s. The remaining of attack is to use the theory of basic linear algebra to recover c_i ’s from such the quadratic form over \mathbb{Z} (e.g., computing eigenvalues of diagonalizable matrix).

Very recently, a candidate fix of the CLT13 scheme [18] by the same authors, say the CLT15 scheme [18], is also broken by two independent works due to Cheon, Lee and Ryu [13] and Minaud and Fouque [40]. Their main idea is to neutralize the effect of ladders of zero encodings in the multiplication procedure, which was the main idea to prevent the Cheon et al.’s zeroizing attack in [18]. We will explain both ideas of the CLT15 scheme and of attacks [13, 40] later in Section 6.1. As pointed out by Coron et al. [17], it is important to make the adversary not to obtain from the multiplication process and the zero-testing process, a simple system of equations over the base ring that can be solved using linear algebraic techniques. Viewed in this light, the multiplication process in both the CLT13 and CLT15 schemes is quite simple and have many vulnerable points since each value in slots of Chinese remaindering behaves independently during the multiplication process. We find that the multiplication technique of leveled fully homomorphic encryption (FHE) schemes [9, 22, 20, 15] can be a good candidate for preventing the zeroizing attack, in particular, integer-based batch version *scale-invariant* FHE schemes [20, 15]. Contrary to the integer-based batch FHE scheme [11], which is the underlying scheme used in the CLT13 scheme, the multiplication process in [20, 15] has much more non-linear aspects. We will show that the multiplication process in the scale-invariant FHE scheme have a resistance to the zeroizing attack in Section 6.1.¹

Sublinear Complexity in terms of Multilinearity. The other important challenge in the context of multilinear maps is to allow large κ -level linearity with small encoding size. The bit-size of each encoding in all current candidate multilinear maps grows quadratically with level κ and overcoming the barrier of $O(\kappa^2)$ complexity in the bit-size of encoding is a challenging open problem; for example, the view of Ananth, Gupta, Isahi, and Sahai [3] is pessimistic about such the large level multilinear map with a small encoding size and Apon, Huang, Katz, and Malozemoff [4] also asked for the construction of such the multilinear map for implementing indistinguishability obfuscation.

What are the Obstacles to Sublinear Complexity? Let us consider the integer-based multilinear map constructions due to Coron, Lepoint, and Tibouchi [19, 18]. Note that recent ideal lattice based multilinear map constructions [25, 38] are broken by Hu and Jia [35]. The current design methodology for GES is using randomized homomorphic encodings with small errors, over which multiplication procedure increases the

¹ In fact, we propose a new scale-invariant FHE scheme and analyze on the proposed scheme.

size of errors additively. Linear increasing of error size is critical to the size of encodings. One has to set η the bit-size of coprime integers p_i to be $O(\kappa\rho)$, where ρ is the bit-size of errors and κ is the multilinearity parameter. To thwart lattice-based attacks, one has to set the bit-size of encodings $\omega(\eta^2 \log_2 \lambda)$, so that the encoding size should be quadratic in κ .

In the context of fully homomorphic encryptions, there are two solutions to resolve this problem of rapid increasing of errors. One of which is so-called bootstrapping technique, and the other is leveled FHE including scale-invariant FHE.

The bootstrapping is a technique for refreshing errors to be small. To the best of our knowledge, the complexity of all known techniques [44, 42] is exponential in the bit-size of message space, but we need exponentially large message space for cryptographic multilinear maps. For example, a kind of the subset membership problem requires the message space to be exponentially large, so that one cannot apply the brute-force attack.

As for the technique used in the previous leveled FHE schemes, it yields usually better performance than that used in non-leveled FHE schemes in terms of ciphertext size since a multiplication increases error size only polylogarithmically in the security parameter. One may think that it is trivial to achieve our goal if one applies the technique used in leveled FHE schemes for encodings of GES. The existing technique in all previous GES proposals, roughly speaking, uses the special element $1/z$ for sharp grading, so that a level- i encoding of message m is of the form $enc(m)/z^i \pmod{N}$ for some N , where $enc(\cdot)$ is an encryption function of the underlying homomorphic encryption scheme. If the multiplication function in the underlying homomorphic encryption scheme is simple (e.g., multiplication in \mathbb{Z} or in \mathbb{Z}_N), then the multiplication between two level-1 encodings $enc(m_0)/z$ and $enc(m_1)/z$ outputs $enc(m_0 \cdot m_1)/z^2$, which is a level-2 encoding of $m_0 \cdot m_1$ as desired. However, the multiplication function in leveled FHE is not simple, so that it would be difficult to handle the special element $1/z$ with the multiplication function in leveled FHE.² More precisely, to make $1/z^2$ from $1/z$, one needs to perform multiplication in some ring (e.g., \mathbb{Z}_N), but multiplication between two encryptions in leveled FHE (over the integers) is not a ring multiplication, so that it is unclear how to handle $enc(m_0)/z$ and $enc(m_1)/z$ for obtaining of the form $enc(m_0 \cdot m_1)/z^2$.

1.1 Our Contributions

We propose a new scale-invariant FHE scheme and then also propose a candidate multilinear map based on our FHE scheme. Our approach to achieve candidate multilinear map is quite different from the previous GES schemes. We introduce a new concept, called *Ring Encoding System (RES)*, which enables one to perform ring operations without the description of the underlying ring from RES. In contrast to GES, RES does not require sharp grading, which is a main difference between RES and GES. Then, the aforementioned problem about harmonizing the technique of leveled FHE with GES does not occur in using RES. In fact, the concept of RES is for self-bilinear map with unknown message space. There is a generic transformation from self-bilinear maps to multilinear maps [14] and a proposal of secure self-bilinear map using the indistinguishability obfuscation [45], which was used to construct unbounded multilinear maps via the indistinguishability obfuscation. The concept of RES can be interpreted as a self-bilinear map encoding. To show the reliability of using RES for cryptographic purpose, we prove that there is no *generic* polynomial time attacker breaking several cryptographic hard problems such as the Computational Diffie-Hellman and the Subgroup Membership problems with non-negligible probability, where the generic attacker means, informally, one who has only oracle access for all procedures of RES instead of real procedures of RES. We note that a realization of RES runs over arithmetic circuits which the underlying scale-invariant FHE can handle. Even though the definition of generic attackers covers limited adversaries, the security against generic attackers shows the reliability of new approach via RES for multilinear maps. Second, we propose a realization of RES by using a new integer-based efficient scale-invariant FHE with special property, which is of independent interest. More precisely, the proposed scale-invariant FHE does not reveal message space (e.g., a ring \mathbb{Z}_N) when the space is exponentially large. We note that unknown message space is an essential condition for RES. We will

² In fact, this is also problematic for the bootstrapping technique.

consider this issue in Section 3. Moreover, we analyze the security of the proposed RES, in particular, to consider *non-generic attacks* such as zeroizing attacks.

Multilinear Maps via Self Bilinear Maps, Revisited. We start from revisiting the approach by Cheon and Lee [14] who showed that (unbounded) multilinear maps can be constructed by *self bilinear maps*, which are bilinear maps whose domain group and range group are the same. There is no predetermined maximum level in unbounded multilinear maps. (Although unbounded property makes some decisional assumptions generically easy, several important assumptions still are survivable; e.g., Subgroup Membership assumption in composite order groups. We discuss this issue further later.) Unbounded multilinear map can be inductively defined by using a self bilinear map. Given a κ -linear map $e_\kappa : G \times \cdots \times G \rightarrow G$ and a self bilinear map \tilde{e} , a $(\kappa + 1)$ -linear map $e_{\kappa+1}$ is defined by $e_{\kappa+1}(g^{a_1}, \dots, g^{a_{\kappa+1}}) := \tilde{e}(e_{\kappa-1}(g^{a_1}, \dots, g^{a_\kappa}), g^{a_{\kappa+1}})$.

When Cheon and Lee proposed such a generic transformation, they also showed that (unbounded) multilinear map obtained from self bilinear map can not hold the computational Diffie-Hellman in the underlying group. Furthermore, the unbounded multilinear maps with the same domain and range can be considered as the *black box field*, so that there exists a sub-exponential time attack against the discrete logarithm problem due to Boneh and Lipton [5]. We note that, however, both the Cheon-Lee attack and the Boneh-Lipton attack essentially use the group order, so that both are not applicable when the underlying group of self bilinear map is unknown. In particular, Yamakawa, Yamada, Hanaoka, and Kunihiko already pointed out this fact and used it for constructing a self-bilinear map using the indistinguishability obfuscation [45]. So far there is no self bilinear map with unknown group order and without using indistinguishability obfuscation, and it's exactly our first goal in this paper.

Self Bilinear Map from (\mathbb{Z}_N) . We begin with observing that \mathbb{Z}_N itself is a \mathbb{Z}_N -module for square-free product of primes N ; that is, \mathbb{Z}_N is a additive cyclic group of order N , where any integer coprime to N is a generator of $(\mathbb{Z}_N, +)$. Using operations in \mathbb{Z}_N we can simply construct an efficiently computable non-degenerate self bilinear map as follows.

$$\begin{aligned} \hat{e} : \mathbb{Z}_N \times \mathbb{Z}_N &\rightarrow \mathbb{Z}_N \\ (\alpha \cdot \omega, \beta \cdot \omega) &\mapsto (\alpha \cdot \omega) \cdot (\beta \cdot \omega), \end{aligned}$$

where scalars α, β are chosen from $\mathbb{Z}_N \setminus \{0\}$, a base ω is chosen from $\mathbb{Z}_N \setminus \{0, 1\}$ with $\gcd(\omega, N) = 1$, and \cdot is a multiplication modulo N . \hat{e} is clearly an efficiently computable non-degenerate self bilinear map, where the underlying group is $(\mathbb{Z}_N, +)$ as a \mathbb{Z}_N -module; e.g., every element in \mathbb{Z}_N can be written of the form $\alpha \cdot \omega$ since ω is invertible in \mathbb{Z}_N . The non-degeneracy is straightforward. The bilinearity is achieved since $\hat{e}((\alpha + \alpha')\omega, \beta\omega) = (\alpha\beta\omega^2 + \alpha'\beta\omega^2) = \hat{e}(\alpha\omega, \beta\omega) + \hat{e}(\alpha'\omega, \beta\omega)$ and the other side can be similarly shown. Unfortunately, however, we cannot use the above example for cryptographic purpose since the group order N is necessary for public evaluation of \hat{e} , which is a multiplication in \mathbb{Z}_N , so that one can apply Cheon-Lee attack or directly compute ω^{-1} by using the Extended Euclidean Algorithm to solve the discrete logarithm problem in $(\mathbb{Z}_N, +)$. Here, our question is that

“Can we perform ring operations without knowing the description of ring explicitly (e.g., N in the above example)?”

Self Bilinear Map via Ring Encoding System. We introduce a new concept, called *Ring Encoding System*, which enables to perform all ring operations without knowing the underlying ring explicitly. That is, given RES construction, one can utilize it like a self-bilinear map as we discussed above, then applying the Cheon-Lee transformation we have a candidate multilinear map. Conceptually, RES is exactly like a combination of FHE and zero-testing parameter for equality test of randomized encodings. Here, an important requirement of FHE is that addition and multiplication over encrypted messages can be performed without the description of the message space.

To realize RES, we first propose a scale-invariant FHE scheme with unknown message space. Almost all (integer-based) scale-invariant FHE schemes [22, 20] use the information about \mathbb{Z}_N for multiplications, so that we cannot use such FHE schemes for RES. We find that a recent scale-invariant FHE proposal due to Cheon and Stehlé [15] uses N for encryption and does not use it for multiplication and addition, so that it can be used for RES. (In our RES, sampling algorithm yields only uniformly distributed encoding by using

additive homomorphic property from a set of encodings, like that of GES, so that the encryption algorithm of the underlying FHE is not necessarily used.) The heart of Cheon and Stehlé FHE is short parameters for small message space, but the public key size of Cheon and Stehlé FHE for exponentially large message space becomes very huge $O(\lambda^7)$. Our proposal has smaller public key size $O(n\lambda^4)$, where n is the number of slots in batch.

Next, we follow the known approach for constructing zero-testing parameter in [18] for zero-testing procedure. We note again that the security of our construction against the Cheon et al. attack is not mainly depending on the zero-testing procedure, but on the multiplication procedure of scale-invariant FHE. Let us present a high level intuition of the reason why we believe that the multiplication procedure in the scale-invariant FHE is resistant against the zeroizing attacks. In our scale-invariant FHE (like the other scale-invariant schemes), there is a special key for multiplication procedure, which is a vector \mathbf{z} of a kind of *fake* encryption in the sense that each component of \mathbb{Z} looks similar to encryptions. Then, the multiplication of given two encryptions a and b is as follows. First, compute a product of a and b over the integers. Second, bitwise decompose ab , that is, $c = ab \mapsto \mathbf{c} = (c_1, \dots, c_{2\gamma})$ for $c = \sum_i c_i 2^i$. Finally, compute an inner product between \mathbf{c} and \mathbf{z} modulo a zero encryption x_0 . We relegate the detailed reason why this process works correctly in Section 4.1. Here, we only focus on the security against the zeroizing attack. The values in a and b effect in each bit c_i in a non-linear way, so that those also effect in the resulting value in $\langle \mathbf{c}, \mathbf{z} \rangle \pmod{x_0}$ in a non-linear way. In particular, if we use a batch version encryption scheme, randomness in each slot of Chinese remaindering are diffused to the other slots. Therefore, we expect that one can hardly make a simple system of equations when the above multiplication procedure is used in the zero-testing procedure. We further analyze the security of our scheme against the zeroizing attacks in Section 6.1.

Finally, we obtain RES realization, which is not the ideal RES realization, but its noisy construction, so that only limited ring operations can be performed over it as long as errors are small enough. Nevertheless, the RES with noisy encoding is enough for large level multilinear maps since the encoding size is polylogarithmic in the maximum allowable multilinearity level κ , that is, $\tilde{O}((\log_2 \kappa)^2 \cdot \lambda^3)$.

1.2 Related Works

Since the introduction of multilinear map by Boneh and Silverberg [6], the construction of multilinear maps have received huge attention from crypto community. In 2013, Garg, Gentry, and Halevi proposed the first construction of the candidate of multilinear maps (GGH) based on ideal lattices [25]. Langlois, Stehlé, and Steinfeld presented GGHLite which improves the efficiency of the GGH [38]. Shortly after the GGH construction, Coron, Lepoint, and Tibouchi proposed a construction over the integers (CLT) [19].

Recently cryptanalysis have been presented for the above constructions. Hu and Jia presented attack for the GGH maps; their attack does not reveal users' secrets, however, it efficiently attacks multipartite key exchange and witness encryption. In [12], Cheon, Han, Lee, Ryu, and Stehlé totally broke the CLT using the so-called zeroizing attack, which exploits encodings of 0 and the zero-testing parameter; it recovers all secret parameters in polynomial time. After then there have been several trials to avoid the zeroizing attack [8, 29], however, Coron et al. extended the Cheon et al. attack and showed those fixes are not secure [17]. Coron, Lepoint, and Tibouchi presented the new CLT map in [18], which is a tentative fix of the CLT13 scheme [19]. However, a polynomial-time attack on the new CLT map has been proposed by Cheon, Lee, and Ryu [13], also independently by Minaud and Fouque [40].

Currently three candidates of multilinear maps seem to be secure against the zeroizing attack. Yamakawa, Yamada, Hanaoka, and Kunihiro proposed a weaker variant of self-bilinear map and constructed unbounded multilinear map from it [45]. Their construction is based on the assumption of the existence of indistinguishability obfuscation (iO) although the current candidate construction for iO uses a multilinear map as a building block. Albrecht, Farshim, Hofheinz, Larraia, and Paterson also proposed multilinear maps from iO [2]. Gentry, Gorbunov, and Halevi presented another multilinear maps from lattices using a directed acyclic graph [31].

1.3 Outline

In the next section, we give definitions for objects we deal with in this paper. In Section 3, we introduce a new concept called Ring Encoding System (RES) and then show how to construct multilinear maps by using RES. In particular, we prove the generic hardness of several cryptographic assumptions in the multilinear maps via RES in Section 3. In Section 4.1, we propose a new scale-invariant homomorphic encryption scheme and show that the message space is *provably hidden* from public key of the proposed scheme. In Section 5, our proposal for RES is presented and its security against non-generic attacks, in particular, zeroizing attacks, is analyzed in Section 6.

2 (Unbounded) Multilinear Maps, Self Bilinear Maps, and Assumptions

In this section, we review definitions of (unbounded) multilinear maps, self bilinear maps and the transformation between them due to Cheon and Lee [14]. Furthermore, to clarify what we can expect from the Cheon-Lee transformation, we investigate the generic easiness/hardness of cryptographic assumptions.

Notation. Through the paper, we use notation λ and $i = 0..a$ to denote the security parameter and $i \in \mathbb{Z} \cap [0, a]$, respectively.

Definition 1 ((Unbounded) Multilinear Maps) *We say an algorithm $\mathcal{G}_{\mathcal{UML}}$ is a multilinear group generator if it takes the security parameter λ as input and outputs $(N, \{G_i\}_{i=1..a}, \{e_{i,j}\}_{i,j=1..a})$ where G_i 's are cyclic groups of order N and $e_{i,j} : G_i \times G_j \rightarrow G_{i+j}$ satisfying the followings.³*

- (non-degeneracy) For generators $g_i \in G_i$ and $g_j \in G_j$, $e_{i,j}(g_i, g_j) = g_{i+j}$ is a generator of G_{i+j} .
- (bilinearity) For any $h_i \in G_i$, $h_j \in G_j$ and $a, b \in \mathbb{Z}_N$, $e_{i,j}(h_i^a, h_j^b) = e_{i,j}(h_i, h_j)^{ab}$.

From the family of bilinear maps $\{e_{i,j}\}$ defined above, for any κ one can easily make $(\kappa + 1)$ -level multilinear maps

$$e(g_1^{a_1}, \dots, g_1^{a_{\kappa+1}}) := e_{\kappa,1}(\dots e_{2,1}(e_{1,1}(g_1^{a_1}, g_1^{a_2}), g_1^{a_3}), \dots), g_1^{a_{\kappa+1}}) = g_{\kappa+1}^{a_1 \dots a_{\kappa+1}},$$

where $g_{\kappa+1} = e_{\kappa,1}(\dots e_{2,1}(e_{1,1}(g_1, g_1), g_1), \dots), g_1)$ is a generator of $G_{\kappa+1}$.

Cheon and Lee showed that unbounded multilinear maps can be constructed by *self bilinear maps*. Informally, self bilinear maps are bilinear maps with the same domain and range group. (Formal definition is given below.) In fact, such the conversion is essentially the same as the above conversion from the family of bilinear maps $\{e_{i,j}\}$ to a unbounded multilinear map; it is defined inductively. Given a κ -linear map $e_\kappa : G \times \dots \times G \rightarrow G$ and a self bilinear map \tilde{e} , a $(\kappa + 1)$ -linear map $e_{\kappa+1}$ is defined by $e_{\kappa+1}(g^{a_1}, \dots, g^{a_{\kappa+1}}) := \tilde{e}(e_\kappa(g^{a_1}, \dots, g^{a_\kappa}), g^{a_{\kappa+1}})$.

Definition 2 (Self Bilinear Maps) *We say an algorithm \mathcal{G}_{SBL} is a self bilinear group generator if it takes the security parameter λ as input and outputs (N, G, \tilde{e}) where G is a cyclic group of order N and $\tilde{e} : G \times G \rightarrow G$ satisfying the followings.*

- (non-degeneracy) For a generator $g \in G$, $\tilde{e}(g, g) = g_t$ is also a generator of G .
- (bilinearity) For a generator $g \in G$ and any $a, b \in \mathbb{Z}_N$, $\tilde{e}(g^a, g^b) = \tilde{e}(g, g)^{ab}$.

If the group order N is a public prime, that is, N is prime and known to the adversary, the unbounded multilinear maps with the same domain and range G can be considered as the *black box field* since both multiplication and addition between exponents of group elements in G are defined. Then, we cannot use such multilinear map obtained from self bilinear map for cryptographic purpose since there exists a sub-exponential time attack to the discrete logarithm problem due to Boneh and Lipton [5]. Furthermore, with public prime N , Cheon and Lee [14] presented a polynomial $O(\log p)$ time attack to the Computational

³ Although the output contains infinite families of groups and bilinear maps, it does not always implies that the output is infinite bit-string. We assume that such the families can be succinctly described with polynomial size bit-string.

Diffie-Hellman problem over the unbounded multilinear maps.⁴ In both attacks, it is crucial to know the group order N , which is prime and hence $\phi(N)$ is easily computable. Therefore, for cryptographic purpose, we are interested in only self bilinear maps with *unknown prime order group*, or *known-but-hard-to-factor composite order group*. In particular, we are interested in hiding group order, rather than using composite order, for reason of precaution; if the discrete logarithm of $\tilde{e}(g, g)$ to the base g is known, there is a polynomial attack for the computational Diffie-Hellman problem even when N is hard-to-factor.⁵ Note that all known candidate multilinear maps also hide their message spaces, which correspond to group order, over which multilinear map is defined.⁶

Regardless of hiding order of underlying group of self bilinear map, there are easy cryptographic problems. We sort out generically easy problem in the underlying group of self bilinear maps with an unknown group order.

Definition 3 (Problems) Let $\mathcal{G}_{SBL}(\lambda) \rightarrow (N, G, \tilde{e})$ for security parameter λ . For positive integer κ , the κ -MCDH/ κ -MDDH/ κ -DLIN problems are defined as follows: Let g be a generator of G and e_κ be a κ -linear map inductively defined via \tilde{e} as the above.

- κ -Multilinear CDH (MCDH) problem: given a tuple $(g, g^{a_1}, \dots, g^{a_{\kappa+1}}) \in G^{\kappa+2}$, find $e_\kappa(g, \dots, g)^{a_1 \cdots a_{\kappa+1}}$.
- κ -Multilinear DDH (MDDH) problem: distinguish between two distributions \mathcal{D}_c^{MDDH} and \mathcal{D}_r^{MDDH} , where $\mathcal{D}_r^{MDDH} = (g, g^{a_1}, \dots, g^{a_{\kappa+1}}, g^r)$ for $r \xleftarrow{\$} \mathbb{Z}_N$ and $\mathcal{D}_c^{MDDH} = (g, g^{a_1}, \dots, g^{a_{\kappa+1}}, e_\kappa(g, \dots, g)^{a_1 \cdots a_{\kappa+1}})$.
- κ -Decisional Linear (DLIN) problem: distinguish between two distributions \mathcal{D}_c^{DLIN} and \mathcal{D}_r^{DLIN} , where $\mathcal{D}_r^{DLIN} = (g, g_1, \dots, g_\kappa, g_1^{a_1}, \dots, g_\kappa^{a_\kappa}, g^r)$ for $r \xleftarrow{\$} \mathbb{Z}_N$ and $\mathcal{D}_c^{DLIN} = (g, g_1, \dots, g_\kappa, g_1^{a_1}, \dots, g_\kappa^{a_\kappa}, g^{\sum_{i=1 \dots \kappa} a_i})$.

To define the Subgroup Membership problem, we assume that the group order N is composite; that is, $N = pq$, where p and q are coprime and not necessarily primes. Let g_p be a generator of subgroup $G_p \subset G$ of order p .

- Subgroup Membership (SubM) problem: given g, g_p and g' , determine whether $g' \xleftarrow{\$} G$ or $g' \xleftarrow{\$} G_p$.

The κ -MCDH/ κ -MDDH/ κ -DLIN/Subgroup Membership assumptions state that the advantage of any polynomial-time adversary in attacking the corresponding problem is negligible in the security parameter.

Lemma 1 (Easy Problems) For any positive integer κ , there are polynomial time attacks to the κ -MDDH problem and the κ -DLIN problem defined over the unbounded multilinear maps via self bilinear maps, regardless of knowing the group order.

The proof of Lemma 1 is given in Appendix A.1.

As for the other problems, we prove generic hardness of them in a special form of groups in the next section. (It sounds paradoxical since the term ‘generic’ usually does not restrict the form of groups. Here, we use the term ‘generic’ differently from the previous ‘generic group model’. More precisely, it is a kind of *generic ring model* for the ring of integers modulus N for some N , where cryptographic hard problem is defined in $(\mathbb{Z}_N, +)$.⁷

In the context of bilinear maps and multilinear maps, the decisional linear problem in prime order groups and the subgroup membership problem in composite order groups are regarded as having similar features since the decisional linear problem can be considered as a kind of the subgroup membership problem in product groups of prime order groups. We once again stress that even though the decisional linear problem is generically easy in the self bilinear map context, the subgroup membership problem could be generically hard.

⁴ Let $\tilde{e}(g, g) = g^t$ for some (unknown) t . Given $g, g^a, g^c, \tilde{e}(\tilde{e}(g^a, g^b), g^{t^{\phi(N)-2}}) = \tilde{e}(g^{abt}, g^{t^{\phi(N)-2}}) = g^{abt^{\phi(N)}} = g^{ab}$, where ϕ is the Euler totient function. The last equality holds, because g^t is also a generator of G , so that t is coprime to N . Since $g^{t^{\phi(N)-2}}$ can be computed with $O(\log \phi(N))$ bilinear map operations, this attack runs in polynomial time.

⁵ Let $\tilde{e}(g, g) = g^t$ and t be given. Then, one can perform the Extended Euclidean Algorithm to compute $t^{-1} \bmod N$ in polynomial time in $\log N$. Then, given $g^a, g^b, \tilde{e}(g^a, g^b)^{t^{-1}} = g^{ab}$.

⁶ There is a technique of partially exposing the message space [18], but it is basically built on the multilinear maps with an unknown message space.

⁷ It does not mean that those problems are generically easy in generic group model.

3 Self Bilinear Map via Ring Encoding System

In this section, we introduce a new approach toward self bilinear maps with holding cryptographic assumptions. In particular, we prove the generic hardness of cryptographic assumptions such as MCDH and SubM in our model.

First, let us begin with an intuitive example for building self bilinear maps from any commutative ring with unity.

Example 1. [Transformation from Ring to Self Bilinear Map] *Given a commutative ring R with unity containing at least one non-unity invertible element ω ,⁸ define a self bilinear map \hat{e} as follows.*

$$\begin{aligned} \hat{e} : \quad R \times R &\rightarrow R \\ (\alpha \cdot \omega, \beta \cdot \omega) &\mapsto (\alpha \cdot \omega) \cdot (\beta \cdot \omega), \end{aligned}$$

where R is considered as a R -module with a basis ω and α, β are scalars chosen from R . One can easily check that \hat{e} is a non-degenerate self bilinear map defined over R -module R .

To establish general notion, we consider the transformation from not only \mathbb{Z}_N but also commutative rings. Nevertheless, one may assume $R = \mathbb{Z}_N$ through the paper since our realization is only for $R = \mathbb{Z}_N$.

Remark 1. The definition of cryptographic multilinear map (of course, bilinear maps, too) is usually defined over abelian groups. However, mathematical definition of multilinear map is defined over not only abelian groups (that is, \mathbb{Z}_N -modules for cyclic groups of order N), but also R -modules for any commutative ring R . In fact, multilinear maps over \mathbb{Z}_N -module are familiar in cryptography community since the standard discrete logarithm problem is usually defined over \mathbb{Z}_N modules; that is, given an element g of a cyclic group G of order N and $x \cdot g$ for scalar $x \in \mathbb{Z}_N$, finding x is hard, where \cdot is a scalar multiplication. The discrete logarithm problem over R -module can be defined by extending \mathbb{Z}_N -module to R -module; given an invertible element ω of R -module G and $\alpha \cdot \omega$ for $\alpha \in R$, finding α is hard, where \cdot is a scalar multiplication.

Although Example 1 shows an easy way to build non-degenerate self bilinear maps from rings, for the practical usage we require that \hat{e} should be efficiently computable. Then, one soon faces a dilemma; for evaluating \hat{e} , one may need the description of R , but by using the description of R , one may be able to efficiently solve hard problems. For example, if $R = \mathbb{Z}_N$, N is necessary for ring operations in \mathbb{Z}_N , but from public N , which is the order of additive group $(\mathbb{Z}_N, +)$, one can apply the Cheon-Lee attack to solve the CDH problem as in Section 2. Indeed, for $R = \mathbb{Z}_N$, public N is much more dangerous since one can perform the Extended Euclidean Algorithm to find ω^{-1} in polynomial time in the size of N .

To resolve this dilemma, we introduce a new concept, called *Ring Encoding System*, for performing ring operations without knowing the underlying ring explicitly, in the sense that the order of embedded group is unknown. Note that what we consider here is similar to *generic ring model* [1, 36] or *black box field* [5], but in contrast to RES both the generic ring model and the black box field make the underlying ring/field public.

3.1 Ring Encoding System

We introduce a new concept, called Ring Encoding System (RES).

Definition 4 (Ring Encoding System) *A Ring Encoding System consists of a commutative ring R with unity containing an invertible element that is not a multiplicative identity and a system of sets $\mathcal{S} = \{S^{(\alpha_0)} \subset \{0, 1\}^* : \forall \alpha_0 \in R\}$, with the following properties.*

1. $S^{(\alpha_0)} \cap S^{(\beta_0)} = \emptyset$ for distinct α_0 and β_0 in R .

⁸ The unity itself can be used as a basis of R -module, but we require another invertible element ω for the discrete logarithm assumption; if we use the unity 1_R instead of ω , the discrete logarithm problem, which requires given $\alpha \cdot 1_R$ finding α , will be trivial.

2. There is an associative binary operation ‘+’ and a self-inverse unary operation ‘-’ such that for every $\alpha_0, \beta_0 \in R$ and every $u \in S^{(\alpha_0)}$ and $v \in S^{(\beta_0)}$, it holds that $u + v \in S^{(\alpha_0 + \beta_0)}$ and $-u \in S^{(-\alpha_0)}$, where $\alpha_0 + \beta_0$ and $-\alpha_0$ are addition and negation in R .
3. There is an associative binary operation ‘ \times ’ such that for every $\alpha_0, \beta_0 \in R$ and $u \in S^{(\alpha_0)}$ and $v \in S^{(\beta_0)}$, it holds that $u \times v \in S^{(\alpha_0 \cdot \beta_0)}$, where $\alpha_0 \cdot \beta_0$ is multiplication in R .

For the sake of simplicity, we prefer to use $[\alpha]$ to denote some encoding of the ring element α , i.e., $[\alpha] \in S^{(\alpha)}$. Through the paper, ω denotes the special non-identity invertible element in R .

We can define self bilinear maps based on RES similarly to Example 1. To make RES hold cryptographic assumptions such as the discrete logarithm, we require that given $[\omega]$, no polynomial time algorithm can find $[\omega^{-1}]$ with non-negligible probability. Therefore, the *non-identity* requirement for ω is necessary.

Remark 2 (Ring Encoding System vs. Graded Encoding System). The definition of graded encoding system, which is another concept for approximate multilinear maps, has similar feature to ours; that is, both systems are for manipulating encodings and two operations $(+, \cdot)$ over them. Contrast to ours, graded encoding system uses sharp grading such that each encoding has a grade, multiplications increase its grade, and additions can be carried out only for encodings with the same grade. In the ring encoding system, there is no such grade for encodings.

RES Procedures, Complete Version. To manipulate ring encodings, we define several procedures.

Instance Generation. The randomized $\text{InstGen}(1^\lambda, R)$ takes as input the security parameter λ and a ring R as input and outputs $(\text{pp}, \mathbf{p}_{zt})$, where pp is a description of a RES as above, in particular, pp contains $[\omega]$, and \mathbf{p}_{zt} is a zero-test parameter. Note that pp does not contain R in an explicit form.

Ring Sampler. The randomized $\text{samp}(\text{pp})$ takes pp as input and outputs a $[\alpha]$ for a nearly uniform element $\alpha \in_R R$. Note that $[\alpha]$ does not need to be uniform in $S^{(\alpha)}$.

Addition, Negation and Multiplication. Given pp and two encodings $[\alpha]$ and $[\beta]$, we have $\text{add}(\text{pp}, [\alpha], [\beta]) \in S^{(\alpha + \beta)}$ and $\text{neg}(\text{pp}, [\alpha]) \in S^{(-\alpha)}$. Furthermore, we have $\text{mul}(\text{pp}, [\alpha], [\beta]) \rightarrow [\alpha \cdot \beta] \in S^{(\alpha \cdot \beta)}$. We write $[\alpha] + [\beta]$, $-[\alpha]$ and $[\alpha] \cdot [\beta]$ as shorthands for applying these procedures, respectively.

Zero-test. The procedure $\text{isZero}(\text{pp}, \mathbf{p}_{zt}, [\alpha])$ outputs 1 if $[\alpha] \in S^{(0)}$ and 0 otherwise.

Extraction. This procedure extracts a random function of ring elements from their encoding. That is, $\text{ext}(\text{pp}, \mathbf{p}_{zt}, [\alpha])$ outputs $s \in \{0, 1\}^\lambda$ satisfying

1. For any $\alpha, \alpha' \in R$ and $[\alpha], [\alpha'] \in S^{(\alpha)}$, $\text{ext}(\text{pp}, \mathbf{p}_{zt}, [\alpha]) = \text{ext}(\text{pp}, \mathbf{p}_{zt}, [\alpha'])$.
2. The distribution $\{\text{ext}(\text{pp}, \mathbf{p}_{zt}, [\alpha]) : \alpha \in_R R\}$ is nearly uniform over $\{0, 1\}^\lambda$.

RES Procedures, Restricted Version. We are also interested in a restricted version of RES procedures, in which a set of permitted circuits is predetermined at the instance generation. There are some reasons to advocate a need for such the restricted concept. First, as aforementioned, we have to preclude adversary from computing an inverse encoding of $[\omega]$, with only additions and multiplications. To the best of our knowledge, it seems infeasible to compute an inverse without knowing the underlying ring. For example, if R is a ring of integers modulo N , then, to the best of our knowledge, all efficient ways⁹ to compute a modular inverse require to know N . One of which is using the extended Euclidean algorithm and the other is using the Euler theorem. Nevertheless, we do not know whether there could exist generic algorithm computing modular inverses or breaking cryptographic assumptions such as the DL and the SubM if ring operations are unlimitedly allowed. We leave it as an interesting open problem to prove generic hardness of cryptographic problems, or to disprove by proposing generic attack algorithms. (For the restricted version of RES over $R = \mathbb{Z}_N$, we prove the generic hardness of cryptographic assumptions in the subsection 3.3.) Second, there is a practical constraint for the restricted version, which is the limit of state-of-the-art technology. As we mentioned in the introduction, unlimitedly performing two homomorphic operations over exponentially large message space is an open problem at the present time.¹⁰

⁹ polynomial time in the security parameter

¹⁰ As of now, the running time of known bootstrapping technique is exponential in $\log N$ [42].

Therefore, in the restricted version, we modify the *Instance Generation* procedure to take as input (or produce as output) the description of the permitted arithmetic circuits \mathcal{P}_C . (In our RES procedures, sampled encodings and ring operations over them can be considered as variables and polynomials over them. For the restricted version, we mainly consider the case that R is the ring of integers modulo N for some N and any polynomials over \mathbb{Z}_N of degree less than a predetermined limit is allowed to perform.) In addition, just like the *real-life version* of GES, we modify *Zero-test* and *Extraction* procedures to allow negligible errors in the restricted version.

Additional Procedures for Multilinear Maps. To make RES become further similar to multilinear maps, we define additional procedures, though these are not newly defined but by using the previous procedures.

Encoding. The algorithm $\text{enc}(\text{pp}, [\alpha_0])$ takes as input **params** and an encoding $[\alpha_0]$ for some $\alpha_0 \in R$, and outputs $\text{mul}(\text{pp}, [\omega], [\alpha_0]) \rightarrow [\alpha_0 \cdot \omega]$.

κ -linear Map. Given **pp** and κ encodings $[\alpha_1 \cdot \omega], \dots, [\alpha_\kappa \cdot \omega]$, the κ -linear map $\kappa\text{-linear}(\text{pp}, [\alpha_1 \cdot \omega], \dots, [\alpha_\kappa \cdot \omega])$ performs

$$\text{mul}(\dots(\text{mul}(\text{pp}, [\alpha_1 \cdot \omega], [\alpha_2 \cdot \omega]) \dots), [\alpha_\kappa \cdot \omega]) \rightarrow [\alpha_1 \cdots \alpha_\kappa \cdot \omega^\kappa] \in \mathcal{S}^{(\alpha_1 \cdots \alpha_\kappa \cdot \omega^\kappa)}.$$

We write $[\alpha_1 \cdot \omega] \cdots [\alpha_\kappa \cdot \omega]$ as a shorthand for applying κ -linear map.

3.2 Assumptions

We change Definition 3 of group-based hard problems to be adapted for RES.

Definition 5 (DL/MCDH/Ext-MCDH/Ext-MDDH problems) For any security parameter $\lambda \in \mathbb{N}$ and the multilinearity $\kappa \in \mathbb{N}$, the DL/ κ -MCDH/Ext- κ -MCDH/Ext- κ -MDDH problems are defined as follows: Fix a RES. Parameters are generated as

$$\begin{aligned} (\text{pp}, \mathbf{p}_{zt}) &\leftarrow \text{InstGen}(1^\lambda, R); \\ \forall j = 1.. \kappa + 1, x_j &\leftarrow \text{samp}(\text{pp}), y_j := x_j \cdot [\omega] \leftarrow \text{enc}(\text{pp}, x_j); \\ y' &\leftarrow \kappa\text{-linear}(\text{pp}, y_1, \dots, y_\kappa), y_c \leftarrow \text{mul}(\text{pp}, x_{\kappa+1}, y'), \\ y_r &\leftarrow \text{enc}(\text{pp}, \text{samp}(\text{pp})); \\ \gamma_c &\leftarrow \text{ext}(\text{pp}, \mathbf{p}_{zt}, y_c), \gamma_r \leftarrow \text{ext}(\text{pp}, \mathbf{p}_{zt}, y_r); \end{aligned}$$

- *Discrete Logarithm (DL) problem:* given **pp**, \mathbf{p}_{zt} , and y_1 , find $[x'_1]$ such that $\text{ext}(\text{pp}, \mathbf{p}_{zt}, [x_1]) = \text{ext}(\text{pp}, \mathbf{p}_{zt}, [x'_1])$.
- *κ -Multilinear Computational Diffie-Hellman (CDH) problem:* given **pp**, \mathbf{p}_{zt} , and $\{y_j\}_{j=1}^{\kappa+1}$, find y'_c such that $\text{ext}(\text{pp}, \mathbf{p}_{zt}, y'_c) = \gamma_c$.
- *Extraction κ -Multilinear Computational Diffie-Hellman (CDH) problem:* given **pp**, \mathbf{p}_{zt} , and $\{y_j\}_{j=1}^{\kappa+1}$, find γ_c .
- *Extraction κ -Multilinear Decisional Diffie-Hellman (DDH) problem:* distinguish two distributions $\mathcal{D}_\delta^{\text{ext}} = (\text{pp}, \mathbf{p}_{zt}, \{y_j\}_{j=1}^{\kappa+1}, \gamma_\delta)$ for $\delta \in \{c, r\}$.

The DL/CDH/Ext-CDH/Ext-DDH assumptions state that the advantage of any polynomial-time adversary in attacking the corresponding problem is negligible in the security parameter.

Submodule Membership Problem over Product Ring. We consider RES with a product ring $R = R_1 \times R_2$, where each ring R_i contains a non-identity invertible element $\omega_i \in R_i$, so that $\omega = (\omega_1, \omega_2)$ is a non-identity invertible element in R . For this variant, we need an additional procedure for sampling encodings from the subring $R_1 \times \{0\} \subset R$. Note that all elements in $R_1 \times \{0\}$ are not invertible in R since 0 is not invertible in R_2 . Considering a subring $R_1 \times \{0\}$ of the product ring R for RES, we need to modify to **InstGen** algorithm to take a subring as input.

Subring Sampler. The randomized **subsamp**(**pp**) takes **pp** as input and outputs an encoding $[\alpha_1]$ for a nearly uniform element $\alpha_1 \in_R R_1 \times \{0\}$.

One can show that the above variant using a product ring has the desirable properties, so called *projecting* [32, 23] and *cancelling* [23] in the context of composite order bilinear groups (of course, multilinear maps, too [33]). Furthermore, we can define the *Submodule Membership problem*, which is an analogue of the Subgroup Membership problem in composite order multilinear groups.¹¹

Definition 6 (Submodule Membership problems) *For any security parameter $\lambda \in \mathbb{N}$, the Submodule Membership problem is defined as follows: Fix a RES for a product ring. Parameters are generated as*

$$\begin{aligned} (\mathbf{pp}, \mathbf{p}_{zt}) &\leftarrow \text{InstGen}(1^\lambda); \\ y_0 &\leftarrow \text{samp}(\mathbf{pp}), \quad y_1 \leftarrow \text{subsamp}(\mathbf{pp}); \end{aligned}$$

- *Submodule Membership (SubM) problem: given $(\mathbf{pp}, \mathbf{p}_{zt}, y_\delta)$ for $\delta \in_R \{0, 1\}$, determine δ .*

3.3 Generic Hardness of Cryptographic Problems in Restricted RES

In this subsection, to show the reliability of cryptographic assumptions over RES, we provide generic hardness of the MCDH and SubM problems. Informally, generic algorithm \mathcal{A} is defined as a game with the challenger \mathcal{O}_{ch} . During the game \mathcal{A} has access to oracles, which are managed by \mathcal{O}_{ch} , for two ring operations in R and equality testing between the results of two sequences of ring operations. We relegate the formal definition of generic RES algorithm in Appendix A.2.

In particular, we consider the case such that $R = \mathbb{Z}_N$ and \mathcal{A} is permitted to evaluate any polynomial of degree less than $2^{\delta-\lambda}$, where N is a product of δ -bit random primes, which covers the case of our realization in Section 5. If we set $\delta = 2\lambda$, then \mathcal{A} can perform any polynomials of degree at most 2^λ , which is reasonably large for a lot of applications.

We use a notation $T_{n,\delta}$ to denote a set of all n -products of δ -bit primes. That is,

$$T_{n,\delta} = \left\{ \prod_{i=1..n} p_i \mid \text{for } i = 1..n, p_i \in \{\delta\text{-bit primes}\} \right\}.$$

Theorem 1 *Let $N \stackrel{\$}{\leftarrow} T_{n,\delta}$, where $n \geq 1$ and $\delta > \lambda$ for the security parameter λ . Let \mathcal{A} be an arbitrary generic polynomial time algorithms (possibly adaptively) computing polynomials of degree less than $2^{\delta-\lambda}$. Then, there is only negligible probability that \mathcal{A} outputs the solution of the MCDH problem, where the MCDH problem is defined over $(\mathbb{Z}_N, +)$. (Therefore, it directly implies that the DL problem is intractable, too.)*

Theorem 2 *Let $N_1 \stackrel{\$}{\leftarrow} T_{n,\delta}$ and $N_2 > 1$ be coprimes, where $n \geq 1$ and $\delta > \lambda$ for the security parameter λ . If arbitrary generic polynomial time algorithm computes only (possibly adaptively) polynomials of degree less than $2^{\delta-\lambda}$, then its advantage in solving the SubM problem is negligible, where the SubM problem is defined over $(\mathbb{Z}_{N_1} \times \{0\}, +)$, which is identified as the submodule of order N_1 in $(\mathbb{Z}_{N_1 N_2}, +)$.*

Both proofs of Theorem 1 and Theorem 2 are given in Appendix A.2.

To prove Theorem 1, we mainly use Shoup’s proving technique for the hardness of DL problem in the generic group model [43]. Although in our model the adversary \mathcal{A} can perform multiplication, contrary to the generic group model, we restrict the maximum degree of polynomials that \mathcal{A} can generate. Similar to Shoup’s technique, we can also use the Schwartz-Zippel lemma to bound the adversary’s success probability. Except for using larger degree polynomials, almost all parts of our proof is identical to the original Shoup’s proof.

To prove Theorem 2, we need a new technique, which is quite different from those of Theorem 1 and other generic proofs in generic ring model [36]. More precisely, the factoring assumption is usually required to prove generic hardness of the subset membership problem. (e.g., the hardness of the subgroup membership

¹¹ Considering multilinear maps and self bilinear maps as being defined over modules rather than abelian groups, we prefer to call *submodule* membership problem than *subgroup* membership problem. However, if we restrict our attention to abelian groups (that is, \mathbb{Z}_N -modules), we use the term of *subgroup* membership problem.

problem in the generic group model due to Katz, Sahai and Waters [37] and the subset membership problem in the generic ring model due to Jager and Schwenk [36]) In our proof, unknown modulus $N = N_1N_2$ is essential, in particular N_1 . Roughly speaking, we show that if N_1 is hidden and large enough, then \mathcal{A} cannot make N_1 part of given element to be zero. If N_1 parts of all elements generated by \mathcal{A} are all non-zero, then \mathcal{A} cannot get any information about N_2 part since generic algorithm \mathcal{A} cannot obtain any information from the representation of elements. That is, if \mathcal{A} cannot make N_1 part to be zero, then \mathcal{A} cannot check whether N_2 part is zero or not regardless of the size of N_2 , where N_2 is coprime to N_1 , so that the submodule membership problem is generically intractable. The essential idea behind the proof is to show that polynomially bounded algorithm \mathcal{A} cannot make any polynomial f of small degree ($< 2^{\delta-\lambda}$) such that it has large numbers of common roots over \mathbb{Z}_{p_i} for large numbers of δ -bit coprimes p_i 's. Then, \mathcal{A} cannot make any element to be its N_1 -part zero for unknown random N_1 and given randomly chosen problem instance. Our proof is of independent interest since it proves the power of hiding message space for the first time; in particular, the factoring assumption is not necessary in the context of subset membership problem.

3.4 Application to Multipartite Diffie-Hellman Key Exchange

We present a one-round N -way Diffie-Hellman key exchange protocol using RES in Appendix B.

4 Homomorphic Encryption with Unknown Message Space

In this section, we construct a new integer-based efficient scale-invariant FHE for a realization of RES. The proposed scheme has the property of unknown message space.

We first give a high-level intuition for our homomorphic encryption scheme. Our construction basically uses a recent multiplication methodology due to Cheon and Stehlé [15] since it is only one construction of scale-invariant FHE with unknown message space. However, their methodology inherently requires large public key size, in particular, multiplication key size. Roughly speaking, the heart of the idea in the Cheon-Stehlé scheme is to minimize the empty space in *Approximate Common Divisors (ACD)* instances, which is reserved for handling error increasing. Therefore, to minimize error increasing for multiplication, a multiplication in the Cheon-Stehlé scheme is carried out by bit operations and additions over ciphertexts; the multiplication procedure in the Cheon-Stehlé scheme consists of two steps. Given two ciphertexts, first bitwise-decompose each ciphertext and compute a tensor product of two ciphertexts. Second, compute the inner product between the result and the multiplication key. Therefore, the size of multiplication key depends on γ^2 where a ciphertext consists of γ bits. Roughly speaking again, the size of ciphertext γ is determined by the complexity of lattice attack; that is, $\gamma = \omega((\eta - \rho)^2 \log \lambda)$, where $\eta - \rho$ means the size of the empty space reserved for error increasing in ciphertext. For small message space, it is sufficient to set $\eta - \rho = O(\log \lambda)$, so that the Cheon-Stehlé scheme achieves very short ciphertext size for binary message space. However, for exponentially large message space $\eta - \rho$ should be larger than the bit-size of message space, that is, $\eta - \rho = O(\lambda)$, then there is no advantage over the other integer-based schemes [22, 20]. Therefore, we modify the Cheon-Stehlé scheme to increase the empty space in ciphertext, so that at least one multiplication of two ciphertext over the integers keeps some specific intermediate format. Then, the remaining procedure is similar to the Cheon-Stehlé scheme; that is, bitwise-decompose and compute the inner product between the result and the multiplication key. Since the multiplication over the integers results only 2γ bits, the multiplication key depends on not γ^2 but 2γ , so that our public key size is much shorter than that of the Cheon-Stehlé scheme. Note that to make sufficiently large empty space in ciphertexts, contrary to the Cheon-Stehlé scheme, we use the modulus p_i^2 instead of p_i and this technique is already used by Coron, Lepoint, and Tibouchi [20], but the Coron-Lepoint-Tibouchi FHE requires the description of the message space for multiplication procedure.

We use the following notations: For three real numbers a , b , and e , we write $a = b \pmod{e}$ if $a - b$ is a multiple of e . Given a real number a , $[a]_g$ is a remainder of dividing a by g , which is also denoted by $a \bmod g$. (Do not confuse with the notation $[a]$ of an encoded ring element.) Given positive integers p , q_0 , and ρ , we define the distribution $\mathcal{D}_{p,q_0}^\rho = \{qp^2 + r : q \in \mathbb{Z} \cap [0, q_0], r \in \mathbb{Z} \cap (-2\rho, 2\rho)\}$. Let n be a positive integer. As in

[9], given $x \in \mathbb{Z} \cap [0, 2^n)$ and $y \in \mathbb{R}$, we define $\text{BD}_n(x) = (x_0, \dots, x_{n-1}) \in \{0, 1\}^n$ such that $x = \sum_{i=0}^{n-1} x_i 2^i$ and $\text{PT}_n(y) = (y, 2y, \dots, 2^{n-1}y) \in \mathbb{R}^n$. We omit n if it is clear in the context.

4.1 The Basic Construction

We present our homomorphic encryption HE. In the following description, we assume that η , γ and τ are functions in the security parameter λ and will be decided later.

HE.KeyGen(1^λ) : Determine the message space \mathbb{Z}_g . Pick an η -bit random prime p with $\text{gcd}(p, g) = 1$ and a γ -bit integer $x_0 = q_0 p^2 + r_0$ with $q_0 \xleftarrow{\$} \mathbb{Z} \cap [0, 2^\gamma/p^2)$ and $r_0 \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho)$. Sample $y', x_i \leftarrow \mathcal{D}_{p, q_0}^\rho$ for $i = 1.. \tau$ and set $y = y' + \lfloor \frac{p}{g} \rfloor$. Generate the multiplication key $\mathbf{z} = (z_0, \dots, z_{2\gamma-1})$ as follows. For $i = 0..2\gamma - 1$, $z_i = q'_i \cdot p^2 + \lfloor \frac{p}{g} \lfloor \frac{g^2 2^{2i}}{p^2} \rfloor \rfloor_g + r'_i$, where $q'_i \xleftarrow{\$} \mathbb{Z} \cap [0, q_0)$ and $r'_i \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho)$. Finally, output $\text{pk} = (x_0, x_1, \dots, x_\tau, y, z_0, \dots, z_{2\gamma-1})$ and $\text{sk} = p$.

HE.Encrypt($\text{pk}, m \in \mathbb{Z}_g$) : Choose a random subset $S \subset \{1, \dots, \tau\}$ and output $c \leftarrow [\sum_{i \in S} x_i + y \cdot m]_{x_0}$.

HE.Decrypt(sk, c) : Output $m \leftarrow \left[\left[\frac{g}{p} \cdot c \right] \right]_g$.

HE.Add(pk, c_1, c_2) : Output $c \leftarrow [c_1 + c_2]_{x_0}$.

HE.Mult(pk, c_1, c_2) : Output $c \leftarrow [\langle \text{BD}_{2\gamma}(c_1 \cdot c_2), \mathbf{z} \rangle]_{x_0}$.

HE.Eval($\text{pk}, \mathcal{C}, c_1, \dots, c_k$) : Given a circuit \mathcal{C} with k inputs, and k ciphertexts $c_i \leftarrow \text{HE.Encrypt}(m_i)$, it performs the integer addition and multiplication gates over c_i 's. When the depth of \mathcal{C} is below the certain level, it holds

$$\text{HE.Decrypt}(\text{sk}, \text{HE.Eval}(\text{pk}, \mathcal{C}, c_1, \dots, c_k)) = \mathcal{C}(m_1, \dots, m_k)$$

with probability larger than or equal to $1 - \lambda^{-\omega(1)}$.

Note that the multiplication key \mathbf{z} can be considered as a vector of encryptions of *secret-key-dependent-messages*; that is, $\mathbf{z} = \mathbf{x}' + \lfloor \frac{p}{g} [\text{PT}_{2\gamma}(\frac{g^2}{p^2})] \rfloor_g$, where $\mathbf{x}' = (x'_0, \dots, x'_{2\gamma-1})$ and $x'_i \leftarrow \mathcal{D}_{p, q_0}^\rho$. Therefore, as usual in the context of homomorphic encryption, one has to assume circular security of the underlying encryption scheme [44, 11, 20, 15].

To show the correctness of the proposed scheme, we first show that the ciphertext is of the form $q \cdot p^2 + \lfloor \frac{p}{g} \rfloor \cdot (m_i + r^* g) + r$ for some noise r^* and r . Next, we show that how many operations on encryptions can be supported, with keeping noise size to be controllable; that is, the resulting ciphertext is decryptable.

Lemma 2 *Let $c \leftarrow \text{HE.Encrypt}(\text{pk}, m)$ and $c_i = q_i p^2 + \lfloor \frac{p}{g} \rfloor ([m_i]_g + r_i^* g) + r_i$ for some q_i, r_i^* , and r_i . Then, the followings hold.*

1. c is of the form $qp^2 + \lfloor \frac{p}{g} \rfloor (m + r^* g) + r$ for some q, r , and r^* with $|r| \leq 2(\tau + g)(2^\rho - 1)$ and $r^* = 0$.
2. **HE.Add**(pk, c_1, c_2) $\rightarrow qp^2 + \lfloor \frac{p}{g} \rfloor ([m_1 + m_2]_g + r^* g) + r$ with $|r| \leq |r_1 + r_2| + 2^\rho - 1$ and $|r^*| \leq |r_1^* + r_2^*| + 1$.
3. **HE.Mult**(pk, c_1, c_2) $\rightarrow qp^2 + \lfloor \frac{p}{g} \rfloor ([m_1 m_2]_g + r^* g) + r$ with $|r^*| < 2\gamma$ and $|r| < 2^{2\rho^* + 4} g^2 (|r_1| + |r_2|)$ if $|r_1^*| < 2^{\rho^*}$ and $|r_2^*| < 2^{\eta + \rho^* + 1}$.

We relegate the proof of Lemma 2 to Appendix C.1. In ciphertexts, there are two different type noises r^* and r . Addition increases noises only linearly. Usually, the bottleneck in practicality of HE is multiplication. As for multiplication, r^* is changed to small one regardless of original ones r_1^* and r_2^* . On the other hand, r is affected by r_i^* , but the size of $|r|$ is increased only linearly if r_1 and r_2 have the same size; that is, if $\log_2 |r_1| = \log_2 |r_2|$, then the noise length in bits has only grown additively $2\rho^* + 4 + 2\log_2 g + 1$. For example, if one has to product 2^m encryptions with the same error size $\log_2 |r|$, then first one makes a binary tree with 2^m leaves, puts encryptions in leaves, and makes an intermediate node as a multiplication of children nodes.

Then, the final output size is upper bounded by $m(2\rho^* + 4 + 2\log_2 g + 1) + \log |r|$.¹² When considering the number of allowable additions and our parameter selection for γ , it is sufficient to assume $\rho^* = O(\lambda)$.

Lemma 3 *Given a ciphertext $c = qp^2 + \left\lfloor \frac{p}{g} \right\rfloor ([m]_g + r^*g) + r$, if $|r^*|g^2 + |r|g < 2^{\eta-1}$, then the decryption algorithm $\text{HE.Decrypt}(\text{sk}, c)$ outputs $[m]_g$.*

Proof. We have, for some $\delta \in [-g/2, g/2] \cap \mathbb{Z}$, $\left\lfloor \frac{g}{p}c \right\rfloor = \left\lfloor qp + \frac{g}{p} \frac{p+\delta}{g} ([m]_g + r^*g) + \frac{g}{p}r \right\rfloor = qp + ([m]_g + r^*g) + \left\lfloor \frac{\delta([m]_g + r^*g) + gr}{p} \right\rfloor$. If $|r^*|g^2 + |r|g < 2^{\eta-1}$, then it is congruent to m modulo g .

Definition 7 *The scheme HE is L-homomorphic if for any depth L circuit C and any set of inputs $m_1, \dots, m_k \in \mathbb{Z}_g$, it holds that*

$$\text{HE.Decrypt}(\text{sk}, \text{HE.Eval}(\text{pk}, C, c_1, \dots, c_k)) = C(m_1, \dots, m_k)$$

with probability $\geq 1 - \lambda^{-\omega(1)}$, where $(\text{pk}, \text{sk}) \leftarrow \text{HE.KeyGen}$ and $c_i \leftarrow \text{HE.Encrypt}(m_i)$ for $i = 1..k$.

Theorem 3 *Suppose $|r^*|$ is bounded by 2^{ρ^*} . The scheme HE is L-homomorphic if $\eta - \rho \geq L(2\rho^* + 2\log_2 g + 5) + \log_2 g + \log_2(g + \tau) + 3$.*

The proof of Theorem 3 is given in Appendix C.1.

Parameter Selection. For the security parameter λ , the below are constraints that the parameters of the scheme must satisfy. We are interested in the case when g is exponentially large in λ .

- $\rho \geq \lambda$ to avoid the brute force attacks on the noise [10, 22],
- $\eta = \rho + O(L\lambda)$ from Theorem 3, where L is the multiplicative depth of the circuit to be evaluated,
- $\gamma = \omega(\eta^2 \log_2 \lambda)$, to thwart lattice-based attacks (see [44, 21, 16, 20]),
- $\tau \geq \gamma + 2\lambda$ in order to apply the leftover hash lemma. (To guarantee statistically close to the uniform distribution over \mathbb{Z}_{x_0} for a γ -bit integer x_0 , we need at least $\gamma + 2\lambda$ number of x_i 's.)

Concretely, one can set parameters: $\rho = \lambda$, $\eta = O(\rho + L\lambda)$, $\gamma = O(L^2\lambda^3)$, and $\tau = \gamma + 2\lambda$. The difference of η from that of [20] comes from the size of message space \mathbb{Z}_g . Since $\log_2 g$ is set to $O(\lambda)$, the number of allowable additions need to be exponentially many, which enlarges the size of ρ^* . However if we take $g = O(\log_2 \lambda)$ as usual homomorphic encryptions, we can set $\eta = \rho + O(L \log_2 \lambda)$. In this case the concrete parameter follows those of [20], i.e., $\rho = \lambda$, $\eta = \tilde{O}(L + \lambda)$, $\gamma = \tilde{O}(L^2\lambda + \lambda^2)$, and $\tau = \gamma + 2\lambda$. We remark that each element in pk can be compressed to roughly $2\eta + \lambda$ bits as in [22].

4.2 Generalization to Batch Homomorphic Encryption

We present the batch version BHE of HE, which is a building block of our construction for multilinear map. For coprimes a_0, \dots, a_n and integers $\alpha_0, \dots, \alpha_n$, we use a notation $c = \text{CRT}_{a_0, \dots, a_n}(\alpha_0, \dots, \alpha_n)$ for the Chinese remaindering; that is, c is the unique integer in $[0, \prod_{i=0}^n a_i)$ that is congruent to α_i modulo a_i for $i = 0..n$. In the paper, for the sake of simplicity, we will frequently use the Chinese remaindering for coprimes q_0, p_1^2, \dots, p_n^2 , and so in this case let us omit these coprime numbers and otherwise we will explicitly write; that is, CRT means $\text{CRT}_{q_0, p_1^2, \dots, p_n^2}$. We define the distribution $\mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$ as

$$\{\text{CRT}(q, r_1 \dots, r_n) : q \xleftarrow{\$} \mathbb{Z} \cap [0, q_0), r_i \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho)\}.$$

To extend our scheme to the batch version, we follow the previous approach [11, 20]; for a message $\mathbf{m} = (m_1, \dots, m_n) \in \mathbb{Z}_{g_1} \times \dots \times \mathbb{Z}_{g_n}$, a corresponding encryption is of the form $c = \left\lfloor \frac{p_i}{g_i} \right\rfloor (m_i + r_i^*g_i) + r_i \pmod{p_i}$ for all $i = 1..n$ and $c = q \pmod{q_0}$ for some random integers q, r_i^* 's and r_i 's. The resulting scheme BHE is as follows.

¹² Note that if one computes sequentially, the final error size will be exponentially large in m .

BHE.KeyGen(1^λ) : Determine the message space $\mathcal{R} = \mathbb{Z}_{g_1} \times \dots \times \mathbb{Z}_{g_n}$. Pick an η -bit random coprime integers p_1, \dots, p_n with $\gcd(p_i, g_i) = 1$ and let $\pi = \prod_{i=1}^n p_i$. Choose a γ -bit integer

$$x_0 = q_0 \cdot \pi^2 + \text{CRT}_{p_1^2, \dots, p_n^2}(r_{0,1}, \dots, r_{0,n})$$

with $r_{0,i} \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho)$ and $q_0 \xleftarrow{\$} \mathbb{Z} \cap [0, 2^\gamma/\pi^2)$ an coprime integer with the p_i 's. Sample $x_i \leftarrow \mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$ for $i = 1..n$ and $y'_i \leftarrow \mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$ for $i = 1..n$ and set

$$y_i = y'_i + \left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot \left(\left(\frac{\pi^2}{p_i^2} \right)^{-1} \bmod p_i^2 \right) \cdot \frac{\pi^2}{p_i^2}.$$

Generate the multiplication key $\mathbf{z} = (z_0, \dots, z_{2\gamma-1})$ as follows. For $i = 0..2\gamma-1$, z_i is defined as

$$\text{CRT}(q'_i, \left\lfloor \frac{p_1}{g_1} \left[\frac{g_1^2 2^i}{p_1^2} \right]_{g_1} \right\rfloor + r'_{i,1}, \dots, \left\lfloor \frac{p_n}{g_n} \left[\frac{g_n^2 2^i}{p_n^2} \right]_{g_n} \right\rfloor + r'_{i,n}),$$

where $q'_i \xleftarrow{\$} \mathbb{Z} \cap [0, q_0)$ and $r'_{i,1}, \dots, r'_{i,n} \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho)$. Finally, output

$$\mathbf{pk} = (x_0, x_1, \dots, x_\tau, y_1, \dots, y_n, z_0, \dots, z_{2\gamma-1})$$

and $\mathbf{sk} = (p_1, \dots, p_n)$.

BHE.Encrypt($\mathbf{pk}, \mathbf{m} \in \mathcal{R}$) : Choose a random subset $S \subset \{1, \dots, \tau\}$ and output $c \leftarrow \left[\sum_{i \in S} x_i + \sum_{i=1}^n y_i \cdot m_i \right]_{x_0}$.

BHE.Decrypt(\mathbf{sk}, c) : Output $m_i \leftarrow \left[\left\lfloor \frac{g_i}{p_i} \cdot c \right\rfloor \right]_{g_i}$ for $i = 1..n$.

BHE.Add(\mathbf{pk}, c_1, c_2) : Output $c \leftarrow [c_1 + c_2]_{x_0}$.

BHE.Mult(\mathbf{pk}, c_1, c_2) : Output $c \leftarrow [\langle \text{BD}(c_1 \cdot c_2), \mathbf{z} \rangle]_{x_0}$.

BHE.Eval($\mathbf{pk}, \mathcal{C}, c_1, \dots, c_k$): Identical to HE.Eval

Parameter Selection. We remark that the asymptotic parameter selection for our BHE is the same to that of HE since x_0 is not error-free. For realization of RES, we consider parameters when $\log_2 g_i = O(\lambda)$ and $q_0 = 1$, i.e., $\prod_{i=1}^n p_i$. Thus we have $\gamma = O(n\eta)$. As in HE, one can set $\rho = \lambda$, $\eta = \rho + L\lambda$. To satisfy $\gamma = \omega(\eta^2)$, n is set to $\omega(\eta \log_2 \lambda)$.

4.3 Unknown Message Space

We argue that in the proposed schemes, the message space \mathbb{Z}_g can be *provably* hidden.¹³ That is, it is not necessary to know g in the procedure of public algorithms BHE.Encrypt, BHE.Add, and BHE.Mult and g is required only in the procedure of BHE.KeyGen and BHE.Decrypt. More rigorously, we can show that g is hidden from \mathbf{pk} under the n -DACD* assumption, i.e., we do not require additional assumption for moving to BHE with unknown message space. We define the n -DACD* assumption in Appendix C.2.

To this end, we show that \mathbf{pk} is indistinguishable from random integers modulo x_0 , where g is not used in the generation x_0 , under the n -DACD* assumption. Formally, we give the following lemma.

Lemma 4 *Let BHE be the encryption scheme with L -homomorphic property, given in Section 4.2. In particular, parameters are chosen according to our setting. Let $(\mathbf{pk}, \mathbf{sk}) \leftarrow \text{BHE.KeyGen}$. Define the distribution*

$$\mathcal{D}_{x_0} = \{\mathbf{x}' \xleftarrow{\$} ([0, x_0) \cap \mathbb{Z})^{\tau+2\gamma+n}\}.$$

Then, there exists an algorithm to generate $\mathbf{pk}' = (x_0, \mathbf{x}) \in \mathbb{Z} \times \mathbb{Z}^{\tau+2\gamma+n}$ such that (1) $(\mathbf{pk}', \mathbf{sk})$ is a valid pair of public key and secret key for BHE, (2) \mathbf{x} is indistinguishable from $\mathbf{x}' \leftarrow \mathcal{D}_{x_0}$ under the n -DACD assumption, and (3) the resulting scheme is at least $(L-1)$ -homomorphic.*

¹³ Instead, the size of message space, denoted by ρ' , is known. Furthermore, we have to slightly modify the definition of the semantic security, so that the adversary has to choose a message from $\{0, 1\}^{\rho'+1}$ for the challenge phase.

	Schemes	CT	PK	Msg Sp.
DGHV	DGHV10 [44]	$\tilde{O}(\lambda^5)$	$\tilde{O}(\lambda^{10})$	unknown
	CNT12a [22]	$\tilde{O}(\lambda^5)$	$\tilde{O}(\lambda^5)$	unknown
	CNT12b [22]	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(L^3\lambda^4)$	known
	batch [11]	$\tilde{O}(\lambda^5)$	$\tilde{O}(n\lambda^7)$	unknown
Scale-inv.	CLT14a [20]	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(L^3\lambda^4)$	known
	CLT14b [20]	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(nL^3\lambda^4)$	known
	CS15 [15]	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(L^5\lambda^7)$	unknown
Ours	non-batch	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(L^3\lambda^4)$	unknown
	batch	$\tilde{O}(L^2\lambda^3)$	$\tilde{O}(nL^3\lambda^4)$	unknown

λ : the security parameter, n : the number of slots in each BHE,
 Msg Sp.: Message Space, L : multiplicative depth,
 CNT12a: Compressed pk, CNT12b: Modulus-switching
 CLT14a: Non-batch version, CLT14b: Batch version

Table 1. Comparison of integer-based HE schemes

The proof of Lemma 4 is straightforward using the standard hybrid argument. Lemma 4 directly implies that **pk** of BHE leaks no information about the message space g since the message space \mathbb{Z}_g is not used at the generation of x_0 .

4.4 Comparison

We compare complexities of performance and properties among integer-based (batch) homomorphic encryption (denoted by (B)HE) schemes in Table 1. In the table we say message space \mathbb{Z}_g is can be unknown if g is not required in the procedure of public algorithms and required only in the procedures of key generation and decryption. We consider the compressed form of ACD instance [22] in public key except the original DGHV (the first row).

5 Our Proposal for Ring Encoding System

Construction We describe our proposal for RES in Figure 1. In the description, we assume that $n, \eta, \gamma, \rho, \ell$ and ν are functions in the security parameter λ and will be decided later. First, let us briefly provide a high-level description of our construction.

Instantiation. $\text{InstGen}(1^\lambda, L)$ algorithm takes the security parameter and the depth L of permitted circuits. Next it generates all secret and public parameters. It runs $\text{BHE.KeyGen}(1^\lambda)$ as a subroutine, where BHE is L -homomorphic,¹⁴ and generates additional parameters $\{x'_j\}_{j=1..l}, y, s$ and \mathbf{p}_{zt} . x'_j 's and y are encryptions of random messages by BHE.Encrypt .¹⁵ \mathbf{p}_{zt} and s are for **isZero** and **ext** algorithms, which are not used in BHE.

Sampling. $\text{samp}(\text{pp})$ generates an encryption of random element $\mathbf{m} \in \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$ by subset-sum of encryptions x'_j 's. If we set $\ell \geq n\alpha + 2\lambda$, then the leftover hash lemma implies that the distribution of $(\{x'_j\}_{j=1..l}, \mathbf{m})$ is statistically indistinguishable from the distribution of $(\{x'_j\}_{j=1..l}, \mathbf{m}')$, where $\mathbf{m}' \xleftarrow{\$} \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$.

Operations. **add** and **mul** algorithms are exactly the same as **BHE.Add** and **BHE.Mult** of the underlying HE scheme, respectively.

¹⁴ All parameters $\gamma, \eta,$ and τ are, in fact, functions in λ and L .

¹⁵ **BHE.Encrypt** is for public key encryption scheme. In fact, we use symmetric key version of encryption function instead of **BHE.Encrypt** since **InstGen** algorithm already knows secret key of homomorphic encryption scheme.

• **InstGen**($1^\lambda, L$): Run $\text{BHE.KeyGen}(1^\lambda) \rightarrow (\text{pk} = (x_0, \{x_i\}_{i=1}^\gamma, \{y_i\}_{i=1}^n, \{z_i\}_{i=1}^{2\gamma-1}), \text{sk} = (p_1, \dots, p_n))$, where q_0 is set by 1 and $\pi^2 = \prod_{i=1..n} p_i^2$ is γ bits. Note that π^2 is not published. Set $\text{pk} = (x_0, z_0, \dots, z_{2\gamma-1})$, which is the evaluation key for BHE.Add and BHE.Mult . (The other parts in pk are for BHE.Encrypt .) Generate another secrets; the underlying hidden ring $R = \mathbb{Z}_{g_1 \dots g_n}$ for $g_i \stackrel{\$}{\leftarrow} \{\alpha\text{-bit primes}\}$ and special non-identity invertible element $\omega = \text{CRT}_{g_1, \dots, g_n}(\omega_1, \dots, \omega_n)$ for $\omega_i \stackrel{\$}{\leftarrow} \mathbb{Z} \cap (0, g_i)$. Compute public parameters, which will be used for sampling, encoding, zero-testing, and extraction;

(1) Sampling: $(m_{ij}) \stackrel{\$}{\leftarrow} R^{n \times \ell}$, $r_{ij} \stackrel{\$}{\leftarrow} (-2^\rho, 2^\rho) \cap \mathbb{Z}$ and let $\mathbf{m}_j = (m_{1j}, \dots, m_{nj})$. For $j = 1.. \ell$, generate $x'_j := [\mathbf{m}_j] \leftarrow \text{CRT}_{p_1^2, \dots, p_n^2}(\lfloor \frac{p_1}{g_1} \rfloor m_{1j} + r_{1j}, \dots, \lfloor \frac{p_n}{g_n} \rfloor m_{nj} + r_{nj})$, so that x'_j has the form $\text{BHE.Encrypt}(\text{pk}, \mathbf{m}_j)$ with smaller error size.

(2) Encoding: Choose $r_i \stackrel{\$}{\leftarrow} (-2^\rho, 2^\rho) \cap \mathbb{Z}$ and generate $y := [\omega] \leftarrow \text{CRT}(\lfloor \frac{p_1}{g_1} \rfloor \omega_1 + r_1, \dots, \lfloor \frac{p_n}{g_n} \rfloor \omega_n + r_n)$, so that y has the form $\text{BHE.Encrypt}(\text{pk}, \omega)$ with smaller error size.

(3) Zero-testing: $N \stackrel{\$}{\leftarrow} \{\gamma + 4\eta + 1\text{-bit primes}\}$, generate $\mathbf{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$ and $(\alpha_1, \dots, \alpha_n) \in \mathbb{Z}^n$, which will be specified later. Then, generate $(\mathbf{p}_{zt})_j = \sum_{i=1..n} h_{ij} \cdot \alpha_i \cdot p_i^{-1} \bmod N$ for $j = 1..n$

(4) Extraction: Generate a seed s for a strong randomness extractor Extract .

Finally, output $\text{pp} := \{\hat{\text{pk}}, \{x'_j\}_{j=1}^\ell, y, s\}$ and \mathbf{p}_{zt} .

- **samp**(pp): Choose a random subset $S \subset \{1, \dots, \ell\}$ and output $c \leftarrow \sum_{j \in S} x'_j$, where the sum is performed by using BHE.Add .
- **add/mul**(pp, c_1, c_2): Output $c_{\text{add}} \leftarrow \text{BHE.Add}(\hat{\text{pk}}, c_1, c_2)$ / $c_{\text{mul}} \leftarrow \text{BHE.Mult}(\hat{\text{pk}}, c_1, c_2)$.
- **isZero**($\text{pp}, \mathbf{p}_{zt}, c$): Output 1 if $\|c \cdot \mathbf{p}_{zt} \bmod N\|_\infty < N \cdot 2^{-\nu}$ for some parameter ν specified later. Otherwise, output 0.
- **ext**($\text{pp}, \mathbf{p}_{zt}, c$): Output $\text{Extract}_s(\text{msbs}_\nu(c \cdot \mathbf{p}_{zt} \bmod N))$, where msbs_ν takes the ν most significant bits of the result.

***** Additional Procedures for Multilinear Maps *****

- **enc**(pp, c): Output $c' \leftarrow \text{mul}(\text{pp}, c, [\omega])$.
- **κ -linear**($\text{pp}, c_1, \dots, c_\kappa$): Output $c' \leftarrow \text{mul}(\dots(\text{mul}(\text{pp}, c_1, c_2) \dots, c_\kappa)$.

Fig. 1. Ring Encoding Scheme

Re-randomization. In our scheme, there is no particular re-randomization procedure. In the previous approximated multilinear maps, the re-randomization of encodings is used to attain an analogue of the discrete logarithm assumption; in particular [18], the way to encode at higher level is just modulus multiplication, so that its inverse is trivial unless there is additional re-randomization procedure. Inverting the multiplication procedure in the underlying BHE scheme (without re-randomization) is a hard subset-sum problem, so that it can be a oneway function. We expect that it still preserves the onewayness with given the zero-testing parameter. The detailed analysis is given in Section 6.

Zero-testing. The basic idea for **isZero** is identical to the previous approximate multilinear maps, in particular [18]. We first explain how to generate the zero-testing vector $\mathbf{p}_{zt} \in \mathbb{Z}^n$, and then why **isZero** works. Similarly to [18], we choose a random prime integer N of size $\gamma + 4\eta + 1$ bits, and then for $i = 1..n$ generate pairs (α_i, β_i) satisfying the following conditions, by using LLL in dimension 2 (that is, by using Lagrange-Gauss reduction)¹⁶. 1) $|\alpha_i| < 2^{\eta-1}$, 2) $|\beta_i| < \frac{4}{3} \cdot \frac{N}{2^{\eta-1}} < 2^{2-\eta} N$, and 3) $\beta_i = \alpha_i \cdot (u'_i/p_i) \pmod N$ where $u'_i = (\frac{\pi}{p_i})^2 [(\frac{\pi}{p_i})^{-2}]_{p_i}$. We also generate an integer matrix $\mathbf{H} = (h_{ij}) \in \mathbb{Z}^{n \times n}$, as in [19, 18]; \mathbf{H} is invertible in \mathbb{Z} and $\|\mathbf{H}\|_\infty \leq 2^\beta$ and $\|\mathbf{H}^{-1}\|_\infty \leq 2^\beta$ for some β specified later. The generation of the \mathbf{H} matrix is given in Appendix D.2. Finally, the zero-testing vector \mathbf{p}_{zt} is computed as $(\mathbf{p}_{zt})_j = \sum_{i=1}^n h_{ij} \cdot \alpha_i \cdot p_i^{-1} \bmod N$. **isZero** algorithm computes $\omega = c \cdot \mathbf{p}_{zt} \bmod N$ and test whether $\|\omega\|_\infty$ is small or not. We relegate the details, in particular, the correctness of zero-testing procedure, in Appendix D.1.

¹⁶ A simple algorithm to generate (α_i, β_i) is given in [18]; $(\alpha_i \lceil N/B^2 \rceil, \beta_i)$ is the shortest vector of the lattice generated by the rows of $(\begin{smallmatrix} \lceil N/B^2 \rceil & u'_i/p_i \bmod N \\ 0 & N \end{smallmatrix})$, where $B = (3/4)^{1/4} 2^{\eta-1}$.

Extraction. ext algorithm is exactly the same as [18], which is essentially the same as [25, 19]. Using the seed s in **pp**, one can extract uniform bits from a strong randomness extractor. We relegate the extraction algorithm in Appendix D.3. (Also, one can find it in [18].)

Parameter Selection. Parameters follow the selection of BHE. The bit-size α of the prime g_i must be large enough so that the message space \mathbb{Z}_{g_i} can be concealable. One can set $\alpha = 2\lambda$ (see Section 3.3). Other parameters can be chosen as follows: 1) The number ℓ of level-0 encodings for **samp** must satisfy $\ell \geq n\alpha + 2\lambda$ in order to apply the leftover hash lemma [19] and 2) For a conservative security as in [18] we let $\beta := 3\lambda$.

6 Security Analysis of RES Proposal

When choosing the parameters of RES to allow computing polynomials of degree less than $2^{\delta-\lambda}$, we can prevent the generic attacks on MCDH and SubM problems (Theorem 1 and 2). On the other hand, it does not guarantee that our RES is secure against non-generic attacks such as zeroizing attack and lattice attacks [12, 13, 40, 18, 17, 35]. In this section, we argue the reliability of the security of our RES construction by considering all known non-generic attacks and plausible new attacks.

6.1 Zeroizing Attacks

We review the Cheon et al.’s zeroizing attack and its variants against integer-based multilinear map [19] in Appendix E.1. In this subsection, we briefly describe a recent attack [13, 40] on the new multilinear map [18], and we apply all known zeroizing attacks [12, 13, 40] to our RES.

A New Attack on CLT15. To thwart the zeroizing attacks, Coron, Lepoint and Tibouchi modified both the multiplication and zero-testing processes [18], so that the result of the Cheon et al. attack does not have a quadratic form any more. To force the multiplication process to make a non-linear part, they hide the modulus $\pi = \prod_i p_i$, which is used in both multiplications and zero-testing in the previous version [19]. Instead, ladders of zero encodings are used to make a similar effect to the modular reduction and those add non-linear noises in the resulting encoding. Let us briefly explain their multiplication procedure. For the sake of the simplicity, we only consider a multiplication of two level-1 encodings. Let $\mathbf{X} = \{X_k\}_k$ be a level-1 ladder of zero encodings. Then, a reduction by \mathbf{X} , denoted by $[\cdot]_{\mathbf{X}}$, is subtracting X_k ’s to reduce the size of input encoding in a certain way. For example, given $c_1 = a_{1i}/z \pmod{p_i}$ and $c_2 = a_{2i}z \pmod{p_i}$, one can compute a multiplication of two encodings by $[c_1 \cdot c_2]_{\mathbf{X}} = c_1 \cdot c_2 - \sum_k b_k X_k < \pi$ for some $b_k \in \{0, 1\}$. Then, the result is congruent to $(a_{1i}a_{2i} - \sum_k b_k r_{ki})/z^2$ modulo p_i , where $X_k = r_{ki}/z^2 \pmod{p_i}$, and the additional term $\sum_k b_k r_{ki}$ makes hard to apply the Cheon et al. attack. It is obvious that if π is known, one can remove such the additional term, so that it is essential to hide π in the modified CLT map. Since π is hidden, the zero-testing process should be modified accordingly. Coron et al. also devise an alternative zero-testing process, which does not require π explicitly but use a new independent larger integer N , and the new zero-testing method also adds non-linear term. These new multiplication and zero-testing processes produce nonlinear part in resulting zero-testing value and so make it hard to extract quadratic form of CRT component over the integers, which is main idea of [12].

Recently, new attacks on the modified CLT scheme [18] are provided in [13] and [40] independently. Both attacks share essentially the same idea; Although the ladders of zero encodings and a new modulus N make non-linear parts, one can trace coefficients of ladders of zero encodings applied, so that remove the effect of non-linear part generated by the modified multiplication and zero-testing processes over the integers. Resultingly, the security of CLT15 [18] is reduced to CLT13 scheme [19] as Minaud and Fouque pointed out [40].

We note that Coron et al.’s zero-testing process is quite similar to ours, but completely different multiplication process. We also hide the modulus π^2 due to complete different reason; what we intended is not to prevent zeroizing attack, but to prevent a trivial factoring of π^2 by computing a square root of it. That is, our zero-testing process also generates some non-linear part, but the security of our scheme against the

zeroizing attack does not depend on it. Therefore, in this paper, in order to verify the non-applicability of the newly proposed attack to our scheme, we only focus on the aspect of the attack removing non-linear part generated by the ladder reduction and omit discussion on the nonlinear part generated by the modulus N .

Let us describe a high level intuition behind the new zeroizing attacks [13, 40]; that is, we explain how to remove nonlinear parts. Let a be a large size encoding. To run zero-testing procedure correctly, first we have to reduce the size of encoding a . We recall the zero-testing lemma provided in [13], which says about the conditions for the correctness of the zero-testing procedure, in an informal way. Denote the resulting ladder reduced encoding by $a' = [a]_{\mathbf{X}} = a - \sum_k b_k X_k < X_1$ for $b_k \in \{0, 1\}$. Let $v_i = [g_i \cdot \hat{p}_i^{-1} / z^\kappa]_{p_i} \cdot \hat{p}_i$ for $\hat{p}_i = \prod_{j \neq i} p_j$.

Lemma 5 (Informal, [13, Zero-testing lemma]) *If $x = \text{CRT}_{p_i}(r_i \cdot g_i / z^\kappa)$ with $|r_i| < p_i/2$, then x can be uniquely written as $x = \sum_i r_i \cdot v_i + t \cdot \pi$. Furthermore, if x is smaller than or equal to X_1 , the equation*

$$[p_{zt} \cdot x]_N = \sum_i r_i \cdot v'_i + t \cdot \pi',$$

holds over the integers, where $v'_i = [p_{zt} \cdot v_i]_N$ and $\pi' = [p_{zt} \cdot \pi]_N$.

For the sake of simplicity, we let $\mathbf{X} = \{X_k\}_k$ be a level- κ ladder of zero encodings. We write $a = \sum_i s_i \cdot v_i + t \cdot \pi$ and $X_k = \sum_i s_{ki} \cdot v_k + t_k \cdot \pi$. Since $a' = [a]_{\mathbf{X}}$ is also an encoding of zero and smaller than X_1 , we have

$$[p_{zt} \cdot a']_N = [p_{zt} \cdot (a - \sum_k b_k X_k)]_N = \sum_i (s_i - \sum_k b_k s_{ki}) \cdot v'_i + (t - \sum_k b_k t_k) \cdot \pi',$$

where the last equality holds over the integers by Lemma 5. If one can find $\sum_i s_i \cdot v'_i + t \cdot \pi'$ over \mathbb{Z} , the quadratic form of input variables over the integers can be obtained, so that the security of the scheme [18] is reduced to the previous version [19]. To find such integer values, we try to remove $\sum_k b_k s_{ki}$ part by considering $f_k := \sum_i s_{ki} \cdot v'_i + t_k \pi'$ over \mathbb{Z} .¹⁷ Using the Lemma 5, the value f_k 's can be computed inductively. The value f_1 is immediately obtained from $[p_{zt} \cdot X_1]_N$, since X_1 equals to X_1 . We compute $[p_{zt} \cdot [X_2]_{\mathbf{X}}]_N$ for $[X_2]_{\mathbf{X}} = X_2 - X_1 \leq X_1$, so that we have $\sum_i (s_{2i} - s_{1i}) \cdot v'_i + (t_2 - t_1) \cdot \pi'$ over the integers. We obtain f_2 by computing $[p_{zt} \cdot [X_2]_{\mathbf{X}}]_N + f_1$ over the integers. For $k > 2$, we can compute f_k inductively. Finally, we can eliminate the nonlinear part $\sum_k b_k s_{ki}$ in $[p_{zt} \cdot a']_N$ by adding $\sum_k b_k f_k$, hence we obtain $\sum_i s_i \cdot v'_i + t \cdot \pi'$ over the integers.

Apply to Our RES. The new attack against the modified CLT scheme [13, 40] can be considered as a method to eliminate non-linear parts and thus to reduce the security of the modified CLT map into that of the original CLT map. Therefore, to argue the security of the proposed RES, we focus on the zeroizing attack against the original CLT map [12]. Nevertheless, we also show that the technique used in the new attack [13, 40] can hardly apply to our RES scheme.

We first show that the multiplication process of our RES generates non-linear parts, which plays an essential role to prevent the zeroizing attack [12]. For easy explanation, we first simplify the variables in encodings by ignoring a noise from rounding function. In fact, there are too many factors generating non-linear part even to analyze. Therefore, for the sake of simplicity, we ignore rounding functions in encodings. We remark that BHE.Mult algorithm outputs $[(\text{BD}_{2\gamma}(a \cdot b), \mathbf{z})]_{x_0}$ for given encodings $a = \text{CRT}_{p_i^2}(a_i)$ and $b = \text{CRT}_{p_i^2}(b_i)$. If we ignore the rounding $[\cdot]$, $\mathbf{z} = \mathbf{x}' + \left(\text{CRT} \left(\left[\frac{p_1}{g_1} \left[\frac{g_1^2 2^j}{p_1^2} \right]_{g_1} \right], \dots, \left[\frac{p_n}{g_n} \left[\frac{g_n^2 2^j}{p_n^2} \right]_{g_n} \right] \right) \right)_{j=1..2\gamma}$ is a vector of encryption of $\text{PT}_{2\gamma}(g_i^2/p_i^2)$. Therefore, $\text{BHE.Mult}(\text{pk}, a, b)$ can be written as

$$\text{CRT}_{p_i^2} \left(\frac{g_i}{p_i} \cdot (a_i \cdot b_i) + \sum_{j=1}^{2\gamma} e_j \cdot r_{ij} \right), \quad (1)$$

¹⁷ One can easily find b_i , since ladder reduction by \mathbf{X} is deterministic process.

from the observation that $x \cdot y = \langle \text{BD}_{2^\gamma}(x), \text{PT}_{2^\gamma}(y) \rangle$ for any $x \in \mathbb{Z} \cap [0, 2^{2^\gamma})$ and $y \in \mathbb{R}$. Here, e_j is the j -th element of $\text{BD}(a \cdot b)$ and $r_{ij} = r'_{ij} + p_i \cdot r''_{ij}$ where r'_{ij} is modulo p_i value of j -th element of \mathbf{x}' and r''_{ij} is $\left(g_i^2 \cdot 2^j / p_i^2 - [g_i^2 \cdot 2^j / p_i^2]_{g_i} \right) / g_i$. We have the nonlinear term $(\sum_{j=1}^{2^\gamma} e_j \cdot r_{ij})$ generated by the multiplication process. Since each bit in $a \cdot b$ contains the information of more than one slot in the Chinese remaindering, one can interpret our multiplication procedure as making *inter-slot diffusion* of the values in the Chinese remaindering. This is one of the most important difference from the previous candidate multilinear constructions over the integers [19, 18], whose multiplication processes do not have property of the inter-slot diffusion. That is, each value in slots of the Chinese remaindering is multiplied independently.

Next, we present more detailed attack and its non-applicability to our RES. As the attack in [12], we consider a product of three encodings including, any encoding a , an encoding of zero b , and a target encoding c . Let $a = \text{CRT}_{p_i^2}(a_i)$, $b = \text{CRT}_{p_i^2}(b_i)$, and $c = \text{CRT}_{p_i^2}(c_i)$, where all encoded by our RES. We consider a product of those three encodings $d = \text{BHE.Mult}(\text{pp}, \text{BHE.Mult}(\text{pp}, a, b), c)$. Then we have

$$\begin{aligned} d &= \text{CRT}_{p_i^2} \left(\frac{g_i}{p_i} \left(\frac{g_i}{p_i} \cdot a_i \cdot b_i + \sum_j e_j \cdot r_{ij} \right) \cdot c_i + \sum_k e'_k \cdot r_{ik} \right) \\ &= \text{CRT}_{p_i^2} \left(\frac{g_i^2}{p_i^2} \cdot a_i \cdot b_i \cdot c_i + \frac{g_i}{p_i} \cdot \sum_j e_j \cdot r_{ij} \cdot c_i + \sum_k e'_k \cdot r_{ik} \right) \\ &= \sum_{i=1}^n \left(\frac{g_i^2}{p_i^2} \cdot a_i \cdot b_i \cdot c_i + \frac{g_i}{p_i} \cdot \sum_j e_j \cdot r_{ij} \cdot c_i + \sum_k e'_k \cdot r_{ik} \right) \cdot u'_i + \alpha' \cdot \pi, \end{aligned}$$

for some $e_j, e'_k \in \{0, 1\}$ and $\alpha' \in \mathbb{Z}$, where $u'_i = \left(\frac{\pi}{p_i} \right)^2 \left[\left(\frac{\pi}{p_i} \right)^{-2} \right]_{p_i}$. Using the definition of p_{zt} and the analysis in Section 5, we have the following equation over the integers:

$$[p_{zt} \cdot d]_N = \sum_{i=1}^n \left(\frac{g_i^2}{p_i^2} \cdot a_i \cdot b_i \cdot c_i + \frac{g_i}{p_i} \cdot \sum_j e_j \cdot r_{ij} \cdot c_i + \sum_k e'_k \cdot r_{ik} \right) \cdot u''_i + \alpha' \cdot \pi',$$

for $u''_i = [p_{zt} \cdot u'_i]_N$ and $\pi' = [p_{zt} \cdot \pi]_N$. We see that the additional part $\left(\frac{g_i}{p_i} \cdot \sum_j e_j \cdot r_{ij} \cdot c_i + \sum_k e'_k \cdot r_{ik} \right)$ depends on d in a nonlinear way. Therefore, if we apply Cheon et al. attack, we cannot construct quadratic form over the integers.

We consider more attacks than the above naive approach. One may try to subtract the nonlinear part using the technique in [13, 40] which removes the effect of the ladders of zero encodings. We argue that it is hard to apply the technique to our RES due to the following reasons. The nonlinear part generated by our multiplication process is related to both input encodings a, b and the multiplication key \mathbf{z} ; in (1) e_j is related to both a and b , and r_{ij} is related to \mathbf{z} . The external noise r_{ij} in \mathbf{z} inserted when running homomorphic multiplication. Since the multiplication key are not encodings of zero, contrary to the ladders of zero encodings [18], it seems hard to remove the nonlinear part without changing underlying message. We note that nonlinear part in [18] comes from the ladder of zero encodings which is independent from message. Furthermore, the noise r_{ij} is added not slot-wise but inter-slot diffused manner on encodings a and b due to the bit decomposition BD procedure. We also consider an attack computing not BHE.Mult but integer multiplication of encodings. We can write $a \cdot b$ over the integers as follows:

$$a \cdot b = \text{CRT}_{p_i^2}(a_i) \cdot \text{CRT}_{p_i^2}(b_i) = \text{CRT}_{p_i^2}([a_i \cdot b_i]_{p_i^2}) + B \cdot \pi^2 = \sum_{i=1}^n ([a_i \cdot b_i]_{p_i^2}) \cdot u_i + B' \cdot \pi^2,$$

for some B and B' of bit length γ and $u_i = \pi^2 / p_i^2 \cdot ((\pi^2 / p_i^2)^{-1} \bmod p_i^2)$. Due to large B' , we obtain only modulo N value of $p_{zt} \cdot (a \cdot b)$ not over \mathbb{Z} . To get quadratic form of CRT component over the integers, the zero-testing value $p_{zt} \cdot (a \cdot b)$ is relatively small compared to modulus N . On the other hand, we cannot run valid zero-testing procedure on $a \cdot b$ because of large value B' , and so we only have an equation over \mathbb{Z}_N not over \mathbb{Z} .

We emphasize again that we use the same zero-testing parameter [18] with the totally different purpose. In [18], they use an independent N to hide $\pi = \prod_i p_i$ when running zero-testing procedure and consequently it produces additional nonlinear which prevents the zeroizing attack [12]. On the other hand, the modulus N no longer plays a role in defending the zeroizing attacks due to [13]. In our RES, we use the same zero-testing parameter $(\mathbf{p}_{zt})_\ell = \sum_{i=1}^n h_{i\ell} \cdot \alpha_i \cdot p_i^{-1} \bmod N$ to prevent factoring attacks on π not the zeroizing attacks.

6.2 Lattice Attacks

We consider lattices attacks on level-0 encodings, public key including zero-testing key and multiplication key. We relegate the details in Appendix E.2

6.3 Assumptions

Decision type assumptions are useful in design of cryptographic protocols. Generally, we have the following relations among variants of the Diffie-Hellman assumption on our RES.

$$MCDH \dashrightarrow Ext-MCDH \dashrightarrow Ext-MDDH \dashrightarrow MDDH$$

Here, Problem A \dashrightarrow Problem B means that Problem A is harder than or equal to Problem B.

As proved in Lemma 1, the MDDH problem is easy in our RES, regardless of knowing the group order. It is straightforward that the Ext-MCDH problem is easier or equivalent to the MCDH problem and the Ext-MDDH problem is easier or equivalent to the Ext-MCDH problem. Therefore, if we assume that the Ext-MDDH is a hard problem, the hardness of the Ext-MCDH and MCDH are guaranteed. It seems hard to reduce to more classical assumptions such as AGCD. In this section, therefore, we show the reason why we believe that the Ext-MDDH is hard.

Let $y_j = x_j \cdot [\omega]$ and y_c, y_r be κ -level encodings of $[\omega^\kappa] \cdot \prod_{i=1}^{\kappa+1} x_i$ and a random message, respectively. The MDDH problem is to determine $b \in \{c, r\}$ from y_b and the Ext-MDDH problem is to determine $b \in \{c, r\}$ from $\gamma_b \leftarrow \text{ext}(\text{pp}, \mathbf{p}_{zt}, y_b)$. In the case of MDDH problem, one can easily check whether y_b is an encoding of $[\omega^\kappa] \cdot \prod_{i=1}^{\kappa+1} x_i$ or not by checking an equality

$$\text{ext}(\text{pp}, \mathbf{p}_{zt}, (\kappa + 1)\text{-linear}(\text{pp}, y_1, \dots, y_{\kappa+1})) \stackrel{?}{=} \text{ext}(\text{pp}, \mathbf{p}_{zt}, \text{mul}(\text{pp}, y_b, [\omega])),$$

which holds if and only if $b = c$. In Ext-MDDH problem, on the other hand, it is hard to make use of the above equality check, since extraction algorithm ext and multiplication algorithm mul do not commute. The reason why it is hard to compute $\text{ext}(\text{pp}, \mathbf{p}_{zt}, \text{mul}(\text{pp}, y_b, [\omega]))$ from $\text{ext}(\text{pp}, \mathbf{p}_{zt}, y_b)$ and $[\omega]$ comes from complicated homomorphic multiplication and independent modulus N in zero-testing procedure.

Submodule Membership problem(SubM). We explain a reason why SubM problem seems to be hard in our RES. We first describe attack against SubM in the original CLT [19]. Even though the CLT scheme is totally broken by Cheon et al. attack and hence SubM assumption does not hold any more, but this gives an evidence why SubM seems hard in our RES setting.

Roughly speaking, *Submodule Membership problem* is to determine an element $\mathbf{m} \in R$ is in a strict submodule R' or not for a given encoding $c = \text{enc}(\mathbf{m})$. Since the underlying hidden ring in our construction is $R = \prod_{i=1}^n \mathbb{Z}_{g_i}$ for prime integers g_i 's, the strict nontrivial subgroup R' is of the form $R' = \prod_{i=1}^t \{0\} \times \prod_{i=t+1}^n \mathbb{Z}_{g_{k_i}}$ for some $1 \leq t < n$ and $k_i \in [1, n]$, where \times is a canonical product. We assume $k_i = i$ and $t = 1$. The problem is given as $c_0 = \text{enc}(\mathbf{m}_0)$ and $c_1 = \text{enc}(\mathbf{m}_1)$ where $\mathbf{m}_0 = (m_{0i})_i \in R$ and $\mathbf{m}_1 = (m_{1i})_i \in R'$. Let c be a zero encoding and compute D_c, D_{c_0} and D_{c_1} for fixed auxiliary sets A and B as in Equation (3):

$$\begin{aligned} D_c &= X_A \cdot \text{diag}(g_1 \cdot r_1, g_1 \cdot r_2, \dots, g_n \cdot r_n) \cdot \text{diag}(u_1, \dots, u_n) \cdot X_B \\ D_{c_0} &= X_A \cdot \text{diag}(g_1 \cdot r_{01} + m_{01}, g_1 \cdot r_{02} + m_{02}, \dots, g_n \cdot r_{0n} + m_{0n}) \cdot \text{diag}(u_1, \dots, u_n) \cdot X_B \\ D_{c_1} &= X_A \cdot \text{diag}(g_1 \cdot r_{11}, g_1 \cdot r_{12} + m_{12}, \dots, g_n \cdot r_{1n} + m_{1n}) \cdot \text{diag}(u_1, \dots, u_n) \cdot X_B. \end{aligned}$$

One can distinguish between c_0 and c_1 by comparing two values $\alpha_0 = \gcd(\det D_c, \det D_{c_0})$ and $\alpha_1 = \gcd(\det D_c, \det D_{c_1})$; α_1 is about g_1 times larger than α_0 with high probability.

It seems difficult to apply this attack on our RES because two non-linear parts comes from modulus N and complicated homomorphic multiplication procedure. Therefore we conjecture that SubM assumption holds in our RES.

Onewayness of BHE.Mult. Contrary to the previous candidate multilinear maps, there is no re-randomization procedure in our RES. Nevertheless, we argue that an analogue of the discrete logarithm assumption is still hard; given pp and $\text{enc}(\text{pp}, c)$, it is hard to find c' such that $\text{ext}(\text{pp}, \mathbf{p}_{zt}, c) = \text{ext}(\text{pp}, \mathbf{p}_{zt}, c')$.

Finding the exact solution c from given $\text{enc}(\text{pp}, c) = \text{BHE.Mult}(\text{pk}, [\boldsymbol{\omega}], c) = [\langle \text{BD}([\boldsymbol{\omega}] \cdot c), \mathbf{z} \rangle]_{x_0}$ is exactly like inverting $\text{BHE.Mult}(\text{pk}, [\boldsymbol{\omega}], \cdot)$ function. Let $\mathbf{c} = \text{BD}([\boldsymbol{\omega}] \cdot c) \in \{0, 1\}^{2\gamma}$. Then, finding c is equivalent to finding \mathbf{c} , so that inverting $\text{BHE.Mult}(\text{pk}, [\boldsymbol{\omega}], c)$ is a subset-sum problem of finding \mathbf{c} with an instance (\mathbf{z}, x_0) . Even though, due to \mathbf{p}_{zt} , \mathbf{z} is not indistinguishable from the uniform distribution over \mathbb{Z}_{x_0} , \mathbf{z} has sufficient entropy; each z_i is generated by using a $(\gamma - \eta)$ -bit random integer and a ρ -bit random integer. Since there are 2γ components in \mathbf{z} , the subset-sum problem instance (\mathbf{z}, x_0) has large enough density $2\gamma / \log_2 x_0 = 2 > 1$ and \mathbf{c} also has sufficient entropy (exactly the same entropy as that of c), we can conclude that this subset-sum problem is exponentially hard, so that equivalently breaking the onewayness of $\text{BHE.Mult}_{c_0}(c_1)$ is infeasible. (For the detailed analysis for the subset-sum problem, we refer to [41, 21].) To break the discrete logarithm assumption, one may try to find $c' \neq c$ such that $\text{ext}(\text{pp}, \mathbf{p}_{zt}, c) = \text{ext}(\text{pp}, \mathbf{p}_{zt}, c')$. If one knows c , it is easy to generate such a c' ; just add an encoding of 0 to c . However, it seems difficult to find such a c' without firstly finding c .

6.4 Noise Growth of BHE

When the proposed BHE scheme is used for restricted RES, it is required that one is not allowed to compute polynomials over the predefined degree in order to guarantee the generic hardness of MCDH and SubM assumptions (Theorem 1 and 2). Unfortunately, it is not straightforward to analyze the distribution of noises after evaluations over BHE-ciphertexts due to the complicated multiplication procedure in terms of randomness diffusion. Indeed, it is a reason why we expect the multiplication procedure prevents the zeroizing attack. Instead, we provide an evidence to support that our BHE is proper to the use of restricted RES by experiments. That is, we show that one cannot evaluate polynomials of degree $2^{\delta - \lambda}$, where δ is the bit-size of g_i 's, in the sense that any evaluation of reasonably selected arbitrary polynomials increases noises large enough, so that the noises will wrap up the modulus p_i 's. Even though our experiments may not perfectly guarantee the non-existence of polynomials that increase noises very small, we expect that one can hardly find such polynomials since the message space and messages, over which polynomials are evaluated, are randomly chosen and completely hidden from the viewpoint of adversary.

We relegate our experimental result in Appendix F.

Acknowledgement We are very thankful to Jung Hee Cheon, Changmin Lee, and Hansol Ryu for explanation about their zeroizing attack on the new CLT multilinear map [13] and helpful discussion about non-applicability to our RES. The third author is also very grateful to Hyung Tae Lee for invaluable discussions at the very early stage of this paper.

References

1. D. Aggarwal and U. M. Maurer. Breaking RSA generically is equivalent to factoring. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 36–53. Springer, 2009.
2. M. R. Albrecht, P. Farshim, D. Hofheinz, E. Larraia, and K. G. Paterson. Multilinear maps from obfuscation. In *IACR-ePrint* (<http://eprint.iacr.org/2015/780>), 2015.
3. P. Ananth, D. Gupta, Y. Ishai, and A. Sahai. Optimizing obfuscation: avoiding Barrington’s theorem. In *ACM Conference on Computer and Communications Security 2014*, pages 646–658. ACM, 2014.

4. D. Apon, Y. Huang, J. Katz, and A. J. Malozemoff. Implementing cryptographic program obfuscation. In *IACR-ePrint* (<http://eprint.iacr.org/2014/779>), 2014.
5. D. Boneh and R. J. Lipton. Algorithms for black-box fields and their application to cryptography (extended abstract). In *CRYPTO 1996*, volume 1109 of *LNCS*, pages 283–297. Springer, 1996.
6. D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.
7. D. Boneh, B. Waters, and M. Zhandry. Low overhead broadcast encryption from multilinear maps. In *CRYPTO (1) 2014*, volume 8616 of *LNCS*, pages 206–223. Springer, 2014.
8. D. Boneh, D. J. Wu, and J. Zimmerman. Immunizing multilinear maps against zeroizing attacks. In *IACR-ePrint* (<http://eprint.iacr.org/2015/930>), 2014.
9. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) Fully homomorphic encryption without bootstrapping. In *ITCS 2012*, pages 309–325, 2012.
10. Y. Chen and P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 502–519. Springer, 2012.
11. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 315–335. Springer, 2013.
12. J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. In *EUROCRYPT (1) 2015*, volume 9056 of *LNCS*, pages 3–12. Springer, 2015.
13. J. H. Cheon, C. Lee, and H. Ryu. Cryptanalysis of the new CLT multilinear maps. In *IACR-ePrint* (<http://eprint.iacr.org/2015/934>), 2015.
14. J. H. Cheon and D. H. Lee. A note on self-bilinear maps. *Bulletin of the Korean Mathematical Society*, 46(2):303–309, 2013.
15. J. H. Cheon and D. Stehlé. Fully homomorphic encryption over the integers revisited. In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 513–536. Springer, 2015.
16. H. Cohn and N. Heninger. Approximate common divisors via lattices. In *IACR-ePrint* (<http://eprint.iacr.org/2011/437>), page 437, 2011.
17. J. Coron, C. Gentry, S. Halevi, T. Lepoint, H. K. Maji, E. Miles, M. Raykova, A. Sahai, and M. Tibouchi. Zeroizing without low-level zeroes: new MMAP attacks and their limitations. In *CRYPTO (1) 2015*, volume 9215 of *LNCS*, pages 247–266. Springer, 2015.
18. J. Coron, T. Lepoint, and M. Tibouchi. New multilinear maps over the integers. In *CRYPTO (1) 2015*, volume 9215 of *LNCS*, pages 267–286. Springer, 2015.
19. J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1) 2013*, volume 8042 of *LNCS*, pages 476–493. Springer, 2013.
20. J.-S. Coron, T. Lepoint, and M. Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *PKC 2014*, volume 8383 of *LNCS*. Springer, 2014.
21. J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *CRYPTO 2011*, volume 6841 of *LNCS*, pages 487–504. Springer, 2011.
22. J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 446–464. Springer, 2012.
23. D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, 2010.
24. E. S. V. Freire, D. Hofheinz, K. G. Paterson, and C. Striecks. Programmable hash functions in the multilinear setting. In *CRYPTO (1) 2013*, volume 8042 of *LNCS*, pages 513–530. Springer, 2013.
25. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, 2013.
26. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS 2013*, pages 40–49. IEEE Computer Society, 2013.
27. S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO (2) 2013*, volume 8043 of *LNCS*, pages 479–499. Springer, 2013.
28. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure attribute based encryption from multilinear maps. In *IACR-ePrint* (<http://eprint.iacr.org/2014/622>), 2014.
29. S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. In *IACR-ePrint* (<http://eprint.iacr.org/2015/666>), 2014.
30. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *STOC 2013*, pages 467–476. ACM, 2013.

31. C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. In *TCC 2015*, volume 9015 of *LNCS*, pages 498–527. Springer, 2015.
32. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
33. G. Herold, J. Hesse, D. Hofheinz, C. Ràfols, and A. Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In *CRYPTO (1) 2014*, volume 8616 of *LNCS*, pages 261–279. Springer, 2014.
34. S. Hohenberger, A. Sahai, and B. Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In *CRYPTO (1) 2013*, volume 8042 of *LNCS*, pages 494–512. Springer, 2013.
35. Y. Hu and H. Jia. Cryptanalysis of GGH map. In *IACR-ePrint (http://eprint.iacr.org/2015/301)*, 2015.
36. T. Jager and J. Schwenk. On the analysis of cryptographic assumptions in the generic ring model. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 399–416. Springer, 2009.
37. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, 2008.
38. A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, 2014.
39. H. T. Lee and J. H. Seo. Security analysis of multilinear maps over the integers. In *CRYPTO (1) 2014*, volume 8616 of *LNCS*, pages 224–240. Springer, 2014.
40. B. Minaud and P.-A. Fouque. Cryptanalysis of the new multilinear map over the integers. In *IACR-ePrint (http://eprint.iacr.org/2015/941)*, 2015.
41. P. Q. Nguyen and J. Stern. The hardness of the hidden subset sum problem and its cryptographic implications. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 31–46. Springer, 1999.
42. K. Nuida and K. Kurosawa. (Batch) Fully homomorphic encryption over integers for non-binary message spaces. In *EUROCRYPT (1) 2015*, volume 9056 of *LNCS*, pages 537–555. Springer, 2015.
43. V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–266. Springer, 1997.
44. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 24–43. Springer, 2010.
45. T. Yamakawa, S. Yamada, G. Hanaoka, and N. Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In *CRYPTO (2) 2014*, volume 8617 of *LNCS*, pages 90–107. Springer, 2014.

A Generic Easiness and Hardness of Problems

A.1 Generically Easy Problems

Lemma 1 (Easy Problems) *For any positive integer κ , there are polynomial time attacks to the κ -MDDH problem and the κ -DLIN problem defined over the unbounded multilinear maps via self bilinear maps, regardless of knowing the group order.*

Proof. Let $\tilde{e}(g, g) = g^t$ for some t so that $e_\kappa(g, \dots, g) = g^{t^{\kappa-1}}$. Given a κ -MDDH problem instance $(g, g^{a_1}, \dots, g^{a_{\kappa+1}}, g^c)$, one can check whether $c = t^{\kappa-1} a_1 \cdots a_{\kappa+1}$ holds or not by checking an equality $e_{\kappa+1}(g^{a_1}, \dots, g^{a_{\kappa+1}}) \stackrel{?}{=} \tilde{e}(g, g^c)$, which holds if and only if $c = t^{\kappa-1} a_1 \cdots a_{\kappa+1}$.

Let $e_\kappa(g_1, \dots, g_\kappa) = g^s$ for some s . Given a κ -DLIN problem instance $(g, g_1, \dots, g_\kappa, g_1^{a_1}, \dots, g_\kappa^{a_\kappa}, g^c)$, one can check whether $c = \sum_{i=1.. \kappa} a_i$ holds by checking an equality $\prod_{i=1.. \kappa} e_{\kappa+1}(g_1, \dots, g_{i-1}, g_i^{a_i}, \dots, g_\kappa, g) \stackrel{?}{=} \tilde{e}(g^c, g)$. The left-hand side is equal to $\prod_{i=1.. \kappa} \tilde{e}(g^{s a_i}, g) = g^{t s \sum_{i=1.. \kappa} a_i}$. The right-hand side is equal to $g^{t s \sum_{i=1.. \kappa} a_i}$ if and only if $c = \sum_{i=1.. \kappa} a_i$. \square

A.2 Generic Hardness of Cryptographic Problems in Restricted RES

We formalize a new concept of generic algorithm in the RES context, in terms of a game between algorithm \mathcal{A} and a challenger \mathcal{O}_{ch} . \mathcal{O}_{ch} is given as input description of a ring R , over which RES is defined, and problem instance $x_0, \dots, x_\ell \in R$. We assume that the ring description contains efficient ways to uniformly sample an element in R and to perform two ring operations $(+, \cdot)$ in R . During interaction, \mathcal{O}_{ch} keeps a list *List*, which

is initialized by $(0, x_0), \dots, (-\ell, x_{-\ell})$ at the beginning of the game. \mathcal{A} can issues the following three types of oracle queries to \mathcal{O}_{ch} .

- *Sampling.* \mathcal{O}_{ch} chooses a random element $x_{\ell+i}$ from R for the i -th sampling query and stores $(-\ell-i, x_{\ell+i})$ in $List$.
- *Ring Operation.* Suppose that \mathcal{A} issues the k -th ring operation query (i, j, \circ) for $i, j < k$ and $\circ \in \{+, \cdot\}$. Then, \mathcal{O}_{ch} appends a vector $(k, (i, j, \circ))$ into $List$.
- *Equality-Test.* Let P_k be a straight line program that takes as input x_k 's of $(k, x_k) \in List$ for $k \leq 0$ and performs a sequence of operations in order from $(1, *)$ to $(i, *)$ in $List$. Once \mathcal{O}_{ch} receives (i, j) from \mathcal{A} , \mathcal{O}_{ch} checks whether $P_i = P_j$. \mathcal{O}_{ch} returns **true** to \mathcal{A} if the equality holds. Otherwise, \mathcal{O}_{ch} sends **false** to \mathcal{A} .

Theorem 1 Let $N \stackrel{\$}{\leftarrow} T_{n,\delta}$, where $n \geq 1$ and $\delta > \lambda$ for the security parameter λ . Let \mathcal{A} be an arbitrary generic polynomial time algorithms (possibly adaptively) computing polynomials of degree less than $2^{\delta-\lambda}$. Then, there is only negligible probability that \mathcal{A} outputs the solution of the MCDH problem, where the MCDH problem is defined over $(\mathbb{Z}_N, +)$. (Therefore, it directly implies that the DL problem is intractable, too.)

Proof. To bound the success probability of arbitrary generic algorithm \mathcal{A} , we consider another game between \mathcal{A} and a simulator \mathcal{O}_{sim} .

Simulator Description. \mathcal{O}_{sim} begins with choosing $N \stackrel{\$}{\leftarrow} T_{n,\delta}$ and setting variables $W, X_1W, \dots, X_{\kappa+1}W$ as the κ -MCDH problem instance. During the simulation, the simulator keeps a list $List$, which is initiated by row vectors $(0, W, 1), (-1, X_1W, 2), \dots, (-(\kappa+1), X_{\kappa+1}W, 2)$, where each vector is assigned for each variable of the MCDH instance, respectively. The first component is reserved for index, the second component is used for the assigned variable (or operations), and the third one is for degree of the corresponding polynomial. We use two notations P_i and $dg(i)$ to denote the corresponding multivariate polynomial and its degree, respectively.

As for queries, the simulator responds as follows.

- *Sampling.* \mathcal{O}_{sim} prepares a new variable A_i for the i -th sampling query and stores $(-(\kappa+1)-i, A_i, 1)$ in $List$.
- *Ring Operation.* Suppose that \mathcal{A} issues the k -th ring operation query (i, j, \circ) for $\circ \in \{+, \cdot\}$. If $\circ = +$, then \mathcal{O}_{sim} stores a vector $(k, (i, j, \circ), \max\{dg(i), dg(j)\})$. If $\circ = \cdot$, then \mathcal{O}_{sim} stores a vector $(k, (i, j, \circ), dg(i) + dg(j))$. If $dg(k) \geq 2^{\delta-\lambda}$, then \mathcal{O}_{sim} stops the simulation and outputs \perp .
- *Equality-Test.* Once \mathcal{O}_{sim} receives (i, j) from \mathcal{A} , \mathcal{O}_{sim} checks whether $P_i(A_1, \dots, A_\ell, W, X_1, \dots, X_{\kappa+1}) \stackrel{?}{=} P_j(A_1, \dots, A_\ell, W, X_1, \dots, X_{\kappa+1})$, where the number of all sampling queries before the present time is ℓ . This test can be done with overwhelming probability by choosing $\omega', x'_1, \dots, x'_{\kappa+1}$ and a'_1, \dots, a'_ℓ from \mathbb{Z}_N at random and checking $P_i(a'_1, \dots, a'_\ell, \omega', x'_1, \dots, x'_{\kappa+1}) = P_j(a'_1, \dots, a'_\ell, \omega', x'_1, \dots, x'_{\kappa+1}) \pmod{N}$. (The Schwartz-Zippel lemma exactly guarantees the overwhelming success probability of this test; if $P_i - P_j$ is a non-zero polynomial over \mathbb{Z}_N , then there exists a δ -bit prime p such that $p|N$ and $P_i - P_j \not\equiv 0 \pmod{p}$. Since the degree of $P_i - P_j$ is less than $2^{\delta-\lambda}$, the probability that a random point hits a root of $P_i - P_j$ over \mathbb{Z}_p is less than $\frac{2^{\delta-\lambda}}{p} < \frac{1}{2^\lambda}$.)

At the end of interaction, \mathcal{O}_{sim} chooses $\omega, x_1, \dots, a_1, \dots, a_L \stackrel{\$}{\leftarrow} \mathbb{Z}_N$, where L is the number of all sampling queries. If there exists an index i in the list $List$ such that $P_i(a_1, \dots, a_L, \omega, x_1, \dots, x_{\kappa+1}) = \omega^\kappa \prod_{j=1.. \kappa+1} x_j \pmod{N}$, then \mathcal{A} wins.

Analysis. In the adversarial view, the distribution of the above simulation and that of the original game is identical unless there are queries making two distinct polynomials P_i and P_j over \mathbb{Z}_N such that $P_i(a_1, \dots, a_L, \omega, x_1, \dots, x_{\kappa+1}) = P_j(a_1, \dots, a_L, \omega, x_1, \dots, x_{\kappa+1}) \pmod{N}$. We call such the event **Fail**. Therefore, \mathcal{A} 's success probability in the original game is bounded above by $\Pr[\text{Fail}] + \Pr[\mathcal{A} \text{ wins} | \neg \text{Fail}]$. Both $\Pr[\text{Fail}]$ and $\Pr[\mathcal{A} \text{ wins} | \neg \text{Fail}]$ are related to the probability that a randomly generated point hits a root of multivariate polynomial of degree less than $2^{\delta-\lambda}$ over \mathbb{Z}_p for some $p|N$, which is bounded above by a negligible function $\frac{1}{2^\lambda}$. This can be easily proven by well-known the Schwartz-Zippel lemma. Since we only consider polynomial time adversaries, the number of pairs of polynomials for the event **Fail** is bounded by polynomial in λ , so

that we conclude that the adversarial success probability in the original game should be negligible function in λ . \square

Remark 3. In the proof of Theorem 1, the modulus N needs not to be hidden. In fact, the reason why we consider the unknown modulus in the RES context is mainly due to the Cheon-Lee attack, which computes $g^{t^{\phi(N)-2}}$ for given a group generator g of order N and some element g^t . In the restricted version RES context, computing $t^{\phi(N)-2}$ from t is a large degree polynomial, so that it is not allowed adversaries to perform. Nevertheless, we require unknown modulus due to holding the SubM assumption.

Theorem 2 *Let $N_1 \stackrel{\$}{\leftarrow} T_{n,\delta}$ and $N_2 > 1$ be coprimes, where $n \geq 1$ and $\delta > \lambda$ for the security parameter λ . If arbitrary generic polynomial time algorithm computes only (possibly adaptively) polynomials of degree less than $2^{\delta-\lambda}$, then its advantage in solving the SubM problem is negligible, where the SubM problem is defined over $(\mathbb{Z}_{N_1} \times \{0\}, +)$, which is identified as the submodule of order N_1 in $(\mathbb{Z}_{N_1 N_2}, +)$.*

Proof. To bound the success probability of arbitrary generic algorithm \mathcal{A} , we consider another game between \mathcal{A} and a simulator \mathcal{O}_{sim} .

Simulator Description. \mathcal{O}_{sim} begins with setting variables X, Y, Z as the SubM problem instance. During the simulation, the simulator keeps a list *List*, which is initiated by row vectors $(0, X, 1), (-1, Y, 1), \dots, (-2, Z, 1)$, similarly to the simulator in the proof Theorem 1. We also use the same notations P_i and $dg(i)$ in the proof of Theorem 1.

As for queries, the simulator responds as follows.

- *Sampling.* For a sampling query from the whole module, \mathcal{O}_{sim} prepares a new variable A_i and stores $(-2-i, A_i, 1)$ in *List*. Similarly, for a sampling query from the submodule, \mathcal{O}_{sim} prepares a new variable B_i and stores $(-2-i, B_i, 1)$ in *List*.
- *Ring Operation.* Suppose that \mathcal{A} issues the k -th ring operation query (i, j, \circ) for $\circ \in \{+, \cdot\}$. If $\circ = +$, then \mathcal{O}_{sim} stores a vector $(k, (i, j, \circ), \max\{dg(i), dg(j)\})$. If $\circ = \cdot$, then \mathcal{O}_{sim} stores a vector $(k, (i, j, \circ), dg(i) + dg(j))$. If $dg(k) \geq 2^{\delta-\lambda}$, then \mathcal{O}_{sim} stops the simulation and outputs \perp .
- *Equality-Test.* Once \mathcal{O}_{sim} receives (i, j) from \mathcal{A} , \mathcal{O}_{sim} checks whether $P_i(A_1, \dots, B_1, \dots, X, Y, Z) \stackrel{?}{=} P_j(A_1, \dots, B_1, \dots, X, Y, Z)$, where the number of all sampling queries before the present time is ℓ . It can be done with overwhelming probability by choosing $N' \stackrel{\$}{\leftarrow} T_{n,\delta}$ and $x', y', z', a'_1, \dots, b'_1, \dots \stackrel{\$}{\leftarrow} \mathbb{Z}_{N'}$ and checking $P_i(a'_1, \dots, b'_1, \dots, x', y', z') = P_j(a'_1, \dots, b'_1, \dots, x', y', z') \pmod{N'}$. (More precisely, the detail about error probability of this test is given in Lemma 6.)

At the end of interaction, \mathcal{A} outputs her guess β' for whether Z indicates an element in the whole module or the submodule. Then, \mathcal{O}_{sim} chooses $N_1 \stackrel{\$}{\leftarrow} T_{n,\delta}$ and $N_2 > 1$ with condition $\gcd(N_1, N_2) = 1$ and $\beta \stackrel{\$}{\leftarrow} \{0, 1\}$, $x, a_1, \dots, a_L \stackrel{\$}{\leftarrow} \mathbb{Z}_{N_1 N_2}$, $y, b_1, \dots, b_{L'} \stackrel{\$}{\leftarrow} \mathbb{Z}_{N_1} \times \{0\}$, where L and L' are the number of all sampling queries for the whole module and the submodule, respectively and $\mathbb{Z}_{N_1} \times \{0\}$ is identified as the subring of order N_1 in $\mathbb{Z}_{N_1 N_2}$. In addition, if $\beta = 0$, $z \stackrel{\$}{\leftarrow} \mathbb{Z}_{N_1 N_2}$. Otherwise, $z \stackrel{\$}{\leftarrow} \mathbb{Z}_{N_1} \times \{0\}$. \mathcal{A} wins if and only if $\beta' = \beta$.

Analysis. In the adversarial view, the distribution of the above simulation and that of the original game is indistinguishable¹⁸ unless there are queries making two distinct polynomials P_i and P_j over \mathbb{Z} such that $P_i(a_1, \dots, b_1, \dots, x, y, z) = P_j(a_1, \dots, b_1, \dots, x, y, z) \pmod{N_1 N_2}$. We call such the event **Fail**. Unless the event **Fail** occurs, \mathcal{A} 's advantage is 0 since β is chosen at random after receiving β' .¹⁹ Therefore, \mathcal{A} 's advantage in the original game is bounded above by $\Pr[\text{Fail}]$. To compute an upper bound of $\Pr[\text{Fail}]$, we consider a larger event $\widetilde{\text{Fail}}$ that there exists a pair of two distinct polynomials P_i and P_j over \mathbb{Z} generated by \mathcal{A} such that the equality $P_i(a_1, \dots, b_1, \dots, x, y, z) = P_j(a_1, \dots, b_1, \dots, x, y, z)$ holds over \mathbb{Z}_{N_1} instead of $\mathbb{Z}_{N_1 N_2}$. Then, for this equality, we can consider $a_1, \dots, b_1, \dots, x, y, z$ as independent random integers over \mathbb{Z}_{N_1} . The following Lemma 6 shows that $\Pr[\widetilde{\text{Fail}}]$ is negligible in λ . \square

¹⁸ There could be negligible errors from polynomial number of equality test queries.

¹⁹ Again, we ignore negligible errors occurred during the equality test queries.

Lemma 6 Suppose that arbitrary generic polynomial time algorithm \mathcal{A} is given variables X_1, \dots, X_t . If \mathcal{A} generates a non-zero polynomial $f(X_1, \dots, X_t)$ in $\mathbb{Z}[X_1, \dots, X_t]$ of degree less than $2^{\delta-\lambda}$ by using generic ring operations from given variables X_1, \dots, X_t , then

$$\Pr_{\substack{N_1 \stackrel{\$}{\leftarrow} T_{n,\delta} \\ a_i \stackrel{\$}{\leftarrow} \mathbb{Z}_{N_1}}} [f(a_1, \dots, a_t) = 0 \pmod{N_1}]$$

is negligible in λ , where $n \geq 1$ and $\delta > \lambda$.

Proof. Ignoring negligible distance, it is sufficient to show that

$$\Pr_{\substack{p \stackrel{\$}{\leftarrow} \{\delta\text{-bit primes}\} \\ a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}}} [f(a_1, \dots, a_t) = 0 \pmod{p}]$$

is negligible in λ since N_1 is a product of δ -bit primes and the distribution of $a_i \pmod{p}$ is close to the uniform distribution in \mathbb{Z}_p , with negligible statistical distance. This probability is equal to

$$\begin{aligned} & \Pr_{a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}} [f(a_1, \dots, a_t) = 0] + \Pr_{a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}} [f(a_1, \dots, a_t) \neq 0] \\ & \cdot \Pr_{\substack{p \stackrel{\$}{\leftarrow} \{\delta\text{-bit primes}\} \\ a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}}} [f(a_1, \dots, a_t) = 0 \pmod{p} | f(a_1, \dots, a_t) \neq 0]. \end{aligned}$$

Since $\deg(f) < 2^{\delta-\lambda}$, $\Pr_{a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}} [f(a_1, \dots, a_t) = 0] < \frac{2^{\delta-\lambda}}{2^{\delta+\lambda}} = \frac{1}{2^{2\lambda}}$ is negligible. (We omit the details, which is almost the same as that of the Schwartz-Zippel lemma, though the Schwartz-Zippel lemma deals with only finite fields.)

Next, we show that

$$\Pr_{\substack{p \stackrel{\$}{\leftarrow} \{\delta\text{-bit primes}\} \\ a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}}} [f(a_1, \dots, a_t) = 0 \pmod{p} | f(a_1, \dots, a_t) \neq 0]$$

is negligible. Since we assume that \mathcal{A} is a generic PPT algorithm, the number of sum-gates, of fan-in 2, performed by \mathcal{A} is polynomially bounded, say q . Let $F_i = 2^{(2^{i-1})q + i(\delta+\lambda)}$ and $h(X_1, \dots, X_t)$ be a degree- i polynomial generated by \mathcal{A} . We first argue that $\max_{a_i \in \{0, \dots, 2^{\delta+\lambda}\}} \{|h(a_1, \dots, a_t)|\} < F_i$. We inductively shows that such the inequality holds for $i \geq 1$. From q sum gates, the maximum scalar \mathcal{A} can make is bounded above by 2^q . (Recall that \mathcal{A} begins with variables X_1, \dots, X_t .) Thus, $F_1 = 2^{q+\delta+\lambda} = 2^q \cdot 2^{\delta+\lambda}$ can be an upper bound for any degree-1 polynomial h . Suppose that the argument is true for $i = 1..k-1$. Let us consider arbitrary degree- k polynomial h , which is generated by computing a sum of products of lower-degree polynomials. Note that this is the all cases that \mathcal{A} can make a degree- k polynomial from lower degree polynomials generically. Let i_1, \dots, i_ℓ be positive integers such that $i_1 + \dots + i_\ell \leq k$. If g is an ℓ -product of polynomials of degree i_1, \dots, i_ℓ , then $\max_{a_i \in \{0, \dots, 2^{\delta+\lambda}\}} \{|g(a_1, \dots, a_t)|\}$ is bounded above by $2^{(2^{k-\ell})q + k(\delta+\lambda)} = \prod_{j \in \{i_1, \dots, i_\ell\}} 2^{(2^j-1)q + j(\delta+\lambda)}$ due to our hypothesis. From $\ell \geq 2$, we obtain the bound $2^{(2^{k-2})q + k(\delta+\lambda)}$. There are at most 2^q additions in the sum after performing at most q sum-gates, so that we obtain the desired bound $F_k = 2^{(2^{k-1})q + k(\delta+\lambda)}$ for arbitrary h of degree k .

From this bound of point evaluation of polynomials of degree at most $2^{\delta-\lambda}$, which is generated by \mathcal{A} , we know that for any non-zero polynomial $f(X_1, \dots, X_t)$ generated by \mathcal{A} and any a_i 's chosen from $\{0, \dots, 2^{\delta+\lambda}\}$, $f(a_1, \dots, a_t)$ has at most $(\frac{q}{\delta} + 1)2^{\delta-\lambda+1}$ distinct δ -bit prime factors. Since the number of all δ -bit primes is approximately $\frac{2^\delta}{2\delta}$ (from the prime number theorem) which is exponentially larger than $(\frac{q}{\delta} + 1)2^{\delta-\lambda+1}$, we conclude that $\Pr_{\substack{p \stackrel{\$}{\leftarrow} \{\delta\text{-bit primes}\} \\ a_i \stackrel{\$}{\leftarrow} \{0, \dots, 2^{\delta+\lambda}\}}} [f(a_1, \dots, a_t) = 0 \pmod{p} | f(a_1, \dots, a_t) \neq 0] \leq \frac{4(q+\delta)}{2^\lambda}$ is negligible. (Recall that both q and δ are polynomially bounded.) The theorem statement directly follows. \square

B One-Round Multipartite Diffie-Hellman Key Exchange from RES

We present a one-round N -way Diffie-Hellman key exchange protocol from RES.

Setup($1^\lambda, N$): Run $\text{InstGen}(1^\lambda, \mathcal{C}) \rightarrow (\mathbf{pp}, \mathbf{p}_{zt})$, where the permitted circuit \mathcal{C} contains monomials of degree N .

Publish(\mathbf{pp}, i): The i -th party \mathcal{P}_i samples $c_i \leftarrow \text{samp}(\mathbf{pp})$. Run $\text{enc}(\mathbf{pp}, c_i) \rightarrow c'_i$. Publish c'_i as \mathcal{P}_i 's public key and keep c_i as his secret key.

KeyGen($\mathbf{pp}, i, c_i, \{c'_j\}_{j \neq i}$): Run $(N-1)\text{-linear}(\mathbf{pp}, c'_1, \dots, c'_N) \rightarrow c'$ and $\text{mul}(c', c_i) \rightarrow \tilde{c}$. Then, extract the secret s by running $\text{ext}(\mathbf{pp}, \mathbf{p}_{zt}, \tilde{c}) \rightarrow s$.

In the proposed RES realization, we can set \mathcal{C} to contain any circuits of depth L and then the encoding size will be $\tilde{O}(L^2 \lambda^3)$ as in Table 1. Therefore, if we set $L = \lceil \log N \rceil$, then the encoding size will become polylogarithmic in the total number of participants.

It is straightforward that the above N -party Diffie-Hellman key exchange protocol is secure under the Ext-DDH assumption; that is, given $\mathbf{pp}, \mathbf{p}_{zt}$ and $\{c'_j\}_{j=1, \dots, N}$, the resulting secret s is indistinguishable from random bit-strings of equal length.

C Missing Parts in Proposed Scale-invariant Homomorphic Encryption

C.1 Missing Proofs

Proof of Lemma 2. (1) By definition, a fresh ciphertext c is equal to $\sum_{i \in S} x_i + y \cdot m - kx_0$ for some set S and $k \in [0, \tau + g]$, and so $c = (\sum_{i \in S} q_i + q_y m - kq_0)p^2 + \left\lfloor \frac{p}{g} \right\rfloor m + \sum_{i \in S} r_i + r_y m - kr_0$. Therefore, the absolute value of the noise of c is bounded above by $2(\tau + g)(2^\rho - 1)$.

(2) As for the addition, we have

$$\begin{aligned} [c_1 + c_2]_{x_0} &= (q_1 + q_2)p^2 + \left\lfloor \frac{p}{g} \right\rfloor ([m_1]_g + [m_2]_g + (r_1^* + r_2^*)g) + r_1 + r_2 - kx_0 \\ &= \left\lfloor \frac{p}{g} \right\rfloor ([m_1 + m_2]_g + (r_1^* + r_2^* + \delta)g) + r_1 + r_2 - kr_0 \pmod{p} \end{aligned}$$

for some $k, \delta \in [0, 1]$. Therefore, the noise $|r_1 + r_2 - kr_0| \leq |r_1 + r_2| + 2^\rho - 1$ and $|r_1^* + r_2^* + \delta| \leq |r_1^* + r_2^*| + 1$.

(3) We have

$$\begin{aligned} c_1 \cdot c_2 &= qp^2 + \left\lfloor \frac{p}{g} \right\rfloor^2 ([m_1]_g + r_1^*g)([m_2]_g + r_2^*g) \\ &\quad + \left\lfloor \frac{p}{g} \right\rfloor ([m_1]_g + r_1^*g)r_2 + ([m_2]_g + r_2^*g)r_1 + r_1r_2 \\ &= qp^2 + \left\lfloor \frac{p}{g} \right\rfloor^2 ([m_1m_2]_g + r^*g) + r \end{aligned}$$

with $|r^*| \leq g2^{2\rho^*+2}$ and $|r| \leq 2^{\rho^*+1+\eta}(|r_1| + |r_2|)$.

Let us consider $\langle \text{BD}(c_1 \cdot c_2), [\text{PT}(\frac{g^2}{p^2})]_g \rangle$. Since all entries in $\text{BD}(\cdot)$ are bits, it is congruent to $\langle \text{BD}(c_1 \cdot c_2), \text{PT}(\frac{g^2}{p^2}) \rangle$ modulo g , and by definition of BD and PT , $\langle \text{BD}(c_1 \cdot c_2), \text{PT}(\frac{g^2}{p^2}) \rangle = c_1c_2\frac{g^2}{p^2} = qg^2 + ([m_1m_2]_g + r^*g) + \epsilon$ with $|\epsilon| \leq g^32^{2\rho^*+2-\eta}(|r_1| + |r_2|)$. Therefore, we have

$$\langle \text{BD}(c_1 \cdot c_2), [\text{PT}(\frac{g^2}{p^2})]_g \rangle = q'g + [m_1m_2]_g + \epsilon$$

with $q' \leq 2\gamma$ and $|\epsilon| \leq g^32^{2\rho^*+2-\eta}(|r_1| + |r_2|)$. This implies that

$$\langle \text{BD}(c_1 \cdot c_2), \frac{p}{g} [\text{PT}(\frac{g^2}{p^2})]_g \rangle = \left\lfloor \frac{p}{g} \right\rfloor ([m_1m_2]_g + q'g) + r',$$

with $|r'| \leq \frac{1}{2}([m_1 m_2]_g + q'g) + \frac{p}{g}\epsilon \leq g^2 2^{2\rho^*+3}(|r_1| + |r_2|)$. Here, we assume that $\gamma \ll g^{2\rho^*}(|r_1| + |r_2|)$ and it holds in our parameter selection.

Next, we consider $\langle \text{BD}(c_1 c_2), \mathbf{x} \rangle$, where $\mathbf{x} = (x'_0, \dots, x'_{2\gamma-1})$ and $x'_i \leftarrow \mathcal{D}_{p, q_0}^\rho$. It is equal to $qp^2 + r$ for some q and r with $|r| \leq 2\gamma 2^\rho$.

Putting in all together, we have that the output of HE.Mult is equal to

$$\begin{aligned} [\langle \text{BD}(c_1 \cdot c_2), \mathbf{z} \rangle]_{x_0} &= [\langle \text{BD}(c_1 c_2), \mathbf{x} + \left\lfloor \frac{p}{g} \left[\text{PT} \left(\frac{g^2}{p^2} \right) \right]_g \right\rfloor \rangle]_{x_0} \\ &= qp^2 + \left\lfloor \frac{p}{g} \right\rfloor ([m_1 m_2]_g + q'g) + r'' \end{aligned}$$

with $|q'| \leq 2\gamma$ and $|r''| \leq g^2 2^{2\rho^*+4}(|r_1| + |r_2|)$. Here we assume that $2\gamma 2^{\rho+1} < g^2 2^{2\rho^*+3}(|r_1| + |r_2|)$ and it holds in our parameter selection. This completes the proof. \square

Proof of Theorem 3. For each $i = 0..L$, Let $c_i = q_i p^2 + \left\lfloor \frac{p}{g} \right\rfloor (m_i + r_i^* g) + r_i$ be a ciphertext after the evaluation of the i -th level gates. Let R_i and R_i^* be the bounds for $|r_i|$ and $|r_i^*|$, respectively. From 1 of Lemma 2, $R_0 \leq 2(\tau + g)(2^\rho - 1)$ and $R_0^* = 0$. 3 of Lemma 2 then implies $R_{i+1} < 2^{2\rho^*+5} g^2 R_i$ and $R_i^* < 2\gamma$. Thus we have $c_L = q_L p^2 + \left\lfloor \frac{p}{g} \right\rfloor (m_L + r_L^* g) + r_L$, where $R_L < (2^{2\rho^*+5} g^2)^L \cdot 2(\tau + g)(2^\rho - 1)$ and $R_L^* < 2\gamma$. By Lemma 3, it is required that $2\gamma g^2 + (2^{2\rho^*+5} g^2)^L \cdot 2g(\tau + g)(2^\rho - 1) < (2^{2\rho^*+5} g^2)^L g(\tau + g) 2^{\rho+2} \leq 2^{\eta-1}$ for the correct decryption of c_L .

C.2 Security Analysis

To prove the semantic security of our homomorphic encryption schemes, we follow the approach used in the security arguments in [20]; In [20], the semantic security of the proposed homomorphic encryptions are reduced to the *decisional approximate common divisors (DACD)* and the *n-decisional approximate common divisors (n-DACD)* assumptions [11]. The reduction given in [20] is quite straightforward since, informally, the assumptions just say that given the public key a random encryption of zero is indistinguishable from a random integer modulo x_0 if the number of encryptions of zero in public key is sufficient to apply the leftover hash lemma. To be accurate, both the instances in the DACD and the n -DACD assumptions do not contain the multiplication key of homomorphic encryption scheme, so that what the authors show is the proposed non-batch homomorphic scheme (and its batch version, resp.) *without the multiplication key* is semantically secure under the DACD (the n -DACD, resp.) assumption. For the semantic security of the full scheme containing the multiplication key, one can modify the DACD and n -DACD assumptions to contain the multiplication key in their problem instance. Or one can assume a ‘circular security’ of the restricted scheme; the multiplication key in the scale-invariant scheme has a form of a *fake* encryption of secret key bits. In the context of the homomorphic encryption schemes, it is common to assume the circular security [44, 11, 20, 15].

We first give natural generalizations of the DACD and the n -DACD assumption by using large message space instead of the binary field.

Definition 8 ((ρ, η, γ)-DACD [11]) Let p be a random odd integer of η bits, $q_0 \xleftarrow{\$} \mathbb{Z} \cap [0, 2^\gamma/p^2)$, $r_0 \xleftarrow{\$} \mathbb{Z} \cap (2^{-\rho}, 2^\rho)$. Given $x_0 = q_0 \cdot p^2 + r_0$, polynomially many samples from $\mathcal{D}_{p, q_0}^\rho$ and $y \leftarrow \mathcal{D}_{p, q_0}^\rho + \left\lfloor \frac{p}{g} \right\rfloor$, determine $b \in \{0, 1\}$ from $c = x + b \cdot r \bmod x_0$ where $x \leftarrow \mathcal{D}_{p, q_0}^\rho$ and $r \leftarrow [0, x_0) \cap \mathbb{Z}$.

Definition 9 ((ρ, η, γ)- n -DACD [11]) Let p_1, \dots, p_n be coprime random η -bit integers of product π , $q_0 \xleftarrow{\$} \mathbb{Z} \cap [0, 2^\gamma/\pi^2)$ that is coprime to p_j 's, $r_{0,j} \xleftarrow{\$} \mathbb{Z} \cap (2^{-\rho}, 2^\rho)$.

Given $x_0 = q_0 \cdot \pi^2 + \text{CRT}_{p_1^2, \dots, p_n^2}(r_{0,1}, \dots, r_{0,n})$ and polynomially many samples from $\mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$, and $y_j = y'_j + \left\lfloor \frac{p_j}{g_j} \right\rfloor \left(\left(\frac{\pi^2}{p_j^2} \right)^{-1} \bmod p_j^2 \right) \cdot \frac{\pi^2}{p_j^2}$ where $y'_j \leftarrow \mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$ for $j = 1..n$, determine $b \in \{0, 1\}$ from $c = x + b \cdot r \bmod x_0$ where $x \leftarrow \mathcal{D}_{p_1, \dots, p_n, q_0}^\rho$ and $r \leftarrow [0, x_0) \cap \mathbb{Z}$.

As commonly happens in the previous scale-invariant homomorphic encryption scheme [20, 15], to prove the semantic security of the full scheme, i.e., allowing multiplication, it suffices to include the public key for the multiplication \mathbf{z} in the above decisional assumptions. We call such assumptions DACD* and n -DACD*, respectively.²⁰

The following theorem shows that the semantic security of our schemes in Section 4.1 and Section 4.2. Note that using the standard hybrid argument, the proofs are straightforward, as aforementioned.

Theorem 4 *The scheme with the multiplication parameters \mathbf{z} given in Section 4.1 is semantically secure under the (ρ, η, γ) -DACD* assumption. The batch version with the parameters \mathbf{z} given in Section 4.2 is semantically secure under the (ρ, η, γ) - n -DACD* assumption.*

D Missing Parts in Proposed RES

D.1 Zero-testing Procedure

Given a level- κ encoding c of (m_1, \dots, m_n) , we can write

$$\begin{aligned} c &= \sum_{i=1}^n \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor ([w_i^\kappa m_i]_{g_i} + g_i r_i^*) + r_i \right) \cdot u_i - a \cdot \pi^2 \\ &= \sum_{i=1}^n \left(r_i^* p_i + \left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot u_i - a \cdot \pi^2 \end{aligned}$$

for some r_i^* 's, r_i 's, r_i' 's, and $a < n2^{2\eta}$, where $u_i = \left(\frac{\pi}{p_i}\right)^2 \left[\left(\frac{\pi}{p_i}\right)^{-2}\right]_{p_i^2}$. Letting $u_i = \left(\frac{\pi}{p_i}\right)^2 (e_i p_i + \left[\left(\frac{\pi}{p_i}\right)^{-2}\right]_{p_i})$ for some $0 \leq e_i < p_i$, we have

$$\begin{aligned} c &= \sum_{i=1}^n \left(r_i^* p_i + \left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \left(\frac{\pi}{p_i}\right)^2 (e_i p_i + \left[\left(\frac{\pi}{p_i}\right)^{-2}\right]_{p_i}) - a \cdot \pi^2 \\ &= \sum_{i=1}^n \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot u_i' + a' \cdot \pi, \end{aligned}$$

where $a' < n2^{2\eta+\gamma/2+2}$ and $u_i' = \left(\frac{\pi}{p_i}\right)^2 \left[\left(\frac{\pi}{p_i}\right)^{-2}\right]_{p_i}$. We then get

$$\begin{aligned} (\boldsymbol{\omega})_j &= (c \cdot \mathbf{p}_{zt} \bmod N)_j \\ &= \sum_{i=1}^n h_{ij} \alpha_i p_i^{-1} \cdot \left(\sum_{k=1}^n \left(\left\lfloor \frac{p_k}{g_k} \right\rfloor [w_k^\kappa m_k]_{g_k} + r_k' \right) \cdot u_k' + a' \cdot \pi \right) \bmod N \\ &= \sum_{i=1}^n h_{ij} \cdot \left(\left(\left\lfloor \frac{p_i}{g_i} \right\rfloor [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \beta_i \right. \\ &\quad \left. + \sum_{k=1, k \neq i}^n \left(\left\lfloor \frac{p_k}{g_k} \right\rfloor [w_k^\kappa m_k]_{g_k} + r_k' \right) \cdot \alpha_i \cdot \frac{u_k'}{p_i} + a' \cdot \alpha_i \cdot \frac{\pi}{p_i} \right) \bmod N \end{aligned}$$

We argue that if $m_i = 0$ for all $i = 1..n$, then ω_j is sufficiently smaller than N . We have that $\beta_i, \alpha_i \cdot \frac{u_k'}{p_i}$, and $a' \cdot \alpha_i \cdot \frac{\pi}{p_i}$ have size at most $|N| - \eta + 2$, $(\eta - 1) + (\gamma - \eta) - \eta = \gamma - \eta - 1$, $\log_2 n + (2\eta + \gamma/2) + (\eta - 1) + (\gamma/2 - \eta) = \log_2 n + \gamma + 2\eta - 1$ bits, respectively. Therefore, if m_i 's are zero and r_i 's are sufficiently small, each component in the right-hand side is small compared to N , which is $\gamma + 4\eta + 1$ bits. Furthermore, we can show that if $m_i \neq 0$ for some i , then $\|\boldsymbol{\omega}\|_\infty$ must be large due to the term $\sum_{i=1}^n h_{ij} \cdot \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \beta_i$, which is larger than N . We provide the following lemma.

Lemma 7 *Let all parameters are selected according to our parameter setting. Let $3 + 2\log_2 n \leq \rho_f \leq \eta - \alpha - 2\beta - \lambda - 9$ and $\nu = \eta - \rho_f - \beta - \lambda - 4 \geq \alpha + \beta + 5$. Let c be a level- κ encoding of message $\mathbf{m} = (m_1, \dots, m_n)$ such that $c = \left\lfloor \frac{p_i}{g_i} \right\rfloor ([w_i^\kappa m_i]_{g_i} + g_i r_i^*) + r_i \pmod{p_i^2}$ for all $i = 1..n$. Assume that $\|\mathbf{r}^*\|_\infty, \|\mathbf{r}\|_\infty < 2^{\rho_f}$, where $\mathbf{r}^* = (g_n r_n^*, \dots, g_1 r_1^*), \mathbf{r} = (r_1, \dots, r_n)$. If $\mathbf{m} = \mathbf{0}$ then $\|\boldsymbol{\omega}\|_\infty < 2^{-\nu-\lambda} \cdot N$. Conversely, if $\mathbf{m} \neq \mathbf{0}$ then $\|\boldsymbol{\omega}\|_\infty > 2^{-\nu+2} \cdot N$.*

²⁰ Or we can use a circular security assumption as in the previous homomorphic encryption schemes [44, 11, 20, 15].

Proof. We begin with a level- κ encoding c of a message $\mathbf{m} = (m_1, \dots, m_n)$, which can be written as

$$\begin{aligned} c &= \sum_{i=1}^n \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor ([w_i^\kappa m_i]_{g_i} + g_i r_i^*) + r_i \right) \cdot u_i - a \cdot \pi^2 \\ &= \sum_{i=1}^n \left(r_i^* p_i + \left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot u_i - a \cdot \pi^2 \\ &= \sum_{i=1}^n \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot u_i' + a' \cdot \pi \end{aligned}$$

for some r_i^* 's, r_i 's, r_i' 's, $a < n2^{2\eta}$, and $a' < n2^{2\eta+\gamma/2+2}$, where $u_i = (\frac{\pi}{p_i})^2 [(\frac{\pi}{p_i})^{-2}]_{p_i^2}$ and $u_i' = (\frac{\pi}{p_i})^2 [(\frac{\pi}{p_i})^{-2}]_{p_i}$. Here, we have $|r_i'| < |r_i| + |g_i r_i^*|$, so that $\|\mathbf{r}'\|_\infty < 2^{\rho_f+1}$, where $\mathbf{r}' = (r_1', \dots, r_n')$.

Let us consider each component in $(\boldsymbol{\omega})_j$, which is equal to

$$\sum_{i=1}^n h_{ij} \cdot \left(\left(\left\lfloor \frac{p_i}{g_i} \right\rfloor [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \beta_i + \sum_{\substack{k=1, \\ k \neq i}}^n \left(\left\lfloor \frac{p_k}{g_k} \right\rfloor [w_k^\kappa m_k]_{g_k} + r_k' \right) \cdot \alpha_i \cdot \frac{u_k'}{p_i} + a' \cdot \alpha_i \cdot \frac{\pi}{p_i} \right) \bmod N.$$

Let $\mathbf{s} = (s_i)_{i=1..n}$ and $\mathbf{t} = (t_i)_{i=1..n}$ such that

$$\begin{aligned} s_i &= \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \beta_i \bmod N, \\ t_i &= \sum_{\substack{k=1, \\ k \neq i}}^n \left(\left\lfloor \frac{p_k}{g_k} \right\rfloor [w_k^\kappa m_k]_{g_k} + r_k' \right) \cdot \alpha_i \cdot \frac{u_k'}{p_i} + a' \cdot \alpha_i \cdot \frac{\pi}{p_i} \bmod N. \end{aligned}$$

Here, we use the centered modular reduction, so that $|x \bmod N| \leq |x|$ for all $x \in \mathbb{Z}$. For $i = 1..n$, if $\mathbf{m} = 0$, then we have

$$\begin{aligned} |s_i| &= |r_i' \cdot \beta_i \bmod N| \leq 2^{\rho_f - \eta + 3} \cdot N. \\ |t_i| &= \left| \sum_{\substack{k=1, \\ k \neq i}}^n r_k' \cdot \alpha_i \cdot \frac{u_k'}{p_i} + a' \cdot \alpha_i \cdot \frac{\pi}{p_i} \bmod N \right| \leq n2^{\rho_f + \gamma - \eta + 2} + n2^{\gamma + 2\eta + 2} \end{aligned}$$

Using $3 + \log_2 n \leq \rho_f < \eta$, $|t_i|$ is upper-bounded by $n2^{\gamma+2\eta} \leq 2^{\rho_f - \eta + 3} \cdot N$. Therefore, $\|\mathbf{s} + \mathbf{t}\|_\infty \leq 2^{\rho_f - \eta + 4} \cdot N$

From this bound, we have that $\|\boldsymbol{\omega}\|_\infty = \|\mathbf{H}^T(\mathbf{s} + \mathbf{t}) \bmod N\|_\infty$ is upper-bounded by

$$\|\mathbf{H}^T(\mathbf{s} + \mathbf{t})\|_\infty \leq \|\mathbf{H}^T\|_\infty \|\mathbf{s} + \mathbf{t}\|_\infty < 2^{\beta + \rho_f - \eta + 4} \cdot N.$$

Using $\nu = \eta - \rho_f - \beta - \lambda - 4$, we obtain the above is equal to $2^{-\nu - \lambda} \cdot N$, as in the lemma.

Next, we show that if $\mathbf{m} \neq 0$, that is $m_i \neq 0$ for some i , $\|\boldsymbol{\omega}\|_\infty > 2^{\nu+2} \cdot N$. To this end, we show that $\|\mathbf{s}\|_\infty$ is sufficiently large. Suppose that $\mathbf{m} \neq 0$. Consider

$$s_i = \left(\left\lfloor \frac{p_i}{g_i} \right\rfloor [w_i^\kappa m_i]_{g_i} + r_i' \right) \cdot \beta_i \bmod N.$$

$\beta_i = \alpha_i \cdot (u_i'/p_i) \bmod N$ and the following lemma implies that $2^{-\eta-1} \cdot N < |\beta_i|$.

Lemma 8 (Appendix D, [18]) *For any integers a, b, m such that $|b| < m/2$, $\gcd(a, m) = 1$ and $a \neq 0$ does not divide b , if $x = b/a \bmod m$, then $|x| \geq \frac{m}{2|a|}$.*

Therefore,

$$|s_i| \geq \left\lfloor \frac{p_i}{g_i} \right\rfloor \cdot \beta_i > 2^{\eta - \alpha - 1} 2^{-\eta - 1} \cdot N = 2^{-\alpha - 2} \cdot N$$

so that $\|\mathbf{s}\|_\infty \geq 2^{-\alpha - 2} \cdot N$ and

$$\|\mathbf{s} + \mathbf{t}\|_\infty \geq \|\mathbf{s}\|_\infty - \|\mathbf{t}\|_\infty > 2^{-\alpha - 3} \cdot N.$$

Finally, we have, as required,

$$\begin{aligned} \|\boldsymbol{\omega}\|_\infty &\geq \frac{\|(\mathbf{H}^T)^{-1} \boldsymbol{\omega}\|_\infty}{\|(\mathbf{H}^T)^{-1}\|_\infty} \geq \frac{\|(\mathbf{H}^T)^{-1} \boldsymbol{\omega} \bmod N\|_\infty}{\|(\mathbf{H}^T)^{-1}\|_\infty} = \frac{\|\mathbf{s} + \mathbf{t}\|_\infty}{\|(\mathbf{H}^T)^{-1}\|_\infty} \\ &> 2^{-\alpha - \beta - 3} \cdot N \geq 2^{-\nu + 2} \cdot N. \end{aligned}$$

□

D.2 Generation of H Matrix

To generate a random matrix $\mathbf{H} \in \mathbb{Z}^{n \times n}$ satisfying two bounds $\|\mathbf{H}^T\|_\infty \leq 2^\beta$ and $\|(\mathbf{H}^{-1})^T\|_\infty \leq 2^\beta$, we use the algorithm proposed by Coron *et al.* [18], which revises the original algorithm [19] to depend the weakness pointed out by Lee and Seo [39]. We explain the Coron *et al.*'s algorithm.

For any matrix $A \in \text{Mat}_{\lfloor n/2 \rfloor \times \lceil n/2 \rceil}(\{-1, 0, 1\})$, we define $\mathbf{H}_A \in \mathbb{Z}^{n \times n}$ as $\mathbf{H}_A = \begin{pmatrix} I_{\lfloor n/2 \rfloor} & A \\ 0 & I_{\lceil n/2 \rceil} \end{pmatrix}$. Then, \mathbf{H} is generated by a product of \mathbf{H}_A 's and \mathbf{H}_A^T 's; we compute alternatively \mathbf{H}_A and \mathbf{H}_A^T , so that \mathbf{H} has of the form $\mathbf{H} = \prod_{i=1}^{\beta'} \mathbf{H}_i$, where for odd i 's $\mathbf{H}_i = \mathbf{H}_A$, for even i 's $\mathbf{H}_i = \mathbf{H}_A^T$, and β' is decided as follows: we can keep the norm of the product and if such the norm exceeds $2^\beta/(1 + \lceil n/2 \rceil)$, then stop multiplying. Note that the norm of both matrix \mathbf{H}_A and its transpose are bounded by $1 + \lceil n/2 \rceil$, so that β' is decided to maximize the norms of both \mathbf{H} and \mathbf{H}^T with the upper bound 2^β .

D.3 Extraction

Our extraction algorithm is the same as that of Coron *et al.*'s [18]. For completeness, we recall the extraction algorithm in [18].

Given an encoding c , we extract a message-only-dependent value in the sense that the resulting value is independent from the randomness used in c . The algorithm is simple; (1) Multiply c by \mathbf{p}_{zt} modulo N , (2) Collect the ν most significant bits of each of the n components of the resulting vector, (3) by using the seed s in pp , apply a strong randomness extractor. That is, $\text{Extract}_s(\text{msbs}_\nu(c \cdot \mathbf{p}_{zt} \bmod N))$, where msbs_ν takes the ν most significant bits of the result.

Lemma 7 guarantees that for two encodings of the same message ext outputs the same result, and for two encodings of different message it always outputs different result. More precisely,

$$\|(c - c') \cdot \mathbf{p}_{zt} \bmod N\|_\infty \begin{cases} < 2^{-\nu-\lambda} \cdot N & \text{if } c \text{ and } c' \text{ encode} \\ & \text{the same } \mathbf{m}, \\ > 2^{\nu+2} \cdot N & \text{otherwise.} \end{cases}$$

This implies that $\text{msbs}_\nu(c \cdot \mathbf{p}_{zt} \bmod N)$ has the min-entropy at least $\log_2 |\mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}| \geq n(\alpha - 1)$. Therefore, a strong randomness extractor can extract a nearly uniform bit-string of length $\lfloor \log_2 |\mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}| \rfloor - \lambda$.

E Security Analysis of RES

E.1 Review Zeroizing Attacks

We first describe how the homomorphic multiplication on encodings and zero-testing procedure in the CLT scheme [19] work. Let p_1, \dots, p_n be secret key and $\pi = \prod_i p_i$ be public key. Let $c_1 = \text{CRT}_{p_i}(r_{1i} \cdot g_i + m_{1i})/z^s \bmod \pi$, $c_2 = \text{CRT}_{p_i}(r_{2i} \cdot g_i + m_{2i})/z^t \bmod \pi$ be encodings of $\mathbf{m}_1 = (m_{1i})_i, \mathbf{m}_2 = (m_{2i})_i$ with level s, t , respectively. The homomorphic multiplication on c_1 and c_2 is just $c_3 = [c_1 \cdot c_2]_\pi$, and so we have the $(s + t)$ -level encoding $c_3 = \text{CRT}_{p_i}(r_{3i} \cdot g_i + m_{1i} \cdot m_{2i})/z^{s+t} \bmod \pi$ for $r_{3i} = r_{1i} \cdot r_{2i} \cdot g_i + r_{1i} \cdot m_{2i} + r_{2i} \cdot m_{1i}$.

For a κ -level encoding $c = \text{enc}(\mathbf{m}) = \text{CRT}_{p_i}(r_i \cdot g_i + m_i)/z^\kappa \pmod{\pi}$, the zero-testing procedure is done by computing $[p_{zt} \cdot c]_\pi$, where $p_{zt} = \sum_i h_i \cdot ([z^\kappa \cdot g_i^{-1}]_{p_i}) \cdot \prod_{j \neq i} p_j \bmod \pi$. We observe that

$$w = [p_{zt} \cdot c]_\pi = \sum_{i=1}^n h_i \cdot (r_i + m_i \cdot (g_i^{-1} \bmod p_i)) \cdot \prod_{j \neq i} p_j \bmod \pi.$$

Therefore if $m_i = 0$ for all $i = 1..n$, since h_i 's r_i 's are small compared to p_i , we obtain that w is relatively small compared to π and so $w = \text{CRT}_{p_i}(h_i \cdot r_i \cdot \hat{p}_i) < \pi$ for $\hat{p}_i = \prod_{j \neq i} p_j$. This enables us to test whether c is an encoding of zero or not.

With Low-Level Zeroes. Recently, Cheon et al. [12] proposed a zeroizing attack to the Coron et al. GES [19]. Using low-level encodings of zero, they describe an attack that expresses the zero-testing procedure as a diagonal quadratic form in the CRT component of encodings over \mathbb{Z} to recover all p_i 's in polynomial time from $\pi = \prod_{i=1}^n p_i$.

We briefly describe the Cheon et al. attack to the original CLT scheme [19] that allows κ -level multilinearity. For given s -level encodings $c^{(b)}$ for $b \in \{0, 1\}$, let $A = \{a_1, \dots, a_n\}$ be the set of t -level of encodings and $B = \{b_1, \dots, b_n\}$ be the set of $(\kappa - t - s)$ -level encodings of zero for $1 \leq s, t \leq \kappa$. We note that the set A and B can be derived from 1-level encodings of zero and one. Let $a_j = \text{CRT}_{p_i}(a_{ji})/z^t \bmod \pi$, $b_k = \text{CRT}_{p_i}(b_{ki} \cdot g_i)/z^{\kappa-t-s} \bmod \pi$ and $c^{(b)} = \text{CRT}_{p_i}(c_i^{(b)})/z^s \bmod \pi$, where $a_{ji}, b_{ki}, c_i^{(b)} \in \mathbb{Z}_{p_i}$ for all $i, j, k \in [1, n]$.²¹ Then $a_j \cdot c^{(b)} \cdot b_k = \text{CRT}_{p_i}(a_{ji} \cdot c_i^{(b)} \cdot b_{ki} \cdot g_i)/z^\kappa \bmod \pi$ is a κ -level encodings of zero and we have

$$\begin{aligned} d_{jk}^{(b)} &= p_{zt} \cdot (a_j \cdot c^{(b)} \cdot b_k) \quad \bmod \pi \\ &= \text{CRT}_{p_i}(a_{ji} \cdot c_i^{(b)} \cdot h_i \cdot b_{ki}) \quad \bmod \pi \end{aligned} \quad (2)$$

from $p_{zt} = \text{CRT}_{p_i}(h_i \cdot g_i^{-1}) \cdot z^\kappa \bmod \pi$. Since $a_j \cdot c^{(b)} \cdot b_k$ is an encoding of zero, the last equality in (2) holds over the integers. Hence, we have a quadratic form in the CRT component over \mathbb{Z} ,

$$d_{jk}^{(b)} = \mathbf{a}_j \cdot \text{diag}(c_1^{(b)} u_1, \dots, c_n^{(b)} u_n) \cdot \mathbf{b}_k^T$$

where $\mathbf{a}_j = (a_{j1}, \dots, a_{jn})$, $\mathbf{b}_k = (b_{k1}, \dots, b_{kn})$ and $u_i = h_i \cdot \frac{\pi}{p_i} \cdot ((\frac{\pi}{p_i})^{-1} \bmod p_i)$. We can write $D_b = (d_{jk}^{(b)})$ of the form

$$D_b = X_A \cdot \text{diag}(c_1^{(b)} u_1, \dots, c_n^{(b)} u_n) \cdot X_B \in \mathbb{Z}^{n \times n} \quad (3)$$

over the integers for $(n \times n)$ matrices X_A, X_B whose j -th row and k -th column are \mathbf{a}_j and \mathbf{b}_k^T , respectively. By computing eigenvalues of $D_0 \cdot D_1^{-1}$, we can recover all p_i in polynomial time.

Without Low-Level Zeroes. Cheon et al. attack depends on successful zero-testing on top-level zero encodings. They assume the adversary can access low-level encodings of zero. On the other hand, some applications that do not provide low-level encodings such as indistinguishability obfuscations [26, 3].

Coron, Gentry, Halevi, Lepoint, Maji, Miles, Raykova, Sahai and Tibouchi extend Cheon et al. attack by considering several methods to make top-level zero encodings without low-level encodings of zero [17]. One of them is to find encoding sets such that they share the zero in CRT component. That is, we consider the case such that $a \cdot c \cdot b$ is a level- κ encoding of zero, even though a, b and c are non-zero encodings. In this case, we have the same quadratic form with Cheon et al. attack. The other way is to find multinomial on encodings to have top-level zero encodings. In this approach, since each monomial encoding may not be an encoding of zero, some g_i^{-1} 's are remained in diagonal matrix, which yields a rational quadratic form of CRT components. However it does not matter because g_i^{-1} factors fall off when computing $D_0 \cdot D_1^{-1}$.

We describe only different things of their attacks using a concrete example. For non-zero encodings a, a', b, b', c and c' , we assume that $(a \cdot c \cdot b + a' \cdot c' \cdot b')$ is a top-level zero encoding. Here, we denote $(\mathbf{e} || \mathbf{e}')$ by concatenation of two vectors \mathbf{e} and \mathbf{e}' . Since $(a \cdot c \cdot b + a' \cdot c' \cdot b')$ is an encoding of zero, the value $d = p_{zt} \cdot (a \cdot c \cdot b + a' \cdot c' \cdot b') \bmod \pi$ can be represented over rational without modulo reduction as follows:

$$d = (\mathbf{a} || \mathbf{a}') \cdot \text{diag}(c_1 u_1 / g_1, \dots, c_n u_n / g_n || c'_1 u_1 / g_1, \dots, c'_n u_n / g_n) \cdot (\mathbf{b} || \mathbf{b}')^T \in \mathbb{Q}$$

where $u_i = h_i \cdot \frac{\pi}{p_i} \cdot ((\frac{\pi}{p_i})^{-1} \bmod p_i)$, $\mathbf{a} = ([z^{t_a} \cdot a]_{p_i})_i$, $\mathbf{b} = ([z^{t_b} \cdot b]_{p_i})_i$ and \mathbf{a}', \mathbf{b}' are defined similarly for $i \in [1, n]$.²² If we consider sets $A = \{a_j\}_j, A' = \{a'_j\}_j, B = \{b_k\}_k, B' = \{b'_k\}_k, C = \{c^{(0)}, c^{(1)}\}$ and $C' = \{c'^{(0)}, c'^{(1)}\}$ such that $a_j \cdot c^{(b)} \cdot b_k + a'_j \cdot c'^{(b)} \cdot b'_k$ is an encoding of zero for each $j, k \in [1, 2n]$, we have $D_b = (d_{jk}^{(b)})$ for $d_{jk}^{(b)} = p_{zt} \cdot (a_j \cdot c^{(b)} \cdot b_k + a'_j \cdot c'^{(b)} \cdot b'_k) \bmod \pi$ as follows:

$$D_b = X_{A,A'} \cdot \text{diag}(c_1^{(b)} u_1 / g_1, \dots, c_n^{(b)} u_n / g_n || c_1'^{(b)} u_1 / g_1, \dots, c_n'^{(b)} u_n / g_n) \cdot X_{B,B'} \in \mathbb{Q}^{2n \times 2n},$$

²¹ Here, we simply denote $\text{CRT}_{p_i}(a_i)$ by the unique element in $\mathbb{Z}_{\prod p_i}$ that is congruent to a_i modulo p_i for all i .

²² Here, t_a and t_b denote the level of encodings a and b , respectively.

where $X_{A,A}, X_{B,B'}$ are $(2n \times 2n)$ integer matrices whose j -th row and k -th column are $(\mathbf{a}_j || \mathbf{a}'_j)$ and $(\mathbf{b}_k || \mathbf{b}'_k)^T$, respectively. Since each $a_j \cdot c^{(b)} \cdot b_k$ and $a'_j \cdot c'^{(b)} \cdot b'_k$ may not be zero encoding, g_i^{-1} factors are remained in the diagonal matrix, but they are removed when computing $D_0 \cdot D_1^{-1}$.

E.2 Lattice Attacks

We consider lattices attacks on level-0 encodings, public key including zero-testing key and multiplication key.

Level-0 Encodings. The level-0 encodings in our construction are instances of approximate GCD problem. There are several lattice attacks in [44, 21, 22] and we can apply the same argument except we use p_i^2 instead of p_i . One of the lattice attacks to level-0 encodings given in [19] is based on computing a short vector which is orthogonal to a vector $\mathbf{x} = (x_1, \dots, x_t)$ modulo x_0 , where the all x_i 's are level-0 encodings of zero. If the reduced vector is short enough, the error $\mathbf{r}_i \equiv \mathbf{x} \pmod{p_i}$ can be recovered and so the secret key p_i 's are revealed. We have the constraint on parameters $\gamma = (2\eta)^2 \cdot \omega(\log \lambda)$ as in [19]. From $\gamma = n \cdot \eta$, we choose $n = (\eta/4) \cdot \omega(\log \lambda)$ to defeat the attack. For more attacks and details, refer [44].

Zero-Testing Parameter. The zero-testing parameter \mathbf{p}_{zt} is almost same with [18]. When choosing parameters α_i and β_i , we have the same constraint on size, $|\alpha_i| < 2^{\eta-1}$, $|\beta_i| < 2^{2-\eta}N$ and different constraint on their relation, $\beta_i = \alpha \cdot (u'_i/p_i) \pmod{N}$ where $u'_i = (\frac{\pi}{p_i})^2 [(\frac{\pi}{p_i})^{-2}]_{p_i^2}$ from [18]. We also use 2η -bit larger modulo integer N . The lattice attack on the zero parameter in [18] only depends on the size of α, β, N and p_i 's, we can consider similar attack to our construction, including, the hidden subset sum attack, the Coppersmith attack and the inverse zero-testing matrix attack [18].

The zero-testing parameter $\mathbf{p}_{zt} = \sum_{i=1}^n \mathbf{h}_i \cdot \alpha_i \cdot p_i^{-1} \pmod{N}$ is a linear combination form of secret vectors \mathbf{h}_i . So we can think similar approach to the hidden subset sum attack [41] to find vectors \mathbf{h}_i . For any vector \mathbf{u} in $L_{\mathbf{p}_{zt}}^\perp$, it satisfies $\sum_{i=1}^n \langle \mathbf{h}_i, \mathbf{u} \rangle \cdot \alpha_i \cdot p_i^{-1} = 0 \pmod{N}$, which yields $L_{\mathbf{p}_{zt}}^\perp \subset L_{\mathbf{a}}^\perp$ for $\mathbf{a} = (a_1/p_1, \dots, a_n/p_n)$. We hope that the vector \mathbf{u} is short enough that it must be orthogonal to \mathbf{h}_i for each i . That is, if \mathbf{u} is short to ensure that $\mathbf{v} := (\langle \mathbf{h}_i, \mathbf{u} \rangle)_i$ is shorter than $\lambda_1(L_{\mathbf{a}}^\perp)$, we have $\mathbf{v} = \mathbf{0}$. If we can find sufficiently many such \mathbf{u} , the hidden vector \mathbf{h}_i can be recovered. On the other hand, we can not guarantee the existence of \mathbf{u} , since the size of vector \mathbf{v} is about $2^\beta \cdot \lambda_1(L_{\mathbf{p}_{zt}}^\perp)$ which is much larger than $\lambda_1(L_{\mathbf{a}}^\perp)$. We note that $\lambda_1(L_{\mathbf{p}_{zt}}^\perp)$ and $\lambda_1(L_{\mathbf{a}}^\perp)$ are both $N^{1/n}$. Therefore, it is hard to apply hidden subset sum attack to our construction.

Both the Coppersmith attack and computing the inverse zero testing matrix are also thwarted by our choice of parameters using almost same argument in [18]. We omit the detail. See [18].

Zero-Testing on Encodings. Even though several attacks on the zero testing parameter \mathbf{p}_{zt} and encodings are failed by our choice of parameters, the secret information may be extracted during zero-testing procedure. Let $\mathbf{w} = \mathbf{p}_{zt} \cdot c \pmod{N}$ be a resulting vector of zero-testing procedure on an encoding c . Then,

$$\begin{aligned} (\omega)_j &= (c \cdot \mathbf{p}_{zt} \pmod{N})_j \\ &= \sum_{i=1}^n h_{ij} \alpha_i p_i^{-1} \cdot \left(\sum_{k=1}^n \left(\left[\frac{p_k}{g_k} \right] [w_k^k m_k]_{g_k} + r'_k \right) \cdot u'_k + a' \cdot \pi \right) \pmod{N} \end{aligned}$$

We can consider the hidden subset sum attack to find hidden vector \mathbf{h}_i 's as the above attack. The attack is also thwarted by the same reason.

Multiplication Key. If we assume circular security, the multiplication key \mathbf{z} does not affect the security of our FHE scheme. In multilinear map, on the other hand, the secret information may be leaked when manipulating the multiplication key with the zero-testing parameter \mathbf{p}_{zt} . Since the multiplication key resembles an encodings $([g_1^2 2^i / p_1^2]_{g_1}, \dots, [g_n^2 2^i / p_n^2]_{g_n})$, there is no plausible attack on them as discussed above.

λ	ρ	$\log_2 g_i$	η	n	γ	τ
20	20	40	1018	5	10180	10220
30	30	60	1508	15	45240	45300
40	40	80	1998	52	207792	207872

Table 2. Parameter selection for implementation of BHE

F Noise Growth of BHE

We implement BHE for three security parameters $\lambda = 20, 30, 40$. For experiment we set $L = 7$ and $\log_2 g_i = 2\lambda$ as in Theorem 1 and 2. We present other parameters in Table 2 for each security level; in particular, $\eta = \rho + L(2\rho + 2\log_2 g_i + 5) + 3\log_2 g_i + 3$ according to Theorem 3 and γ is chosen according to the methodology suggested by Coron et al. in [21, Section 6.2], which derives parameters resistant to lattice-based attacks.

In the 1st experiment, we investigate noise size of multiplication of 2^i BHE-ciphertexts for $i = 0..7$, which corresponds to the evaluation of monomials of degree 2^i over BHE-ciphertexts. We compute them from trees of depth i ; Input values (fresh ciphertexts) are assigned to leaves. Each internal node is multiplication of two BHE-ciphertexts located at its child nodes. The root node (output), then, is a BHE-ciphertext of multiplication of 2^i ciphertexts in leaves. We call a resulted BHE-ciphertext from multiplication of 2^i ciphertexts in this way a BHE-ciphertext of degree 2^i . We investigate bit-length of noise added in a BHE-ciphertext of each root node in Figure 2 for $\lambda = 20, 30, 40$. Each number beside points on the line presents average of noises in slots of a BHE-ciphertext at the root node. Points without numbers are estimated noises of BHE-ciphertexts of degree 2^i for $i > 7$ from previous noises. A Horizontal dotted-line is the noise bound of each security level λ at $L = 7$. Figure 2 shows that the bit-length of noise in a BHE-ciphertext of degree 2^i increases almost linearly in i . A bit-length of fresh BHE-ciphertext (degree 1) is about $\log_2(\tau + g_i) + \rho$, where τ is set to $\gamma + 2\lambda$. The difference of bit-length of noises between two BHE-ciphertexts of degree 2^i and 2^{i-1} is about $\log_2 g_i + \log_2 \gamma$. Note that at each security level λ , it is not allowed to compute BHE-ciphertexts of degree 2^λ .

In the 2nd experiment we evaluate more complicated polynomial over ciphertext. We consider a circuit to compute polynomials of degree 128; Polynomials are computed via tree of depth 7 as follows: We prepare 4 trees of depth 7 to compute monomials of degree 128. We set fresh BHE-ciphertexts of random messages to leaves of each tree and compute multiplications of BHE-ciphertexts as in the 1st experiment. We then construct a new tree as follows: At each depth $7 - i$ in the tree for $i = 0..7$, the individual nodes are 1) added by 4 BHE-ciphertexts at the same position in 4 precomputed trees and 2) multiplied by a ciphertext at a neighborhood node, which results in a sum of BHE-ciphertexts of degree 2^i . We investigate average bit-length of noises added in BHE-ciphertexts at each depth in the constructed tree in Figure 3. Figure 3 shows noises and their behavior are quite similar with the 1st experiment. We also note that in this experiment it is not allowed to compute BHE-ciphertexts of degree 2^λ .

Note that both Theorem 1 and 2 assume that an adversary can compute polynomials of degree at most $2^{\delta-\lambda}$, where $\delta = \log_2 g_i = 2\lambda$. According to Figure 2 and 3, if we set $\log_2 g_i = 2\lambda$, $L = c\lambda$ for properly chosen $0 < c < 1$ ($c = 0.125$ for our experiments), then we expect that BHE no longer permits circuits to compute polynomials of degree 2^λ . Therefore, our BHE seems to satisfy conditions for restricted RES.

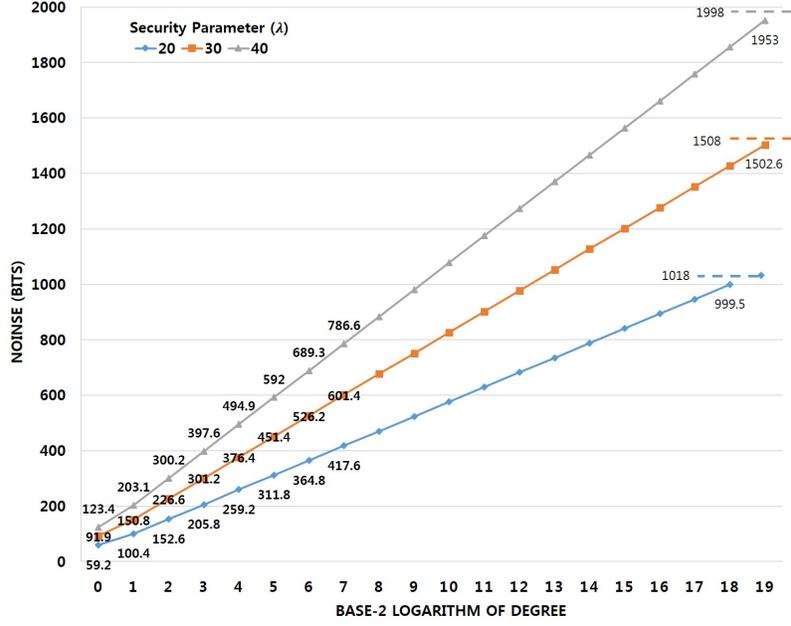


Fig. 2. Noise increase of Monomial with respect to Degree

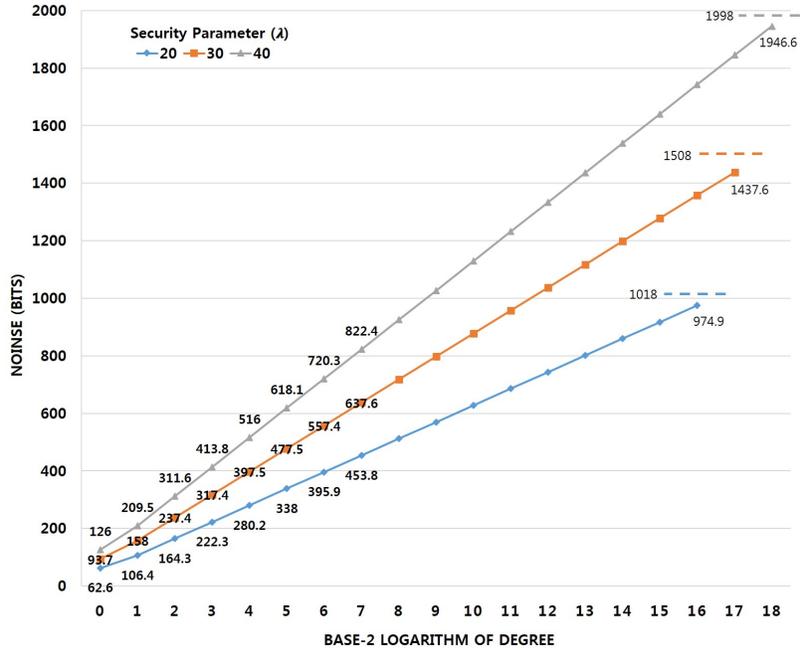


Fig. 3. Noise increase of Polynomial with respect to Degree