Heuristic Ideal Obfuscation Based on Evasive LWR-Based Multilinear Maps and Its Underlying Problem Reduction

Zhuang Shan[1] [0000−0003−4309−4505], Leyou Zhang[1,*] [0000−0003−4950−1140], and Qing Wu[2] [0000−0002−9400−236X]

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
arcsec30@163com, lyzhang@mail.xidian.edu.cn

[2] School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China xiyouwuq@126.com

Abstract. This paper introduces a heuristic ideal obfuscation scheme grounded in the learning problem, which differs from that proposed by Jain, Lin, and Luo [JLLW23]. The approach in this paper follows a methodology akin to that of Brakerski, Dottling, Garg, and Malavolta [BDGM22,BDGM20] for building iO. We construct a new ideal obfuscation by leveraging a variant of LWR to build LHE and employing *Evasive LWR* to construct multilinear maps. In contrast to the methodology of Jain et al., this paper provides a more detailed approach. Initially, we reprove the hardness of LWR using the prime number theorem and the fixed-point theorem, showing that the statistical distance between $\lfloor As \rfloor_p$ and $\lfloor u \rfloor_p$ does not exceed $\exp\left(-\frac{n \log_2 n \ln p}{\sqrt{5}}\right)$ when the security parameter $q > 2^n p$. Additionally, we provide definitions for *Evasive LWR* and *composite homomorphic pseudorandom function*, and based on these, we construct multilinear maps, thereby establishing the ideal obfuscation scheme proposed in this paper.

Keywords: Ideal obfuscation · Split FHE · Multilinear maps · Lattice problem reduction · Evasive Lattice.

1 Introduction

In 2000, Hada[Had00] first introduced the definition of virtual black box obfuscation, which is essential for embedding a circuit C into an opaque black box that cannot be opened. By inputting x into one end of the black box, the other end automatically outputs $C(x)$. Since the entire circuit is hidden inside the black box, no specific information about the construction of C can be obtained. The only action we can take is to provide input and observe the output on the other side.

VBB functions like a virtualized black box, where a circuit C obfuscated by VBB prevents us from obtaining any information related to its construction through the obfuscated output. The only action possible is to provide input x and compute $C(x)$[Yue20]. Unfortunately, Barak et al.[BGI$^+$01] have proven that virtual black box obfuscation does not exist.

In 2001, while Barak et al. proved the nonexistence of virtual black box obfuscation, they also presented a new definition for obfuscation: to obfuscate two circuits C_1 and C_2 such that the obfuscated circuits have the same functionality and an adversary cannot distinguish between the two circuits. This is known as indistinguishable obfuscation.

In 2013, Garg et al. introduced indistinguishable obfuscation based on multilinear maps [GGH$^+$13b] and applied it to functional encryption. It is noteworthy that multilinear maps were also proposed by Garg et al. [GGH13a]. Subsequently, significant work using program obfuscation(e.g., [BZ17,GGHR14,SW21]) has shown that most interesting cryptographic applications can be realized using iO (and one-way functions).

Due to its importance, many scholars have begun to focus on researching how to construct indistinguishable obfuscation. One construction method is based on new multilinear maps, which extends its applicability to a wider range [GGH13a,CLT13,GGH15]. However, in 2016, Hu and Jia [HJ16] broke the indistinguishable obfuscation based on multilinear maps proposed by Garg et al. [GGH13a]. In the same year, Miles, Sahai, and Zhandry [MSZ16] partially broke another indistinguishable obfuscation scheme by Garg et al. [GGH$^+$13b]. Since 2015, the field of obfuscation with multilinear pairings has entered a cycle where proposed schemes are quickly broken, leading to improvements based on the attacks, only to be broken again shortly thereafter.

Recently, Bitansky and Vaikuntanathan [BV18] and Ananth and Jain [AJ15] have independently proven through different methods that when Compact FE (Functional Encryption with compact ciphertexts) exists, then indistinguishable obfuscation can be achieved. Based on these results, the current construction methods for indistinguishable obfuscation mainly fall into two categories, namely:

1. The first approach is to restrict the depth of multilinear maps to achieve indistinguishable obfuscation. For example, in 2016, Lin restricted the depth to 5 layers [Lin17], and later with Tessaro restricted it to 3 layers [LT17]. In 2020, Jain, Lin, and Sahai [JLS21] successfully constructed indistinguishable obfuscation based on bilinear pairings, LWE, LPN[YZ21], and sPFG.
2. The second approach is to achieve indistinguishable obfuscation through splitting fully homomorphic encryption. For example, Brakerski, Dottling, Garg, and Malavolta [BDGM22,BDGM20] combined fully homomorphic encryption (FHE) with leveled homomorphic encryption (LHE) (Damgård-Jurik). By cleverly leveraging circular-security assumptions, they enable ciphertexts to circulate between the two encryption systems, ultimately constructing indistinguishable obfuscation.

In 2023, Jain, Lin, and Luo introduced a new concept called ideal obfuscation [JLLW23]. This concept is a refinement of Jain's work on indistinguishable obfuscation.

1.1 Our work

New LWR reduction The LWR problem is a variant of the LWE problem [Reg04], while being reduced to the SVP and CVP problem. In 2012, Banerjee, Peikert, and Rosen first proposed this problem, which is primarily used to construct pseudorandom functions and deterministic encryption [XXZ12]. In 2013, Alwen and others used information entropy theory to reduce the learning with rounding problem to the learning with errors problem, indirectly leading to a reduction to the SVP and CVP in lattices, and requires $q > \beta 2^n p$.

In 2024, Dr. Chen [Che24] published an article on quantum algorithms for LWE, although there are some issues with the proof process, but this also serves as a warning that not all lattice problems can be reduced to LWE. Therefore, this paper redefines the LWR problem without using any intermediate problems as reduction bridges. Instead, it calculates the maximum statistical distance between $\lfloor As \rfloor_p$ and $\lfloor u \rfloor_p$.

Define $\mathcal{S}_q$ as the probability set for any $i \in \mathbb{Z}_q$, undoubtedly, $\Pr(i) = \frac{1}{q}$, for all $i \in \mathbb{Z}_q$. $\mathcal{S}_q[i] = \Pr(i)$. Now, calculate the value when $\Pr(i) \in \overline{\mathcal{S}}_q := \mathcal{S}_q \times \mathcal{S}_q$.

Since $\Pr(i)$ is related to the prime factorization of i, the value of $\Pr(i)$ is given based on the prime number theorem and the theory of prime factorization. Define its maximum value as $\Pr(\mathfrak{A})$, with the corresponding element $\mathfrak{A} \in \mathbb{Z}_q$. At this point, we are calculating the probability of $a_1 s_1 = b_1 \to \mathbb{Z}_q$, then we calculate $\Pr(i) \in (\overline{\mathcal{S}}_q + \overline{\mathcal{S}}_q)$, which is the probability of $a_1 s_1 + a_2 s_2 = b_2 \to \mathbb{Z}_q$, and so on. We find that this relationship is consistent with the relationship $i_{n+1} = T_{i_n} i_n$, where

$$T_{i_n} = \begin{pmatrix} i_n^{(1)} & i_n^{(q)} & \cdots & i_n^{(2)} \\ i_n^{(2)} & i_n^{(1)} & \cdots & i_n^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ i_n^{(q)} & i_n^{(q-1)} & \cdots & i_n^{(1)} \end{pmatrix},$$

$i_n^{(j)} = \Pr(j)$. So the idea arose whether we could leverage the theory of fixed points to prove the convergence of this sequence, and perhaps even establish the hardness of LWR? Indeed, it is affirmative, because T_{i_n} satisfies the conditions of the fixed-point theorem, namely, T_{i_n} is a κ-contraction operator, and it converges to $(\underbrace{\frac{1}{q}, \ldots, \frac{1}{q}}_{q})$. However, it is worth noting that T_{i_n} has more than one fixed point. So why does it only converge to $(\underbrace{\frac{1}{q}, \ldots, \frac{1}{q}}_{q})$? This is because the other fixed points take the form $i^{(k)} = 1$, $i^{(l)} = 0$, where $l \neq k$, $k \in \mathbb{Z}_q$. However, the previous analysis on probabilities indicates that $i_n^{(j)} \in (0, 1)$. Therefore, it will only converge to $(\underbrace{\frac{1}{q}, \ldots, \frac{1}{q}}_{q})$. Furthermore, we can obtain

$$\|i_{n+1} - i_n\| \leq \kappa \|i_n - i_{n-1}\| \leq \cdots \leq \kappa^{n-1} \|i_2 - i_1\|.$$

Therefore, as long as κ^{n-1} meets the requirement, we can complete the reduction to LWR. Based on the learning with rounding problem, provide two variants of the indistinguishability theorem, as follows:

Theorem (Informal). *Let $q > 2^n p$ be prime numbers, $A \in \mathbb{Z}_q^{m \times n}$, $s \in \mathbb{Z}_3^n$, $u \in \mathbb{Z}_q^m$. If it is difficult to distinguish between $(A, \lfloor As \rceil_p)$ and $(A, \lfloor u \rceil_p)$, then for $a, b \in_R \mathbb{Z}_q$ (or $b \in_R \mathbb{Z}_q$, $a \in_R \mathbb{Z}_q^{m \times n}$), we have*

$$(\odot_q(A, a), \lfloor u \rceil_p) \approx_c \left(\odot_q(A, a), \lfloor \odot_q(A, ab) \cdot s \rceil_p\right).$$

Building on the work of Brakerski et.al.[BDGM22,BDGM20], provide a heuristic notion of indistinguishability obfuscation. Leveraging the second variant problem (referred to for ease of exposition as the delta variant of learning with rounding, LWR.DV), we construct a linear homomorphic scheme. This LWR.DV-based linear homomorphic scheme theoretically possesses properties resistant to quantum attacks.

Evasive LWR and multilinear maps In 2015, Gentry, Gorbunov, and Halevi presented lattice-based Multilinear Maps [GGH15], defined as

$$A, S_1A_1 + E_1, \ldots, S_kA_k + E_k \to \prod_{i=1}^{n} SA + E \bmod q.$$

Most notably, none of the current indistinguishability obfuscation candidates from GGH15 have any formal security guarantees against zeroizing attacks [BGMZ18].

To resist zeroizing attacks, in 2022, Wee introduced the definition of Evasive LWE [Wee22] and proposed new multilinear maps [VWW22], defined as

$$\begin{aligned} &A, (uM_1 \otimes S_1)A_1 + E_1, \ldots, A_{k-1}^{-1}((M_k \otimes S_k)A_k + E_k)) \\ &\qquad \to \left(\left(u\prod_{i=1}^{n} M_i\right) \otimes \left(\prod_{i=1}^{n} S_i\right)\right) A + E \bmod q. \end{aligned}$$

Inspired by this, we attempt to construct a new GGH15 multilinear maps based on Evasive LWR. Here's our first attempt.

First attempt

KeyGen(n, m, q)**.** Generate necessary parameters.
Eval$(M = (M_1, \ldots, M_\ell), (u, \{R_i\}_{i=1}^{\ell}))$**.** • Set S_i as

$$\widehat{S}_i = \begin{cases} u(M_1 \otimes R_1), & \text{when } i = 1, \\ M_i \otimes R_i, & \text{when } i > 1. \end{cases}$$

- Output encrypted result

$$\{\lfloor u(M_1 \otimes R_1)A_1 \rfloor_p\}, \left\{\lfloor A_{i-1}^{-1}(M_i \otimes R_i)A_i \rfloor_p\right\}_{i=1}^{\ell}.$$

Remark 1. $M_i \otimes R_i$ is not a random matrix, hence it does not satisfy the Evasive LWR assumption.

Second attempt

KeyGen(n, m, q**)**. Generate necessary parameters.

Eval($M = (M_1, \ldots, M_\ell), (u, \{R_i\}_{i=1}^{\ell})$**)**. • Set S_i as

$$\widehat{S}_i = \begin{cases} u(M_1 \odot R_1), & \text{when } i = 1, \\ M_i \odot R_i, & \text{when } i > 1. \end{cases}$$

- Output encrypted result

$$\{\lfloor u(M_1 \odot R_1)A_1 \rfloor_p\}, \left\{\lfloor A_{i-1}^{-1}(M_i \odot R_i)A_i \rfloor_p\right\}_{i=1}^{\ell}.$$

Remark 2. Although $M_i \odot R_i$ meets the randomness requirement, its computational cost is slightly high. Therefore, considering increasing the randomness of M_i to reduce computational expenses.

Third Attempt

We refer to the following scheme as composite pseudorandom function.

KeyGen(n, m, q**)**. Generate necessary parameters.

PRF.Enc($\{M_i\}_{i=1}^{\ell}, u, key, n, m, q$**)**.

$$C_i = PRF(\{M_i\}_{i=1}^{\ell}, key)$$

Eval($C = (C_1, \ldots, C_\ell)$**)**. • Set S_i as

$$\widehat{S}_i = \begin{cases} u(C_1), & \text{when } i = 1, \\ C_i, & \text{when } i > 1. \end{cases}$$

- Output encrypted result

$$\{\lfloor u(C_1)A_1 \rfloor_p\}, \left\{\lfloor A_{i-1}^{-1}(C_i)A_i \rfloor_p\right\}_{i=2}^{\ell}.$$

Remark 3. Using a pseudorandom function improves the randomness of M_i while also reducing computational overhead.

Conceptual Ideal Obfuscation Scheme Next, present a conceptual split FHE scheme (ideal obfuscation scheme), which is based on three main techniques: (i) Using multilinear maps to construct FHE scheme, (ii) short decryption gadgets for linear homomorphic encryption schemes (such as the scheme in this paper, based on the LWR.DV problem), and (iii) encrypted hash functions (used for a part of the linear homomorphic encryption scheme). The security of this scheme can be based on a new conjecture regarding the interaction of these primitives, which we believe is a natural strengthening of circular security.

We aim to instantiate the underlying primitives randomly rather than non-randomly, as non-random instantiations of primitives are insecure, and thus would lead to an insecure split FHE scheme. For randomly instantiated primitives, we can speculate about their security.

Security Proof. In order to prove the security of our scheme, demonstrate the existence of an oracle that interacts securely between the underlying primitives and a randomly instantiated scheme. This oracle is defined as $\mathcal{O}_{(\widehat{pk},\overline{pk},q,\tilde{q})}(x)$: given a string $x \in \{0,1\}^*$ and a ciphertext taken from the ciphertext space of the linear homomorphic scheme,

$$c \leftarrow \mathcal{C},$$

it then calculates

$$\tilde{c} \leftarrow \mathrm{Eval}(\widehat{pk}, -\lfloor \overline{\mathrm{DEC}}(\cdot, c)/\tilde{q} \rfloor \cdot \tilde{q}, \widehat{c}),$$

and returns $(c, \tilde{c})$. In this paper, we use this oracle for the security proof of the scheme.

Theorem (Informal). *Assuming the sub-exponential hardness of the LWR problem and the Evasive LWR problem, there exists a sub-exponentially secure split fully homomorphic encryption scheme. Consequently, there exists an ideal obfuscation that can be applied to any circuit.*

1.2 Technical Overview

Next, provide a generalized description of the method for constructing split FHE, and readers can refer to relevant literature for a more detailed description.

Split FHE In 2019, Brakerski et al. [BDGM19] introduced the concept of a split FHE scheme. Asymptotically, they aimed to design an efficient FHE scheme by eliminating linear noise in previous Evasive LWR-based FHE schemes. More specifically, given an FHE ciphertext c and an Evasive LWR key $(s_1, \ldots, s_n)$, we can denote the decryption operator as a linear function $\mathcal{L}_c(\cdot)$, that is

$$\mathcal{L}_c(s_1, \ldots, s_n) = \mathrm{ECC}(m) + e.$$

Here, e is a noise term bounded by B, and ECC is the encoding operator for the text. Then, this paper introduces the construction of a linear homomorphic scheme using LWR.DV, and encrypts the key $(s_1, \ldots, s_n)$ with this homomorphic encryption scheme, allowing the compression of FHE ciphertexts through the computation of $\mathcal{L}_c(\cdot)$. The public key of this scheme is $(r \in_R \{0,1\}^n, \odot_q(A,l))$, and it computes the encryption of a message m as

$$c = \lfloor \odot_q(A, lu)(m+k) \rfloor_p.$$

Here, $u = H(r)$, where $H : \{0,1\}^n \to \mathbb{Z}_q$ and $k \in_R \{0, \ell+1\}$. Furthermore, this scheme possesses an additional property, which refer to as split decryption. If the decryption algorithm can be divided into a private subroutine and a public subroutine, then the scheme has split decryption:

– The private process takes a ciphertext c and key $(\odot_q(A, lu), T_{sk})$ as input, outputs $\tilde{m} = \mathrm{LWRInvert}(T_{sk}, \odot_q(A, lu), c)$. For each component $\tilde{m}_i$ of $\tilde{m}$,

$$\begin{cases} \tilde{k}_i = 0, & \text{if } \tilde{m}_i \in \{0,1\}, \\ \tilde{k}_i = \tilde{m}_i, & \text{if } \tilde{m}_i \in \{(\ell+1), \ldots, n(\ell+1)\}, \\ \tilde{k}_i = \tilde{m}_i - 1, & \text{if } \tilde{m}_i \notin \{0, 1, (\ell+1), \ldots, n(\ell+1)\}. \end{cases}$$

It returns the decryption primer $\rho = \left(sk, \tilde{k} = (\tilde{k}_i)_{i \in \{1,\ldots,\ell\}}\right)$.

– The public process takes the ciphertext c and decryption primer ρ as inputs, outputs $\tilde{m} = \mathrm{LWRInvert}(T_{sk}, \odot_q(A, lu), c)$, decrypts $m' = \tilde{m} - \tilde{k}$.

In summary, m can be fully recovered by passing a fixed-size decryption primer, especially independent of the norm of m. As we will discuss later, this property will be the main feature in constructing universal obfuscation.

FHE scheme and sFHE scheme based on multilinear maps Because Evasive LWR itself possesses certain homomorphic properties, namely

$$\lfloor A_{i-1}^{-1}(C_i^{(1)})A_i \rfloor_p + \lfloor A_{i-1}^{-1}(C_i^{(2)})A_i \rfloor_p \approx \lfloor A_{i-1}^{-1}(C_i^{(1)} + C_i^{(2)})A_i \rfloor_p,$$

and

$$\lfloor A_{i-1}^{-1}(C_i)A_i \rfloor_p \cdot \frac{q}{p} \lfloor A_i^{-1}(C_{i+1})A_{i+1} \rfloor_p \approx \lfloor A_{i-1}^{-1}(C_i C_{i+1})A_{i+1} \rfloor_p.$$

Thus, it is possible to obtain $C_i^{(1)} + C_i^{(2)}$ and $C_i C_{i+1}$ using LWR gates. However, to obtain the corresponding $M_i^{(1)} + M_i^{(2)}$ and $M_i M_{i+1}$, the PRF.Enc function must also possess homomorphic and decryptable properties, which is not a feature of ordinary pseudorandom functions.

The Security of Split FHE We now discuss the security of the split FHE scheme. Our primary concern is ensuring that the decryption primer does not carry any information about the plaintext; otherwise, the simplicity of the split encryption process and straightforward output of keys in every scheme would be moot. We propose a more profound indistinguishability definition, meaning that for all plaintext pairs (m_0, m_1) and any set of circuits $(C_1, \ldots, C_\beta)$, we have $C_i(m_0) = C_i(m_1)$. Even if an adversary knows the decryption primer ρ_i, they cannot distinguish between the encryptions of (m_0, m_1) as (c_0, c_1). The condition $C_i(m_0) = C_i(m_1)$ eliminates some other attacks, where the adversary only needs to check the obfuscator's output. Here, $\beta = \beta(\lambda)$ is a priori bounded polynomial of a security parameter.

Theorem (Informal). *Assuming the sub-exponential hardness of the LWR problem and the Evasive LWR problem, there exists a split FHE scheme secure under the $\mathcal{O}$-hybrid security model.*

From Split FHE Scheme to Ideal Obfuscation Utilize the split FHE scheme presented in this paper to construct ideal obfuscation. Building on the work of Lin et al. [Lin17], we achieve an obfuscated circuit C with input domain $\{0,1\}^\eta$ whose length does not exceed $\mathrm{poly}(\lambda, |C|) \cdot 2\eta \cdot (1-\varepsilon)$, where $\varepsilon > 0$. This implies that split FHE signifies the existence of an obfuscator with non-trivial efficiency (for circuits with polynomial-size input domains).

2 Preliminary

We define a function negl$(\cdot)$, which is an infinitesimal of any polynomial function poly, and we refer to it as "negligible". Given a set $\mathcal{S}$, $s \in_R \mathcal{S}$ means randomly selecting an element s from the set $\mathcal{S}$. When an algorithm can be computed within a polynomial function poly, we say that this algorithm is "computable in polynomial time".

Lemma 1 ([AJLA+12], Smudging). *Let $B_1 = B_1(n)$, $B_2 = B_2(n)$ be positive integers, and $e_1 \in [B_1]$. Let $e_2 \in_R [B_2]$. If $B_1/B_2 = \mathrm{negl}(n)$, then the distribution of e_2 is computationally indistinguishable from the distribution of $e_2 + e_1$.*

Definition 1 ([GSM18], page 32, Section 4.1.6). *We say that $\varepsilon(n)$ is negligible associated with n if $\varepsilon(n)$ can be expressed as*

$$\varepsilon(n) = \frac{1}{O(e^n)},$$

and the notation $O(n)$ represents a quantity that grows at most as fast as n approaches infinity.

2.1 The new reduction for LWR.

Dr. Chen Yilei's attack does have some flaws. However, it also serves as a warning that we cannot reduce all difficult problems to LWE. We utilize the prime number theorem and fixed-point theory to reevaluate its reduction.

Lemma 2 ([Nor66]). *For $q \in \mathbb{Z}$, the prime distribution over the set $\mathcal{S}_q = \{1, \ldots, q\}$ satisfies the following relationship:*

$$\lim_{q \to \infty} \frac{\pi(q)}{\int_2^q \frac{1}{\ln(t)} dt} = 1,$$

where $\pi(q)$ denotes the number of primes.

Claim 1. Let $\pi(q)$ denote the number of prime numbers in the set $\mathcal{S}_q$, and let $P_q := \{p_1, \ldots, p_{\pi(q)}\}$ be the set of all prime numbers. Then, the number of prime numbers in the set $\mathcal{S}_q \times \mathcal{S}_q$ is still $\pi(q)$, and we have

$$\overline{\mathcal{S}}_q := \mathcal{S}_q \times \mathcal{S}_q = \mathcal{S}_{q^2} \setminus \left\{ \mathcal{S}_q \times (P_{q^2} \setminus P_q) \right\}.$$

Claim 2. For the set $\mathcal{S}_q$, for an element $a \in \overline{\mathcal{S}}_q$ with prime factorization $\mathfrak{p}_1^{\alpha_1} \cdots \mathfrak{p}_\ell^{\alpha_\ell}$, the probability of a occurring in the set $\overline{\mathcal{S}}_q$ is

$$\Pr(a) := \frac{CN_q(a)}{q^2} = \sum_{\mathfrak{P}_1, \mathfrak{P}_2 \in \overline{\mathcal{S}}_q} \frac{2}{q^2},$$

where $\mathfrak{P}_1 = \mathfrak{p}_1^{\alpha_1'} \cdots \mathfrak{p}_\ell^{\alpha_\ell'}$, $\mathfrak{P}_2 = \mathfrak{p}_1^{\alpha_1''} \cdots \mathfrak{p}_\ell^{\alpha_\ell''}$, $\alpha_i' + \alpha_i'' = \alpha_i$, $i \in \mathcal{S}$.

Claim 3. For the set $\mathcal{S}_q$, where each event occurs with probability q^{-1}, then for the set $\mathcal{S}_q \times \mathcal{S}_q$, the probability of each event a occurring is

$$\Pr(a) = \begin{cases} \dfrac{1}{q^2}, & a = 1, \\ \dfrac{2}{q^2}, & a \text{ is prime}, \\ \dfrac{CN_q(a)}{q^2}, & a \text{ is composite}. \end{cases}$$

Claim 4. For $\mathfrak{A} \in \overline{\mathcal{S}}_q$, and assuming

$$\Pr(\mathfrak{A}) = \max_{a \in \overline{\mathcal{S}}_q} \Pr(a) = \frac{CN_q(\mathfrak{A})}{q^2},$$

then for any $k \in (\overline{\mathcal{S}}_q + \overline{\mathcal{S}}_q) \bmod q \to \mathcal{S}_{q^2}$, we have

$$\frac{1}{q^2} \leq \Pr(k) \leq \Pr(\mathfrak{A}) = \frac{CN_q(\mathfrak{A})}{q^2}.$$

Proof.

$$\begin{aligned} \Pr(k) &= \sum_{i=1}^{k} \Pr(i) \Pr(k+i-1) + \sum_{i=k+1}^{q+k-1} \Pr(i) \Pr(q+k+1-i) \\ &= \sum_{i=1}^{k} \frac{a_i}{q^2} \frac{a_{k+i-1}}{q^2} + \sum_{i=k+1}^{q+k-1} \frac{a_i}{q^2} \frac{a_{q+k+1-i}}{q^2} \\ &\leq \frac{CN_q(\mathfrak{A})}{q^2} \sum_{i=1}^{q^2} \frac{a_i}{q^2} = \frac{CN_q(\mathfrak{A})}{q^2}, \end{aligned}$$

and

$$\begin{aligned} \Pr(k) &= \sum_{i=1}^{k} \Pr(i) \Pr(k+i-1) + \sum_{i=k+1}^{q+k-1} \Pr(i) \Pr(q+k+1-i) \\ &\geq \frac{1}{q^2} \sum_{i=1}^{q^2} \frac{a_i}{q^2} = \frac{1}{q^2}. \end{aligned}$$

Definition 2 ([Ceg12], Definition 2.1.6). *Let $\mathcal{H}$ be a Hilbert space, and let $T : \mathcal{H} \to \mathcal{H}$ be an operator. If $T(\cdot)$ satisfies*

$$\|Tx - Ty\| < \|x - y\|, \ \forall x,\ y \in \mathcal{H},$$

then $T(\cdot)$ is called a contraction operator.

Lemma 3 ([Ceg12], Proposition 2.1.11). *If $\mathcal{H}$ is a closed set (every Cauchy sequence in $\mathcal{H}$ converges to a point within $\mathcal{H}$), and $T(\cdot)$ is a contraction operator, and $Fix(T)$ is a closed convex set, then the algorithm $x_{n+1} = Tx_n$ converges to some $x \in Fix(T)$, where $Fix(T)$ denotes the set of fixed points of the operator $T(\cdot)$.*

Remark 4. The convergence mentioned in Lemma 3 should be considered as strong convergence. However, this paper does not discuss the difference between strong and weak convergence, because in finite dimensions strong and weak convergence are equivalent.

Claim 5. For any vector $a = (a^{(1)}, a^{(2)}, \ldots, a^{(q)})$, where $a_i \in [0,1]$ and $\sum_{i=1}^{q} a^{(i)} = 1$, let $A_k = \max_{i\in\mathcal{S}_q}(a^{(i)})$. Then, the matrix M_a defined as follows is a contraction operator

$$M_a = \begin{pmatrix} a^{(1)} & a^{(q)} & \cdots & a^{(2)} \\ a^{(2)} & a^{(1)} & \cdots & a^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ a^{(q)} & a^{(q-1)} & \cdots & a^{(1)} \end{pmatrix}.$$

Proof. For any vectors $b = (b^{(1)}, b^{(2)}, \ldots, b^{(q)})$ and $c = (c^{(1)}, c^{(2)}, \ldots, c^{(q)})$ satisfying the conditions of vector a, and

$$\begin{aligned} \|M_a b - M_a c\| &= \|M_a(b-c)\| \le \|M_a\|\|b-c\| \\ &= \sqrt{\frac{CN_q^2(\mathfrak{A})}{q^2} + \frac{q-2}{q^2} + \frac{(2\sqrt{q} - CN_q(\mathfrak{A}))^2}{q^2}}\|b-c\| \\ &= \frac{\sqrt{5q - 4\sqrt{q}CN_q(\mathfrak{A}) + 2CN_q^2(\mathfrak{A}) + 2}}{q}\|b-c\| \\ &< \|b-c\|. \end{aligned}$$

Lemma 4. *For any initial vector $a_0 = (a_0^{(1)}, a_0^{(2)}, \ldots, a_0^{(q)})$, where $a_0^{(i)} \in [0,1]$ and $\sum_{i=1}^{q} a_0^{(i)} = 1$, and $CN_q(\mathfrak{A}) = \max_{i\in\mathcal{S}_q}(a_0^{(i)})$, the matrix M_{a_0} is generated as follows:*

$$M_{a_0} = \begin{pmatrix} a_0^{(1)} & a_0^{(q)} & \cdots & a_0^{(2)} \\ a_0^{(2)} & a_0^{(1)} & \cdots & a_0^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{(q)} & a_0^{(q-1)} & \cdots & a_0^{(1)} \end{pmatrix}.$$

Then, let $a_{n+1} := M_{a_n} a_n := T a_n$*, then* $\{a_n\}_{n=1}^{\infty}$ *is a Cauchy sequence and converges to* $(\underbrace{\frac{1}{q}, \ldots, \frac{1}{q}}_{q})$.

Proof. According to Claim 5, we know that M_{a_n} is a contraction operator, and

$$\|M_{a_n}\| \leq \frac{\sqrt{5q - 4\sqrt{q}CN_q(\mathfrak{A}) + 2CN_q^2(\mathfrak{A}) + 2}}{q}.$$

Moreover, since $a_{n+1} := M_{a_n} a_n$ is itself an algorithm for finding fixed points, the sequence $\{a_n\}_{n=1}^{\infty}$ converges, and it converges to the fixed point of $T(\cdot)$.

Lemma 5. *For the set* $\mathcal{S}_q^i := \{1, 2, \ldots, q\}$*, where each event* $j \in \mathcal{S}_q$ *has a probability of occurrence* a_j*, as* n *approaches infinity, the probability of each event after taking modulo* q *over* $\mathcal{S}_q := \prod_{i=1}^{n} \mathcal{S}_q^i \bmod q = \{1, 2, \ldots, q\}$ *tends toward* $\frac{1}{q}$.

Theorem 1. *Given* $\{a_j\}_{j=1}^{n}$ *and* $\{s_j\}_{j=1}^{n}$ *such that* $a_j,\ s_j \in_R \mathbb{Z}_q$*. Then for any* $i \in_R \mathbb{Z}_q$*, we have*

$$\max_{i \in \mathcal{S}_q} \left| \Pr\left(\sum_{j=1}^{n} (a_j s_j) = i \right) - \Pr(u = i) \right| \leq \exp\left(2n \ln\left(\frac{\sqrt{5q - 4\sqrt{q}CN_q(\mathfrak{A}) + 2CN_q^2(\mathfrak{A}) + 2}}{q} \right) \right).$$

Corollary 1. *For any* $A \in \mathbb{Z}_q^{m \times n}$*,* $s \in \mathbb{Z}_q^n$*, and* $u \in \mathbb{Z}_q^m$*, where* $q > 2^n p$*, the indistinguishability probability between* As *and* u *is bounded by*

$$\exp\left(-2 \log_2 n \ln\left(\frac{q}{\sqrt{5q - 4\sqrt{q}CN_q(\mathfrak{A}) + 2CN_q^2(\mathfrak{A}) + 2}} \right) \right) \leq \exp\left(-\frac{n \log_2 n \ln p}{\sqrt{5}} \right).$$

Theorem 2. *For any* $A \in \mathbb{Z}_q^{m \times n}$*,* $s \in \mathbb{Z}_q^n$*, and* $u \in \mathbb{Z}_q^m$*, where* $q > 2^n p$*, the indistinguishability probability between* $\lfloor As \rfloor_p$ *and* $\lfloor u \rfloor_p$ *is bounded by*

$$\exp\left(-\frac{n \log_2 n \ln p}{\sqrt{5}} \right).$$

That is, the adversary's advantage in distinguishing between $\lfloor As \rfloor_p$ *and* $\lfloor u \rfloor_p$ *can be neglected.*

2.2 LWR Trapdoor Algorithm

Algorithm 1 LWR Trapdoor Algorithm [AKPW13]

GenTrap($\boldsymbol{n, m, q}$): A method that outputs $A \in \mathbb{Z}_q^{m\times n}$ and a trapdoor T in polynomial time, where the input to this algorithm is an integer n, q, and a sufficiently large integer m. The matrix A is uniformly distributed in $\mathbb{Z}_q^{m\times n}$.

Invert($\boldsymbol{T, A, c}$): A method that outputs $s \in \mathbb{Z}_q^n$ from $c = As + e \in \mathbb{Z}_q^m$ in polynomial time, with $\|e\|_2 \leq \gamma$. The input to this algorithm is the output A and trapdoor T from the **GenTrap($\boldsymbol{n, m, q}$)** algorithm.

LWRInvert($\boldsymbol{T, A, c}$): A method that outputs $s \in \mathbb{Z}_q^n$ from $c = \lfloor As \rfloor_p \in \mathbb{Z}_p^m$ in polynomial time. The input to this algorithm is the output A and trapdoor T from the **GenTrap($\boldsymbol{n, m, q}$)** algorithm.

Lemma 6 (Existence Lemma of LWR Trapdoor Algorithm, [AKPW13]). *The LWR trapdoor algorithm definitely exists, that is, for integers n, q, sufficiently large integer $m \geq O(n\log q)$, and sufficiently large integer $p \geq O(\sqrt{n\log q})$, there exist algorithms* GenTrap$(\boldsymbol{n, m, q})$ *and* LWRInvert$(\boldsymbol{T, A, c})$ *that output results in polynomial time.*

2.3 Variants of LWR and Their Applications

Lemma 7. *If $a \in_R \mathbb{Z}_q$, then for $r \in_R \mathbb{Z}_p$, $a^r \bmod q$ is indistinguishable from $u \in_R \mathbb{Z}_q$.*

Proof. According to Lemma 1, it is easy to prove.

Lemma 8. *If $A \in_R \mathbb{Z}_q^{m\times n}$, then $\odot_q(A, r) \in_R \mathbb{Z}_q^{m\times n}$. Where the operation $\odot_q(A, r)$ is defined as follows:*

$$\odot_q(A,r) = \begin{cases} \tilde{A}\left(a_{ij} \in A, a_{ij}^r \in \tilde{A}, i \in \{1,\dots,m\}, j \in \{1,\dots,n\}\right) \bmod q, & \textit{for } r \in \mathbb{Z}_q, \\ \tilde{A}\left(a_{ij} \in A, a_{ij}^{r_{ij}} \in \tilde{A}, i \in \{1,\dots,m\}, j \in \{1,\dots,n\}\right) \bmod q, & \textit{for } r \in \mathbb{Z}_q^{m\times n}. \end{cases}$$

Proof. From Lemma 7, if $a_{ij} \in_R \mathbb{Z}_q$, then $a_{ij}^r \bmod q \in_R \mathbb{Z}_q$ (or $a_{ij}^{r_{ij}} \bmod q \in_R \mathbb{Z}_q$). Therefore, $\odot_q(A, r) \in_R \mathbb{Z}_q^{m\times n}$.

Lemma 9. *If $(A, \lfloor As \rfloor_p)$ and $(A, \lfloor u \rfloor_p)$ are indistinguishable, then for $r \in_R \mathbb{Z}_p$ (or $r \in \mathbb{Z}_q^{m\times n}$), $(\odot_q(A, r), \lfloor \odot_q(A, r) \cdot s \rfloor_p)$ and $(\odot_q(A, r), \lfloor u \rfloor_p)$ are also indistinguishable.*

Proof. According to the form of LWR, when $A \in \mathbb{Z}_q^{m\times n}$, we have $\tilde{A} = \odot_q(A, r) \in \mathbb{Z}_q^{m\times n}$. Therefore, $(\tilde{A}, \lfloor \tilde{A}s \rfloor_p)$ and $(\tilde{A}, \lfloor u \rfloor_p)$ still maintain indistinguishability.

Corollary 2. *If there exists an algorithm $\mathcal{O}$ to solve the LWR problem, then there also exists an algorithm $\mathcal{O}'$ to solve the Variant LWR problem, and vice versa.*

Proof. According to Lemma 8, the sufficiency of the proposition is established. Now, to prove the necessity, since f is a bijection, there exists f^{-1} such that $f^{-1} \cdot f = f \cdot f^{-1} = Id$. It can be easily shown that f^{-1} is also a bijection. Hence, when $\odot_q(A, r) \in_R \mathbb{Z}_q^{m\times n}$, it implies $A \in_R \mathbb{Z}_q^{m\times n}$, thus the necessity is proved.

Lemma 10. *If $(A, \lfloor As \rfloor_p)$ is indistinguishable from $(A, \lfloor u \rfloor_p)$, then for randomly chosen $b \in_R \mathbb{Z}_p$ and $a \in_R \mathbb{Z}_p^{m\times n}$, we have*

$$(\odot_q(A, a), \lfloor u \rfloor_p) \approx_c \left(\odot_q(A, a), \lfloor \odot_q(A, ab) \cdot s \rfloor_p\right).$$

Proof.

$$\begin{aligned}(\odot_q(A, a), \lfloor u \rfloor_p) &\approx_c \left(\odot_q(A, ab), \lfloor u \rfloor_p\right) \\ &\approx_c \left(\odot_q(A, ab), \lfloor \odot_q(A, ab) \cdot s \rfloor_p\right) \\ &\approx_c \left(\odot_q(A, a), \lfloor \odot_q(A, ab) \cdot s \rfloor_p\right).\end{aligned}$$

Corollary 3. *If $(A, \lfloor As \rfloor_p)$ is indistinguishable from $(A, \lfloor u \rfloor_p)$, $s \in \mathbb{Z}_3^n$, then for randomly chosen $b \in_R \mathbb{Z}_p$ and $a \in_R \mathbb{Z}_p^{m\times n}$, we have*

$$(\odot_q(A, a), \lfloor u \rfloor_p) \approx_c \left(\odot_q(A, a), \lfloor \odot_q(A, ab) \cdot s \rfloor_p\right).$$

Proof. Let $\overline{\mathcal{S}}_{q,3} := \mathcal{S}_q \times \mathcal{S}_3$, and for all $i \in \mathbb{Z}_q$, we have

$$\Pr(i) := \frac{CN_q(i)}{q^2} = \sum_{\mathfrak{P}_1, \mathfrak{P}_2 \in \overline{\mathcal{S}}_{q,3}} \frac{2}{q^2}.$$

Similarly, we can prove that

$$\frac{1}{q^2} \leq \Pr(i) \leq \frac{CN_q(\mathfrak{A})}{q^2},$$

thus, by utilizing fixed-point theory and Lemma 10, we obtain the conclusion.

2.4 $\aleph$-Graded Encoding System and composite homomorphic pseudorandom function

Definition 3 ($\aleph$-GES, [GGH13a]). *Let $\aleph \in \mathbb{N}^\tau$ be a $\tau > 0$ dimensional natural number vector. An $\aleph$-GES consists of a ring $\mathcal{R}$ and a set $\mathcal{S} = \{S_v^{(\alpha)} \subset \{0,1\}^* : \alpha \in \mathcal{R}, v \leq \aleph\}$ (each element of this set $S_v^{(\alpha)}$ is also a set, representing the set of all order-v encodings for the ring element α) with the following properties:*

1. *For all v, the sets $\{S_v^{(\alpha)} : \alpha \in \mathcal{R}\}$ are disjoint, thus $S_v = \bigcup_\alpha S_v^{(\alpha)}$.*
2. *There exist a binary operator '$+$' and a unary operator '-', such that for all α_1, $\alpha_2 \in \mathcal{R}$, $v \leq \aleph$, $u_1 \in S_v^{(\alpha_1)}$, and $u_2 \in S_v^{(\alpha_2)}$, the following operations hold:*
$$u_1 + u_2 \in S_v^{(\alpha_1+\alpha_2)}, \quad \textit{and} \quad -u_1 \in S_v^{(-\alpha_1)}.$$
Here, $\alpha_1 + \alpha_2$ and $-\alpha_1$ denote addition and negation in the ring $\mathcal{R}$.
3. *There exists a binary operator '$\times$', such that for any $\alpha_1, \alpha_2 \in \mathcal{R}$, $v_1+v_2 \leq \aleph$, $u_1 \in S_{v_1}^{(\alpha_1)}$, and $u_2 \in S_{v_2}^{(\alpha_2)}$, the following operation holds:*
$$u_1 \times u_2 \in S_{v_1+v_2}^{(\alpha_1 \cdot \alpha_2)}.$$
Here, $\alpha_1 \cdot \alpha_2$ represents multiplication in the ring $\mathcal{R}$, and $v_1 + v_2$ represents addition in $\mathbb{N}^\tau$.

Definition 4. *Define homomorphic pseudorandom functions F_1 and F_2 such that:*

1. *$F_1(x, key) \to C_i$, $F_1(0, key) \to 0$, $F_1(x_1, key) + F_1(x_2, key) = F_1(x_1 + x_2, key)$, and $F_1(x_1, key)F_1(x_2, key) = F_1(x_1x_2, key)$.*
2. *$F_2(C = (C_1, \ldots, C_\ell), key) = F_2(F_1(C_1, key), \ldots, F_1(C_\ell, key)) \to y$, where ℓ can be any value less than $\aleph$, and F_2 satisfies:*
   - *$F_2(C_1, key) + F_2(C_2, key) = F_2(C_1 + C_2, key)$,*
   - *$F_2(0, key) = 0$.*

2.5 Homomorphic Encryption and Ideal Obfuscation

Homomorphic encryption is defined as follows:

Definition 5 ([Gen09a]). *A homomorphic encryption scheme consists of the following components:*

– **KeyGen(n):** *Given a security parameter n, the key generation part returns a key pair (sk, pk).*
– **Enc(pk, m):** *Given the public key pk and the plaintext message m, the encryption part returns the encrypted ciphertext c.*
– **Eval(pk, C, $(c_1, \ldots, c_\ell)$):** *Given the public key pk, a circuit C of depth ℓ, and a vector of ciphertexts $(c_1, \ldots, c_\ell)$, the homomorphic operation part returns the ciphertext after homomorphic computation.*
– **Dec(sk, c):** *Given the private key sk and the ciphertext c, the decryption part returns the decrypted plaintext message m.*

Definition 6 (Correctness). *Let $n \in \mathbb{N}$, and C be a circuit of depth ℓ. For an encryption scheme* $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$ *with inputs $(m_1, \ldots, m_\ell)$, key pair* $(\mathrm{pk}, \mathrm{sk})$ *generated by* $\mathrm{KeyGen}(n)$*, and ciphertexts c_i generated by* $\mathrm{Enc}(\mathrm{pk}, m_i)$ *according to the scheme, we have*

$$\Pr[\mathrm{Dec}(\mathrm{sk}, \mathrm{Eval}(\mathrm{pk}, C, (c_1, \ldots, c_\ell))) = C(c_1, \ldots, c_\ell)] = 1.$$

Refer to such an encryption scheme as a homomorphic encryption scheme. We desire that the length of ciphertexts in the scheme does not increase due to the depth ℓ of circuit C, a property referred to as "compactness" (distinct from the concept of "compactness" in functional analysis).

Definition 7 (Compactness). *Let $n \in \mathbb{N}$, C be a circuit of depth ℓ, and* $\mathrm{poly}(\cdot)$ *be a polynomial function. For a homomorphic encryption scheme* $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$ *with inputs $(m_1, \ldots, m_\ell)$, key pair (pk, sk) generated by* $\mathrm{KeyGen}(n)$*, and ciphertexts c_i generated by* $\mathrm{Enc}(pk, m_i)$*, if*

$$|\mathrm{Eval}(\mathrm{pk}, C, (c_1, \ldots, c_\ell))| = \mathrm{poly}(n) \cdot |C(m_1, \ldots, m_\ell)|,$$

then one called the homomorphic encryption scheme compact. Define a weak security notion (implied by standard semantic security [38]) for convenience.

Definition 8 (Semantic Security). *Let $n \in \mathbb{N}$, C be a circuit of depth ℓ, and* $\mathrm{negl}(\cdot)$ *be a negligible function. For a homomorphic encryption scheme* $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$ *with inputs (m_0, m_1), key pair (pk, sk) generated by* $\mathrm{KeyGen}(n)$*, ciphertexts c_i generated by* $\mathrm{Enc}(pk, m_i)$*, and all polynomial-time distinguishers $\mathcal{D}$, if*

$$|\Pr[1 = \mathcal{D}(pk, \mathrm{Enc}(pk, m_0))] - \Pr[1 = \mathcal{D}(pk, \mathrm{Enc}(pk, m_1))]| = \mathrm{negl}(n),$$

then one called the homomorphic encryption scheme semantically secure. Here, the key pair (pk, sk) *is generated by* KeyGen(n) *of the scheme.*

Definition 9 (ϵ-Indistinguishability). *Consider two distributions* $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ *and* $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$*, and* $\epsilon : \mathbb{N} \to [0,1]$*. If for every sufficiently large* $\lambda \in \mathbb{N}$*, it holds that*

$$\left| \Pr_{x\leftarrow\mathcal{X}_\lambda}[\mathcal{A}(1^\lambda, x) = 1] - \Pr_{y\leftarrow\mathcal{Y}_\lambda}[\mathcal{A}(1^\lambda, y) = 1] \right| \le \epsilon(\lambda),$$

one said that the two distributions $\mathcal{X}$ *and* $\mathcal{Y}$ *are indistinguishable. Here,* $\mathcal{A}$ *is a probabilistic polynomial-time adversary. Specifically, when* $\epsilon(\lambda) = \mathrm{negl}(\lambda)$*, one called* $\mathcal{X}$ *and* $\mathcal{Y}$ *indistinguishable with respect to* ϵ*; when* $\epsilon(\lambda) = 2^{-\lambda^c}$*, one called* $\mathcal{X}$ *and* $\mathcal{Y}$ *sub-exponentially indistinguishable.*

Definition 10 (Circuit Obfuscation). *A circuit obfuscation scheme under the ideal model with an oracle* $\mathcal{O}$ *is said to be efficient* $\mathrm{Obf}^{\mathcal{O}}(\lambda, C)$ *if, for a given input circuit* C*, it outputs an obfuscated circuit* $\widehat{C}^\bullet$*. The scheme is required to be correct, meaning that for all* $\lambda \in \mathbb{N}$*, where the circuit* $C : \{0,1\}^D \to \{0,1\}^*$ *and input* $x \in \{0,1\}^D$*, the following relation holds:*

$$\Pr[\widehat{C}^\bullet \leftarrow \mathrm{Obf}^{\mathcal{O}}(\lambda, C) : \widehat{C}^{\mathcal{O}} = C(x)] = 1.$$

Definition 11 (Ideal Obfuscation). *A circuit obfuscation scheme* $\mathrm{Obf}^{\mathcal{O}}(\lambda, C)$ *is said to be ideal if there exists an efficient simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ *such that for all adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the adversary's advantage is negligible, i.e.,*

$$\Pr\left[\begin{array}{c} C \leftarrow \mathcal{A}_1^{\mathcal{O}}(\lambda) \\ \widehat{C}^\bullet \leftarrow \mathrm{Obf}^{\mathcal{O}}(\lambda, C) \end{array} : \mathcal{A}_2^{\mathcal{O}}(\widehat{C}^\bullet) = 1\right] - \Pr\left[\begin{array}{c} C \leftarrow \mathcal{A}_1^{\mathcal{S}_1}(\lambda) \\ \widetilde{C}^\bullet \leftarrow \mathcal{S}_2^{C}(\lambda, D, S) \end{array} : \mathcal{A}_2^{\mathcal{S}_3^C}(\widetilde{C}^\bullet) = 1\right].$$

Here, $D = |x|$ *is the length of the input circuit* C*, and* $S = |C|$ *is the size of the circuit* C*.*

3 Evasive LWR and Multilinear Mapping

3.1 Evasive LWR

Definition 12 (Solution Evasive LWR). *For* $A_i, S_i \in \mathbb{Z}_q^{n\times n}$*,* $i = 1, \ldots, \ell$*, where* A_i^{-1} *denotes the inverse of matrix* A_i *for* $i = 2, \ldots, \ell$*, and* $u \in \mathbb{Z}_q^n$*, the Solution Evasive LWR problem refers to finding* S_i*,* $i = 1, \ldots, \ell$*, from* $(\lfloor uS_1A_1 \rfloor_p,$ $\lfloor A_1^{-1}S_2A_2 \rfloor_p, \ldots, \lfloor A_{\ell-1}^{-1}S_\ell A_\ell \rfloor_p)$*.*

Theorem 3. *For $A_i, S_i \in \mathbb{Z}_q^{n\times n}$, $i = 1, \ldots, \ell$, where A_i^{-1} denotes the inverse of matrix A_i for $i = 2, \ldots, \ell$, and $u \in \mathbb{Z}_q^n$, we have*

$$(\{A_i\}_{i=1}^{\ell}, \lfloor \boxed{kS_1A_1} \rfloor_p, \{\lfloor \boxed{A_{i-1}^{-1}S_iA_i} \rfloor_p\}_{i=2}^{\ell}) \approx_C (\{A_i\}_{i=1}^{\ell}, \lfloor u_1 \rfloor_p, \{\lfloor U_i \rfloor_p\}_{i=2}^{\ell}),$$

as well as

$$\left(A_1, \left\lfloor u \left(\prod_{i=1}^{\ell} S_1\right) A_1 \right\rfloor_p\right) \approx_C \left(A_1, \left\lfloor u_1 \prod_{i=1}^{\ell} U_i \right\rfloor_p\right).$$

Proof. Here is the translation of the provided text into English:

First, let

$$O(n) = \frac{n \log_2 n \ln p}{\sqrt{5}}.$$

On one hand, according to Theorem 2, it is known that

$$|\Pr(\lfloor b_1 \rfloor_p = \lfloor \boxed{uS_1A_1} \rfloor_p) - \Pr(\lfloor u_1 \rfloor_p)| \le 2 \cdot e^{-O(n)} + e^{-2O(n)},$$

and

$$|\Pr(\lfloor B_i \rfloor_p = \lfloor \boxed{A_{i-1}^{-1}S_iA_i} \rfloor_p) - \Pr(\lfloor U_i \rfloor_p)| \le 2 \cdot e^{-O(n)} + e^{-2O(n)},$$

thus we can obtain

$$\begin{aligned}
&(\{A_i\}_{i=1}^{\ell}, \lfloor \boxed{uS_1A_1} \rfloor_p, \{\lfloor \boxed{A_{i-1}^{-1}S_iA_i} \rfloor_p\}_{i=2}^{\ell}) \\
&\qquad \approx_C (\{A_i\}_{i=1}^{\ell}, \lfloor u_1 \rfloor_p, \{\lfloor \boxed{A_{i-1}^{-1}S_iA_i} \rfloor_p\}_{i=2}^{\ell}) \\
&\qquad \approx_C (\{A_i\}_{i=1}^{\ell}, \lfloor u_1 \rfloor_p, \lfloor U_2 \rfloor_p, \{\lfloor \boxed{A_{i-1}^{-1}S_iA_i} \rfloor_p\}_{i=3}^{\ell}) \\
&\qquad \vdots \\
&\qquad \approx_C (\{A_i\}_{i=1}^{\ell}, \lfloor u_1 \rfloor_p, \{\lfloor U_i \rfloor_p\}_{i=2}^{\ell}).
\end{aligned}$$

On the other hand, using the same principle, it can be derived that

$$\left| \Pr\left(\left(A_1, \left\lfloor u \left(\prod_{i=1}^{\ell} S_i\right) A_1 \right\rfloor_p\right)\right) - \Pr\left(\left(A_1, \left\lfloor u_1 \prod_{i=1}^{\ell} U_i \right\rfloor_p\right)\right) \right| \le \ell' \cdot e^{-O(n)},$$

$\ell' \in (\ell, 2\ell)$.

Definition 13 (Evasive LWR Decision). *For $A_i, S_i \in \mathbb{Z}_q^{n\times n}$, $i = 1, \ldots, \ell$, where A_i^{-1} denotes the inverse of matrix A_i for $i = 2, \ldots, \ell$, and $u \in \mathbb{Z}_q^n$,* *the Evasive LWR decision problem refers to distinguishing, in polynomial time, between the distributions*

$$(\lfloor b_1 = uS_1A_1\rfloor_p, \lfloor B_2 = A_1^{-1}S_2A_2\rfloor_p, \ldots, \lfloor B_\ell = A_{\ell-1}^{-1}S_\ell A_\ell\rfloor_p)$$

and

$$(\lfloor u_1\rfloor_p, \lfloor U_2\rfloor_p, \ldots, \lfloor U_\ell\rfloor_p),$$

as well as between the distributions

$$\left(A_1, \left\lfloor u\left(\prod_{i=1}^{\ell} S_i\right)A_1\right\rfloor_p\right) \text{ and } \left(A_1, \left\lfloor u_1\prod_{i=1}^{\ell} U_i\right\rfloor_p\right).$$

3.2 Evasive LWR-base Multilinear Mapping

Algorithm 2 Composite homomorphic pseudorandom function

PRF.KeyGen(n, m, q). Generate necessary parameters.

PRF.Enc($\{M_i\}_{i=1}^{\ell}, key, n, m, q$).

$$C_i = PRF(\{M_i\}_{i=1}^{\ell}, key)$$

PRF.Eval($C = (C_1, \ldots, C_\ell)$).

- Set S_i as

$$\widehat{S}_i = \begin{cases} u(C_1), \text{ if } i = 1, \\ C_i, \quad\ \ \text{if } i > 1. \end{cases}$$

- Output encrypted result

$$\{\lfloor u(C_1)A_1\rfloor_p\}, \{A_{i-1}^{-1}\lfloor (C_i)A_i\rfloor_p\}_{i=2}^{\ell}.$$

Theorem 4. *Composite homomorphic pseudorandom function is a type of graded encoding system.*

Proof. A usable asymmetric graded encoding system comprises the following effective algorithms: Instance Generation, Ring Element Sampling, Encoding, Addition and Multiplication Encodings, Zero Testing, and Extraction, denoted as $\mathcal{GES} = (\texttt{InstGen}, \texttt{samp}, \texttt{enc}, \texttt{add}, \texttt{neg}, \texttt{mul}, \texttt{isZero}, \texttt{ext})$.

Instance Generation. Generates keys for ciphertext fully homomorphic pseudorandom functions, as well as algorithms like Enc and Eval. Security parameters n, and noise ε such that $\|\varepsilon\| \leq \texttt{B}$.

Encoding. For input plaintext to be encrypted $\{m_i\}_{i=1}^{\ell}$, $\ell \leq \aleph$, use the fully homomorphic encryption algorithm $Enc(m_i, key)$ to obtain ciphertext $\{c_i\}_{i=1}^{\ell}$.

Addition and Multiplication Encodings. According to the definition of fully homomorphic encryption schemes or ciphertext fully homomorphic pseudorandom functions: $c_a + c_b = Enc(m_a + m_b, key)$, $c_a \cdot c_b = Enc(m_a \cdot m_b, key)$, $-c_a = Enc(-m_a, key)$.

Furthermore, for $\{m_i\}_{i=1}^{\ell}$, we have that

$$\sum_{i=1}^{\aleph} c_i = Enc(\sum_{i=1}^{\aleph} m_i, key) + \sum_{i=1}^{\aleph} \varepsilon_i, \left\| \sum_{i=1}^{\aleph} \varepsilon_i \right\| \leq \mathtt{B}.$$

Additionally, we know that

$$\prod_{i=1}^{\aleph} c_i = Enc(\prod_{i=1}^{\aleph} m_i, key) + \varXi, \|\varXi\| \leq \mathtt{B}.$$

Zero Testing. Since $Enc(0, key) = Enc(m_a - m_a, key) = Enc(m_a, key) + Enc(-m_a, key) = c_a - c_a \leq \mathtt{B}$, it's easy to prove that zero testing holds.

Extraction. For $\aleph$-level encodings c, c', we have

$$\|c - c'\| = \|Enc(m, key) + \varepsilon - Enc(m, key) - \varepsilon'\| \leq \mathtt{B}.$$

4 Linear Homomorphic Encryption Scheme based on LWR Variant Problems

Scheme 3 LHE Scheme based on LWR Problem

LWR.DV.KeyGen(n, m, q). Choose a random vector $r \in \{0,1\}^n$ and matrices $A, l \in_R \mathbb{Z}_q^{m \times n}$. Let $u = H(r) \in \mathbb{Z}_q$, output sample $(\odot_q(A, lu), T_{sk}) \leftarrow$ GenTrap(n, m, q), let $pk = (r, \odot_q(A, l))$ and $sk = (\odot_q(A, lu), T_{sk})$.

LWR.DV.Enc(pk, s, q, p). Let $u = H(r) \in \mathbb{Z}_q$. For each element a_{ij}^l of $\odot_q(A, l)$, compute $(a_{ij}^l)^u \bmod q = a_{ij}^{lu}$, thus obtaining $\odot_q(A, lu)$. For plaintext $s \in \{0,1\}^n$, choose a random vector $k \in \{0, \ell+1\}^n$, output $c = \lfloor \odot_q(A, lu)(s + k) \rfloor_p$.

LWR.DV.Eval$(pk, q, p, f, (c_1, \ldots, c_\ell))$. Input ciphertext vector $(c_1, \ldots, c_\ell)$ and linear function $g = (\alpha_1, \ldots, \alpha_\ell) \in \{0,1\}^\ell$, compute

$$c = \sum_{i=1}^{\ell} \alpha_i c_i \bmod q.$$

LWR.DV.PDec(sk, c). For ciphertext $c \in \mathbb{Z}_p^n$, output $\tilde{s} = \text{LWRInvert}(T_{sk}, \odot_q(A, lu), c)$, for each component $\tilde{s}_i$ of $\tilde{s}$,

$$\begin{cases} \tilde{k}_i = 0, & \text{when } \tilde{s}_i \in \{0, 1\}, \\ \tilde{k}_i = \tilde{s}_i, & \text{when } \tilde{s}_i \in \{(\ell+1), \ldots, n(\ell+1)\}, \\ \tilde{k}_i = \tilde{s}_i - 1, & \text{when } \tilde{s}_i \notin \{0, 1, (\ell+1), \ldots, n(\ell+1)\}. \end{cases}$$

Return $\rho = \left(sk, \tilde{k} = (\tilde{k}_i)_{i \in \{1, \ldots, \ell\}}\right)$.

LWR.DV.Rec(ρ, c). For ciphertext $c \in \mathbb{Z}_p^n$, output $\tilde{s} = \text{LWRInvert}(T_{sk}, \odot_q(A, lu), c)$, decrypt $s' = \tilde{s} - \tilde{k}$.

Simulatable Decryption Hint. For given ciphertext c and plaintext message $\tilde{s}$ (where c and $\tilde{s}$ are unrelated), choose $\tilde{k} \in_R \{0, \ell+1, \ldots, n(\ell+1)\}^n$, $\tilde{u} \in_R \mathbb{Z}_q$. Let $\tilde{sk} \leftarrow \text{GenTrap}(n, m, q)$, compute simulated ciphertext $\tilde{c}$ and

$$\tilde{c}_i = \left| \left(c - \lfloor \odot_q(A, l\tilde{u})(\tilde{s} + \tilde{k}) \rceil_p \right)_i \right|, i \in \{1, \ldots, n\}.$$

Then output $\tilde{\rho} = (\tilde{sk}, \tilde{k})$.

5 Splitting Fully Homomorphic Encryption Scheme

Next, we will introduce an instantiation of FHE with split decryption. First propose a scheme based on standard assumptions, which assumes the existence of a structured version of a random oracle, and then present a trusted candidate scheme for this oracle.

5.1 Defining a Special Oracle for Constructing Splitting Fully Homomorphic Schemes

Before presenting the split fully homomorphic scheme, define a special oracle. The parameters of this oracle are $(\widehat{pk}, \overline{pk}, q, \tilde{q})$, where the input is a string $x \in \{0, 1\}^*$, and it uniformly outputs encrypted values for LHE and FHE. The oracle is deterministic and accessible to all parties, so when given the same input x, the oracle always outputs the same pair of ciphertexts. The formal definition of this oracle is as follows.

Definition 14 ([BDGM20]). $\mathcal{O}_{(\widehat{pk},\overline{pk},q,\tilde{q})}$*: Given input string* $x \in \{0,1\}^*$*, outputs two ciphertexts that are uniformly distributed:*

$$\overline{\mathrm{Enc}}(\overline{pk}, s) \text{ and } \widehat{\mathrm{Enc}}(\widehat{pk}, -\lfloor s/\tilde{q} \rfloor \cdot \tilde{q})$$

where $s \leftarrow \mathbb{Z}_q$.

The oracle $\mathcal{O}_{(\widehat{pk},\overline{pk},q,\tilde{q})}$ can encrypt the private key of FHE using LHE scheme, and the resulting ciphertexts follow a uniform distribution. This is because we use the decryption and multiplication algorithms DEC&Mult in the FHE scheme to compute $\overline{\mathrm{Enc}}(\overline{pk}, s - \lfloor s/\tilde{q} \rfloor \cdot \tilde{q} + \text{noise})$, where the noise is the decryption noise of the FHE scheme. By choosing appropriate parameters $\tilde{q}$, we can achieve

$$\begin{aligned} \overline{\mathrm{Enc}}(\overline{pk}, s - \lfloor s/\tilde{q} \rfloor \cdot \tilde{q} + \text{noise}) &= \overline{\mathrm{Enc}}(\overline{pk}, (s \bmod \tilde{q}) + \text{ noise}) \\ &\approx_s \overline{\mathrm{Enc}}(\overline{pk}, (s \bmod \tilde{q})). \end{aligned}$$

Thus, one obtained ciphertexts that are statistically indistinguishable through the two encryption systems.

Description. Now, provide a formal description of our scheme. We assume the existence of the following primitives:

- **cPRF** = ($\widehat{\mathbf{KeyGen}}, \widehat{\mathbf{Enc}}, \widehat{\mathbf{Eval}}, \widehat{\mathbf{Dec}}$) with linear decryption-multiplication and noise constraint $\mathtt{B}$.
- **LHE** = ($\overline{\mathbf{KeyGen}}, \overline{\mathbf{Enc}}, \overline{\mathbf{Eval}}, \overline{\mathbf{PDec}}, \overline{\mathbf{Rec}}$) with small decryption hints and simulatable decryption hints, then we refer to LHE as linear homomorphic encryption.

If the underlying FHE scheme is leveled out, then it will result in split FHE. Conversely, if the FHE scheme supports evaluation of unbounded circuits, then the resultant split FHE construction will also do so. The formal description of this scheme is as follows.

Scheme 4 Split Homomorphic Encryption Scheme

KeyGen(n, m, q**)**. Given security parameter n, output sample $(\overline{sk}, \overline{pk}) \leftarrow \overline{\mathrm{KeyGen}}(n)$. Let $\mathbb{Z}_q$ be the plaintext space under LHE definition, output sample $(\widehat{sk}, \widehat{pk}) \leftarrow \widehat{\mathrm{KeyGen}}(n, m, q)$.
Let $\widehat{sk} = (T_1, \ldots, T_n) \in \{0,1\}^{n \times n}$, then return

$$sk = \overline{sk} \text{ and } pk = (\widehat{pk}, \overline{pk}, \bar{c}_1, \ldots, \bar{c}_n).$$

where, for any $i \in [n]$, define $\bar{c}_i \leftarrow \overline{\mathrm{Enc}}(\overline{pk}, T_i)$.

Enc(pk, s). Return the ciphertext

$$c \leftarrow \widehat{\mathrm{Enc}}(\widehat{pk}, s).$$

Eval($pk, f, (c_1, \dots, c_\ell)$). Given a circuit $\mathcal{C}$ of ℓ bits and ciphertexts of length k bits $(c_1, \dots, c_\ell))$. For any $j \in [k]$, $\mathcal{C}_j$ is the j-th component of circuit $\mathcal{C}$, calculate

$$d_j \leftarrow \widehat{\mathrm{Eval}}(\widehat{pk}, C_j, (c_1, \dots, c_\ell)).$$

Define the linear function over $\mathbb{Z}_q$ as

$$g(x_1, \dots, x_n) = \sum_{j=1}^{k} \mathrm{DEC\&Mult}\left((x_1, \dots, x_n), d_j, 2^{\lceil \log(\tilde{q}+(k+1)\mathtt{B}) \rceil + j}\right).$$

Compute $d \leftarrow \overline{\mathrm{Eval}}(\overline{pk}, g, (\overline{c}_1, \dots, \overline{c}_n))$, then query $(a, \tilde{a}) \leftarrow \mathcal{O}_{(\widehat{pk}, \overline{pk}, q, \tilde{q})}(d)$ and define the following linear function

$$\tilde{g}(x_1, \dots, x_n, x_{n+1}, x_{n+2}) = \mathrm{DEC\&Mult}((x_1, \dots, x_n), \tilde{a}, 1) + x_{n+1} + x_{n+2}.$$

Return

$$c \leftarrow \overline{\mathrm{Eval}}(\overline{pk}, \tilde{g}, (\overline{c}_1, \dots, \overline{c}_n), d, a).$$

PDec(sk, c). Given an evaluable ciphertext c, return

$$\rho \leftarrow \overline{\mathrm{PDec}}(\overline{sk}, c).$$

Rec(ρ, c). Given an evaluable ciphertext c, return

$$\tilde{s} \leftarrow \overline{\mathrm{Rec}}(\rho, c),$$

and return the binary representation of $\tilde{s}$ without the $\lceil \log(\tilde{q} + (k+1)\mathtt{B}) \rceil$ least significant bits.

Analysis: During the analysis, set parameters as needed to ensure the scheme can decrypt correctly. Subsequently, demonstrate that our choices lead to a set of satisfiable constraints. These constraints satisfy the conditions of the underlying hard problems, thus the hardness problem assumptions still hold. The following theorem establishes correctness.

Theorem 5 (Correctness of Split Homomorphic Encryption Scheme). *Let* $q \geq 2^k + 2^{\lceil \log(\tilde{q}+(k+1)\mathtt{B}) \rceil}$. *Assuming that* FHE *and* LHE *are correct, then* **Scheme 4** *satisfies the correctness of split homomorphism.*

Proof. We rewrite

$$\tilde{s} = \overline{\mathrm{Rec}}(\rho, c) = \overline{\mathrm{Rec}}(\overline{\mathrm{PDec}}(\overline{sk}, c), c),$$

where $c = \overline{\mathrm{Eval}}(\overline{pk}, \tilde{g}, (\bar{c}_1, \dots, \bar{c}_n), d, a))$. By the correctness of the LHE scheme, we can rewrite d as

$$\begin{aligned} d &= \overline{\mathrm{Eval}}(\overline{pk}, g, (\bar{c}_1, \dots, \bar{c}_n)) \\ &= \overline{\mathrm{Eval}}(\overline{pk}, g, (\overline{\mathrm{Enc}}(\overline{pk}, T_1), \dots, \overline{\mathrm{Enc}}(\overline{pk}, T_n))) \\ &= \overline{\mathrm{Enc}}\left(\overline{pk}, \sum_{j=1}^{k} \mathrm{DEC\&Mult}\left((T_1, \dots, T_n), d_j, 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j}\right)\right). \end{aligned}$$

Where

$$d_j = \widehat{\mathrm{Eval}}(\widehat{pk}, C_j, (c_1, \dots, c_\ell))$$

and $c_i = \widehat{\mathrm{Enc}}(\widehat{pk}, s_i)$. Therefore, by the correctness of the FHE scheme for decryption-multiplication, we can rewrite as

$$\begin{aligned} d &= \overline{\mathrm{Enc}}\left(\overline{pk}, \sum_{j=1}^{k} \mathrm{DEC\&Mult}\left((T_1, \dots, T_n), d_j, 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j}\right)\right) \\ &= \overline{\mathrm{Enc}}\left(\overline{pk}, \sum_{j=1}^{k} 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j} \cdot C_j(s_1, \cdots, s_\ell) + \underbrace{\sum_{j=1}^{k} e_j}_{\tilde{e}}\right). \end{aligned}$$

Let $r \leftarrow \mathbb{Z}_q$ and define the oracle $\mathcal{O}_{(\widehat{pk}, \overline{pk}, q, \tilde{q})}$ such that $a = \overline{\mathrm{Enc}}(\overline{pk}, r)$ and

$$\tilde{g}(x_1, \dots, x_n, x_{n+1}, x_{n+2}) = \mathrm{DEC\&Mult}((x_1, \dots, x_n), \tilde{a}, 1) + x_{n+1} + x_{n+2}.$$

Where, $\tilde{a} = \widehat{\mathrm{Enc}}(\widehat{pk}, -\lfloor r/\tilde{q} \rfloor \cdot \tilde{q})$. Then by the correctness of the FHE scheme, and $c = \overline{\mathrm{Enc}}(\overline{pk}, \tilde{s})$, where $\tilde{s}$ is

$$\begin{aligned} \tilde{s} &= \mathrm{DEC\&Mult}\left((T_1, \dots, T_n), \tilde{a}, 1\right) + \sum_{j=1}^{k} 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j} \cdot C_j(s_1, \cdots, s_\ell) + \tilde{e} + r \\ &= -\lfloor r/\tilde{q} \rfloor \cdot \tilde{q} + e + \sum_{j=1}^{k} 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j} \cdot C_j(s_1, \cdots, s_\ell) + \tilde{e} + r \\ &= \sum_{j=1}^{k} 2^{\lceil \log(\tilde{q}+(k+1)\mathsf{B})\rceil + j} \cdot C_j(s_1, \cdots, s_\ell) + \tilde{e} + e + \underbrace{r \bmod \tilde{q}}_{\tilde{r}}. \end{aligned}$$

Note that an upper bound for $\tilde{e} + e$ is $(k+1) \cdot \mathtt{B}$, and $\tilde{r}$ is a small perturbation due to the modulo $\tilde{q}$. This means that the output of the circuit is encoded as a high-order bit $\tilde{s}$ with probability 1 when q is sufficiently large.

Theorem 6 (Security of Split Homomorphic Encryption Scheme). *Let $q \geq 2^k + 2^{\lceil \log(\tilde{q}+(k+1)\mathtt{B}) \rceil}$. Assuming that the FHE scheme and the LHE scheme are secure schemes, then* **Scheme 4** *satisfies the security model $\mathcal{O}_{(\widehat{pk},\overline{pk},q,\tilde{q})}$ for split homomorphism.*

Proof. Assume $(s_0, s_1, C_1, \ldots, C_\beta)$ is the adversary's input chosen at the beginning of the generation of system π.

Hybrid $\mathcal{H}_0$: Define the following original system. The challenger generates a distribution using a random coin toss as follows:

$$(pk, c = \widehat{\mathrm{Enc}}(\widehat{pk}, s_\delta), \rho_1, \ldots, \rho_\beta).$$

Where

$$pk = (\widehat{pk}, \overline{pk}, \overline{\mathrm{Enc}}(\overline{pk}, T_1), \ldots, \overline{\mathrm{Enc}}(\overline{pk}, T_n)),$$

and ρ_i is obtained from $\mathrm{PDec}(sk, \mathrm{Eval}(pk, C_i, c))$.

Hybrids $\mathcal{H}_1, \ldots, \mathcal{H}_\beta$: Let $\mathrm{Eval}(pk, C_i, c)$ generate $d^{(i)}$. The ith *Hybrids $\mathcal{H}_i$* is defined the same as *Hybrids $\mathcal{H}_{i-1}$* except for the input $d^{(i)}$ and the output a (or $\tilde{a}$) such that

$$c = \overline{\mathrm{Enc}}\left(\overline{pk}, \mathrm{ECC}(C_i(s_\delta)) + \tilde{e} + e + r - \lfloor r/\tilde{q} \rfloor \cdot \tilde{q}\right),$$

where ECC is the high-order bit encoding defined in the homomorphic encryption part, $\tilde{e} + e$ is the decryption noise after homomorphic computation $(d^{(1)}, \ldots, d^{(k)}, \tilde{a})$, $r \leftarrow \mathbb{Z}_q$, $\tilde{\rho}_i$ is the "decryption tweak" obtained using random coin toss a, which can be used to decrypt the ciphertext c.

Note that the decryption noise $\tilde{e} + e$ can be efficiently calculated using the FHE scheme key, therefore $\tilde{\rho}_i$ can also be computed in polynomial time. The ciphertext distributions of *Hybrids $\mathcal{H}_1, \ldots, \mathcal{H}_\beta$* are consistent, with the only difference being the specific form of $\tilde{\rho}_i$. This is because the LHE scheme has simulatable decryption tweaks, so the distribution of $\mathcal{H}_i$ is consistent with the distribution of $\mathcal{H}_{i-1}$, i.e.,

$$\begin{aligned}&(pk, \widehat{\mathrm{Enc}}(\widehat{pk}, s_\delta), \tilde{\rho}_1, \ldots, \tilde{\rho}_{i-1}, \rho_i, \rho_{i+1}, \ldots, \rho_\beta)\\ &= (pk, \widehat{\mathrm{Enc}}(\widehat{pk}, s_\delta), \tilde{\rho}_1, \ldots, \tilde{\rho}_{i-1}, \tilde{\rho}_i, \rho_{i+1}, \ldots, \rho_\beta).\end{aligned}$$

Hybrids $\mathcal{H}_{\beta+1},\ldots,\mathcal{H}_{2\beta}$: The $\beta+i$th Hybrids and the previous β Hybrids are different mainly in a, i.e.,

$$\begin{aligned} c &= \overline{\mathrm{Enc}}\left(\overline{pk}, \mathrm{ECC}(C_i(s_\delta)) + \tilde{e} + e + \lfloor r/\tilde{q} \rfloor \cdot \tilde{q} + \tilde{r} - \lfloor r/\tilde{q} \rfloor \cdot \tilde{q}\right) \\ &= \overline{\mathrm{Enc}}\left(\overline{pk}, \mathrm{ECC}(C_i(s_\delta)) + \tilde{e} + e + \tilde{r}\right). \end{aligned}$$

Where, $\tilde{r} \leftarrow \mathbb{Z}_{\tilde{q}}$. Note that the distributions caused by these two Hybrids are different only when $r \in R$, where $R := \{q-(q \bmod \tilde{q}), \ldots, q\}$. Because $\tilde{q}/q \leq 2^{-\lambda}$, these two distributions to be statistically close.

Hybrids $\mathcal{H}_{2\beta+1},\ldots,\mathcal{H}_{3\beta}$: The $2\beta+i$th Hybrids are defined the same as the previous ones, except for the value of a, i.e.,

$$c = \overline{\mathrm{Enc}}(\overline{pk}, \mathrm{ECC}(C_i(s_\delta)) + \tilde{y}).$$

Where the noise $\tilde{e}$ can be neglected in the calculation, therefore it is not reflected in the above equation. The difference between this and the previous Hybrids lies in whether the ciphertext contains $\tilde{e}+e$. Since an upper bound of the noise $\tilde{e}+e$ is $(k+1)\cdot \mathtt{B}$, and $\tilde{q} \geq 2^\lambda \cdot (k+1) \cdot B$, according to Lemma 1, the distribution caused by this Hybrids is statistically indistinguishable from the previous one.

Hybrids $\mathcal{H}_{3\beta+1},\ldots,\mathcal{H}_{3\beta+n}$: The $3\beta+i$th Hybrids are defined the same as the previous ones, except that the ciphertext $c_{(\mathrm{LHE},i)}$ is derived from encrypting 0 with the public key. At this point, the LHE scheme key no longer contributes to $(\tilde{\rho}_1,\ldots,\tilde{\rho}_\beta)$, so use indistinguishability to demonstrate the semantic security of these Hybrids.

$$\begin{aligned} &\begin{pmatrix} \overline{\mathrm{Enc}}(\overline{pk},0),\ldots,\overline{\mathrm{Enc}}(\overline{pk},0),\overline{\mathrm{Enc}}(\overline{pk},T_i), \\ \overline{\mathrm{Enc}}(\overline{pk},T_{i+1}),\ldots,\overline{\mathrm{Enc}}(\overline{pk},T_n) \end{pmatrix} \\ \approx_c &\begin{pmatrix} \overline{\mathrm{Enc}}(\overline{pk},0),\ldots,\overline{\mathrm{Enc}}(\overline{pk},0),\overline{\mathrm{Enc}}(\overline{pk},0), \\ \overline{\mathrm{Enc}}(\overline{pk},T_{i+1}),\ldots,\overline{\mathrm{Enc}}(\overline{pk},T_n) \end{pmatrix}. \end{aligned}$$

Hybrids $\mathcal{H}^{(0)}_{3\beta+n},\ldots,\mathcal{H}^{(b)}_{3\beta+n}$: Fix the length of the challenge plaintext to i, and use the symbol $\mathcal{H}^{(i)}_{3\beta+n}$ to represent the Hybrids at this point. The distribution of this Hybrids is

$$(pk, c = \widehat{\mathrm{Enc}}(\widehat{pk}, s_i), \tilde{\rho}_1, \ldots, \tilde{\rho}_\beta),$$

where

$$pk = (\widehat{pk}, \overline{pk}, \overline{\mathrm{Enc}}(\overline{pk},0),\ldots,\overline{\mathrm{Enc}}(\overline{pk},0)).$$

Because the FHE scheme key is no longer encoded in the public parameters, there is no need to compute $(\tilde{\rho}_1, \ldots, \tilde{\rho}_\beta)$. Therefore, any advantage that the adversary has in distinguishing $\mathcal{H}^{(0)}_{3\beta+n}$ and $\mathcal{H}^{(1)}_{3\beta+n}$ cannot be greater than distinguishing $\widehat{\mathrm{Enc}}(\widehat{pk}, s_0)$ and $\widehat{\mathrm{Enc}}(\widehat{pk}, s_1)$. Therefore, the FHE scheme is computationally indistinguishable, thus proving the semantic security of the sFHE scheme.

5.2 Instantiation of Oracle Model

To complete the description of our scheme, we discuss some candidate instantiations $\mathcal{O}_{(\widehat{pk},\overline{pk},q,\tilde{q})}$ of the oracle. We require the underlying LHE scheme to have a dense ciphertext space. We introduced the cyclic assumption introduced by Brakerski et al. [BDGM20] bridging the gap between FHE and LHE schemes. The oracle machine shown in Theorem 6 is just one of them, which is a special program obfuscation that enables the realization of split fully homomorphic schemes. Next, we introduce another oracle constructed by Brakerski et al. [BDGM20].

Simple Candidate Quantum Oracle. Let $\mathcal{C}$ be the ciphertext space of LHE. The first instantiation is to take the encryption algorithm in FHE and encrypt the key in LHE, $\widehat{c} \leftarrow \widehat{\mathrm{Enc}}(\widehat{pk}, \overline{sk})$. Extract the ciphertext hash value of the homomorphic operation obtained through a hash function, which is used to fix the random coin in the algorithm. LHE ciphertext is sampled without knowing the underlying plaintext (which is why we need dense ciphertext), while FHE terms are calculated by homomorphically evaluating the decryption circuit and rounding the resulting message to the nearest multiple of $\tilde{q}$.

Let $D = (D_a)_{a\in\mathcal{C}}$, where D_a is a set in the Hilbert space $\mathcal{H}_{D_a} = \mathbb{C}[\{0,1\}^n \cup \{\perp\}]$. The Hilbert space $\mathcal{H}_{D_a}$ can be seen as a space spanned by a set of orthogonal bases $|b\rangle$, where $b \in \{0,1\}^n \cup \{\perp\}$. Let the unitary transformation U be defined as

$$U|\perp\rangle = |\psi_0\rangle, U|\psi_0\rangle = |\perp\rangle \text{ and } U|\psi_b\rangle = |\psi_b\rangle, \forall b \in \{0,1\}^n \setminus \{0\}^n.$$

where $|\psi_b\rangle := H|b\rangle$, and H is the Hadamard transform on $\mathbb{C}[\{0,1\}^n] = (\mathbb{C}^2)^{\otimes n}$. Let $|b\rangle = 2^{-n/2}\sum_\eta (-1)^{\eta\cdot b}|\psi_\eta\rangle$, then we have

$$U|b\rangle = |b\rangle + 2^{-n/2}(|\perp\rangle - |\psi_0\rangle).$$

When the oracle is queried, the unitary transformation O_{XYZ} acts on the query register X and Y, and the database register D, with the specific expression

$$O_{XYZ} = \sum_a |a\rangle\langle a| \otimes O^a_{YD_a} \text{ and } O^a_{YD_a} = U_{D_a}\mathrm{CNOT}_{YD_a}U_{D_a}.$$

where $\mathrm{CNOT}|b\rangle|b_a\rangle = |b\rangle|b \oplus b_a\rangle$, $b, b_a \in \{0,1\}^n$ and $\mathrm{CNOT}|b\rangle|\bot\rangle = |b\rangle|\bot\rangle$. With these tools, present Don et al.'s quantum hash oracle model as follows:

$$y := \max_{a\in\mathcal{C}} |\{b \in \{0,1\}^n | \langle a,b\rangle \in \mathbb{R}\}|, \quad \tilde{y} \leftarrow \widehat{\mathrm{Eval}}(\widehat{pk}, -\lfloor \overline{\mathrm{Dec}}(\cdot, y)/\tilde{q}\rfloor \cdot \tilde{q}, \widehat{c})$$

Additionally, consider the following projector:

$$\Pi^a_{D_a} := \sum_{\substack{b\ s.t.\\ \langle a,b\rangle\in\mathbb{R}}} |b\rangle\langle b|_{D_a} \text{ and } \Pi^{\emptyset}_{D_a} := \mathbb{1}_D - \sum_{a\in\mathcal{X}} \Pi^a_{D_a} = \bigotimes_{a\in\mathcal{X}} \bar{\Pi}^a_{D_a}.$$

where $\bar{\Pi}^a_{D_a} := \mathbb{1}_{D_a} - \Pi^a_{D_a}$. Furthermore, define the measurement $\mathcal{M} = \mathcal{M}^R$, and the following projector

$$\Sigma^a := \bigotimes_{a'<a} \bar{\Pi}^{a'}_{D_{a'}} \otimes \Pi^a_{D_a} \text{ and } \Sigma^{\emptyset} := \mathbb{1} - \sum_{a'} \Sigma^{a'} = \bigotimes_{a'} \bar{\Pi}^{a'}_{D_{a'}} = \Pi^{\emptyset}.$$

In addition, define the pure state measurement unitary transformation $M_{DP} = M^R_{DP} \in L(\mathcal{H}_D \otimes \mathcal{H}_R)$, i.e.,

$$M_{DP} := |\varphi\rangle_D|w\rangle_P \mapsto |\varphi\rangle_D|w+a\rangle_P.$$

Note that y is an element in the ciphertext domain of LHE, and its form is $y = \overline{\mathrm{Enc}}(\overline{pk}, s)$. For some $s \in \mathbb{Z}_q$, because LHE has a dense ciphertext domain. Furthermore, through the correctness of the FHE and LHE schemes, we have

$$\begin{aligned}
\tilde{y} &= \widehat{\mathrm{Eval}}(\widehat{pk}, -\lfloor \overline{\mathrm{Dec}}(\cdot, y)/\tilde{q}\rfloor \cdot \tilde{q}, \widehat{c}) \\
&= \widehat{\mathrm{Eval}}(\widehat{pk}, -\lfloor \overline{\mathrm{Dec}}(\cdot, y)/\tilde{q}\rfloor \cdot \tilde{q}, \widehat{\mathrm{Enc}}(\widehat{pk}, \overline{sk})) \\
&= \widehat{\mathrm{Enc}}(\widehat{pk}, -\lfloor \overline{\mathrm{Dec}}(\overline{sk}, y)/\tilde{q}\rfloor \cdot \tilde{q}) \\
&= \widehat{\mathrm{Enc}}(\widehat{pk}, -\lfloor s/\tilde{q}\rfloor \cdot \tilde{q}).
\end{aligned}$$

Therefore, it can be seen that the formation of $(y, \tilde{y})$ is based on the following assumptions.

Alternating Encryption Security. The cyclic dependency introduced by $\widehat{c} = \widehat{\mathrm{Enc}}(\widehat{pk}, \overline{sk})$ in the security of LHE and FHE schemes (e.g., the split FHE construction in this paper includes the encryption of $\widehat{sk}$ under $\overline{pk}$ in the public key) is considered a very mild assumption. Currently, it is the only known method to construct FHE from the LWE problem through bootstrapping theorems [Gen09b].

Perturbation. In the case of $y := \max_{a \in \mathcal{C}} |\{b \in \{0,1\}^n | \langle a, b \rangle \in \mathbb{R}\}|$, although $\tilde{y}$ is an FHE encryption of the correct value, it is not necessarily uniformly distributed. In particular, the randomness of $\tilde{y}$ may depend on the low-order bits of s in a complex way. In the specific case of LWR-based schemes, the noise term may carry information about s modulo $\tilde{q}$, which may introduce perturbation that interferes with decryption. However, the noise function is usually highly nonlinear, making it difficult to exploit. Therefore, we only consider the FHE.Eval algorithm.

Perturbation Elimination. Regarding the methods for eliminating the perturbation in LHE and FHE ciphertexts, we naturally think of ciphertext reprocessing techniques [DS16]: it can be expected that repeating bootstraping operations on FHE ciphertexts can eliminate the perturbation from LHE ciphertext noise. Unfortunately, our setting is different from the typical settings considered in the literature, as the ciphertext perturbation reprocessing algorithm must be executed by the distinguisher and cannot use private random coins. Although it seems difficult to formally analyze the effectiveness of these methods in our setting, we hope that these techniques may (at least heuristically) help mitigate the perturbation that interferes with decryption. This paper takes a different approach and provides a simple heuristic to alleviate perturbation. In short, the idea is to sample a set of random plaintexts and define a random string as the sum of a uniform subset $\mathcal{S}$ of these plaintexts. For the construction described earlier, Brakerski et al.'s instantiation includes a ciphertext $\widehat{c} = \widehat{\mathrm{Enc}}(\widehat{pk}, \overline{sk})$. The parameter $\sigma \in \mathrm{poly}(n, m, q, p)$ of the scheme is determined by the length of the set $\mathcal{S}$. The algorithm is presented randomly below, although this simplification can be easily bypassed using standard techniques (e.g., computing random coins using encrypted $\mathrm{Hash}(x)$).

$\mathcal{O}(\widehat{pk}, \overline{pk}, q, \tilde{q})(x)$: Input string $x \in \{0,1\}^*$ and a random set $\mathcal{S} \leftarrow \{0,1\}^\sigma$. For all $i \in [\sigma]$, when $\mathcal{S}_i = 1$, uniformly output sample

$$y_i := \max_{a \in \mathcal{C}} |\{b \in \{0,1\}^n | \langle a, b \rangle \in \mathbb{R}\}|,$$

when $\mathcal{S}_i = 0$, uniformly output sample $y_i \leftarrow \overline{\text{Enc}}(\overline{pk}, s_i)$, where s_i is any known plaintext message. Then compute

$$\tilde{y} \leftarrow \widehat{\text{Eval}}\left(\widehat{pk}, -\sum_{i=1}^{\sigma} \lfloor \overline{\text{Dec}}(\cdot, y)/\tilde{q} \rfloor \cdot \tilde{q}, \widehat{c}\right).$$

Let g be a linear function defined as follows

$$g(x_1, \ldots, x_{\mathcal{S}}) = \sum_{i \in \mathcal{S}} x_i + \sum_{i \notin \mathcal{S}} \lfloor x_i/\tilde{q} \rfloor \cdot \tilde{q}.$$

Then compute $\tilde{y} \leftarrow \overline{\text{Eval}}(\overline{pk}, g, \{y_i\}_{i \in \mathcal{S}})$ and return $(y, \tilde{y})$. By the correctness of homomorphic operations in the FHE scheme, it shown that

$$\tilde{y} = \widehat{\text{Eval}}\left(\widehat{pk}, -\sum_{i=1}^{\sigma} \lfloor \overline{\text{Dec}}(\cdot, y)/\tilde{q} \rfloor \cdot \tilde{q}, \widehat{c}\right) = \widehat{\text{Eval}}\left(\widehat{pk}, -\sum_{i=1}^{\sigma} \lfloor \overline{\text{Dec}}(\cdot, y)/\tilde{q} \rfloor \cdot \tilde{q}, \widehat{\text{Enc}}(\widehat{pk}, \overline{sk})\right)$$
$$= \widehat{\text{Enc}}\left(\widehat{pk}, -\sum_{i=1}^{\sigma} \lfloor \overline{\text{Dec}}(\overline{sk}, y)/\tilde{q} \rfloor \cdot \tilde{q}\right) = \widehat{\text{Enc}}\left(\widehat{pk}, -\sum_{i=1}^{\sigma} \lfloor s/\tilde{q} \rfloor \cdot \tilde{q}\right).$$

Combining with the correctness of the LHE scheme, one obtain

$$y = \overline{\text{Eval}}(\overline{pk}, g, \{y_i\}_{i \in \mathcal{S}}) = \overline{\text{Eval}}(\overline{pk}, g, \{\overline{\text{Enc}}(\overline{pk}, s_i)\}_{i \in \mathcal{S}})$$
$$= \overline{\text{Enc}}\left(\overline{pk}, \sum_{i \in \mathcal{S}} s_i + \sum_{i \notin \mathcal{S}} \lfloor s_i/\tilde{q} \rfloor \cdot \tilde{q}\right) = \overline{\text{Enc}}\left(\overline{pk}, \underbrace{\sum_{i \in \mathcal{S}} (s_i \bmod \tilde{q})}_{\tilde{s}} + \sum_{i \notin \mathcal{S}} \lfloor s_i/\tilde{q} \rfloor \cdot \tilde{q}\right).$$

6 Constructing Ideal Obfuscation using Homomorphic Splitting Encryption Scheme

6.1 Ideal Obfuscation

Scheme 5 Ideal Obfuscation Scheme

KeyGen(n, m, q). For $i \in [0, D)$, $j \in [0, B]$, randomly sample $k_{i,j} \leftarrow \{0,1\}^\lambda$ and compute

$$h_{i,j} = \text{Pr}\mathcal{O}(k_{i.j}, x).$$

Randomly sample $s_\varepsilon \leftarrow \{0,1\}^\lambda$. For $d \in [0, D]$, input security parameter n, output sample $(\overline{sk}_d, \overline{pk}_d) \leftarrow \overline{\text{KeyGen}}(n)$. Let $\mathbb{Z}_q$ be the plaintext space under

LHE definition, output sample $(\widehat{sk}_d, \widehat{pk}_d) \leftarrow \widehat{\text{KeyGen}}(n, m, q)$. Let $\widehat{sk}_d = (T_1, \ldots, T_n) \in \{0,1\}^{n \times n}$, then return

$$sk_d = \overline{sk}_d \text{ and } pk_d = (\widehat{pk}_d, \overline{pk}_d, \overline{c}_1, \ldots, \overline{c}_n).$$

where, for any $i \in [n]$, we define $\overline{c}_i \leftarrow \overline{\text{Enc}}(\overline{pk}_d, T_i)$.

Enc(pk_d, info$_\varepsilon$). For input $\text{info}_\varepsilon = (\text{normal}, \varepsilon, \{k_{i,j}\}_{i \in [0,D), j \in [1,B]}, s_\varepsilon)$, return

$$ct_\varepsilon \leftarrow \widehat{\text{Enc}}(\widehat{pk}_d, \text{info}_\varepsilon).$$

Eval($pk_d, f_d, (c_1, \ldots, c_\ell)$). f_d is provided later. Input circuit $\mathcal{C}$ of ℓ bits and ciphertext of length k bits $(c_1, \ldots, c_\ell)$. For any $j \in [k]$, where $\mathcal{C}_j$ is the j-th component of circuit $\mathcal{C}$, compute

$$\dot{d}_j \leftarrow \widehat{\text{Eval}}(\widehat{pk}_d, C_j, (c_1, \ldots, c_\ell)).$$

Define linear function over $\mathbb{Z}_q$ as

$$g(x_1, \ldots, x_n) = \sum_{j=1}^{k} \text{DEC\&Mult}\left((x_1, \ldots, x_n), \dot{d}_j, 2^{\lceil \log(\tilde{q} + (k+1)B) \rceil + j}\right).$$

Compute $\dot{d} \leftarrow \overline{\text{Eval}}(\overline{pk}_d, g, (\overline{c}_1, \ldots, \overline{c}_n))$, then query $(a, \tilde{a}) \leftarrow \mathcal{O}_{(\widehat{pk}_d, \overline{pk}_d, q, \tilde{q})}(\dot{d})$ and define the following linear function

$$\tilde{g}(x_1, \ldots, x_n, x_{n+1}, x_{n+2}) = \text{DEC\&Mult}((x_1, \ldots, x_n), \tilde{a}, 1) + x_{n+1} + x_{n+2}.$$

Output

$$ct_\varepsilon \leftarrow \overline{\text{Eval}}(\overline{pk}_d, \tilde{g}, (\overline{c}_1, \ldots, \overline{c}_n), \dot{d}, a).$$

Return the obfuscated circuit

$$\widehat{C} = (\{h_{i,j}\}_{i \in [0,D), j \in [1,B]}, ct_\varepsilon, \{sk_d\}_{d \in [0,D]}).$$

Eval&Expand. (normal mode)

- For $d \in [0, D)$, Eval&Expand encrypts $f_d(\text{normal}, \chi, \{k_{i,j}\}_{i \in [0,D), j \in [1,B]}, s_\chi)$
  1. Compute $s_{\chi\|0} \| r_{\chi\|0} \| s_{\chi\|1} \| r_{\chi\|1} \leftarrow G(s_\chi)$.
  2. For $b \in \{0,1\}$, run $ct_{\chi\|b} \leftarrow \widehat{\text{Enc}}(\widehat{pk}_{d+1}, \text{info}_{\chi\|b}; r_{\chi\|b})$. where,

$$\text{info}_{\chi\|b} = (\text{normal}, C, \chi\|b, \{k_{i,j}\}_{i \in [d+1,D), j \in [1,B]}, s_\chi\|b),$$

  C is the circuit to be obfuscated. Output

$$(H(k_{d,1}, \chi)\| \cdots \|H(k_{d,B}, \chi)) \oplus (ct_{\chi\|0} \| ct_{\chi\|1}).$$

– For $d = D$, $f_D(\text{normal}, C, x, s_x)$, output $C(x)$.

$$\widehat{C}^{\mathcal{O}}[ct_\varepsilon, \{sk_d\}_{d\in[0,D]}, \{h_{i,j}\}_{i\in[0,D),j\in[0,B]}](x)$$

Hardwired. ct_ε, initial ciphertext.
sk_d, secret key.
$h_{i,j}$, handles generated by Pr$\mathcal{O}$M.

Input. $x \in \{0,1\}^D$, input circuit.

Output. Compute as follows.

For $d = 0, \ldots, D-1$**:**

$\chi_d \leftarrow x_{\leq d}$

$\nu_{\chi_d} \leftarrow \overline{\text{Rec}}(\rho_{\chi_d}, ct_{\chi_d})$, $\rho_{\chi_d} \leftarrow \overline{\text{PDec}}(sk_d, ct_{\chi_d})$

$\text{otp}_{\chi_d} \leftarrow \mathcal{O}(\text{hEval}, h_{d,1}, \chi_d \| 0^{D-d}) \| \cdots \| \mathcal{O}(\text{hEval}, h_{d,B}, \chi_d \| 0^{D-d})$

$ct_{\chi_d\|0} \| ct_{\chi_d\|1} \leftarrow \nu_{\chi_d} \oplus \text{otp}_{\chi_d}$

Output $\text{Dec}(sk_D, ct_x)$

Fig. 1. Obfuscated Circuit $(\widehat{C}^{\mathcal{O}}) \rightarrow \widehat{C}^{\bullet}[ct_x, \{sk_d\}_{d\in[0,D]}, \{h_{i,j}\}_{i\in[0,D),j\in[0,B]}]$

Correctness Analysis. According to the obfuscation form $\widehat{C}^{\mathcal{O}}$ in Figure 3 and the tree structure in Figure 2.

$$\begin{aligned}&H(k_{d,1}, \chi_d \| 0^{D-d}) \| \cdots \| H(k_{d,B}, \chi_d \| 0^{D-d})\\ =\ &\mathcal{O}(\text{hEval}, h_{d,1}, \chi_d \| 0^{D-d}) \| \cdots \| \mathcal{O}(\text{hEval}, h_{d,B}, \chi_d \| 0^{D-d}).\end{aligned}$$

6.2 Security Analysis

Lemma 11. *Assuming H is a pseudo-random function, G_{sr}, G_v are pseudo-random generators, and* (Gen, Enc, Enc) *is adaptively secure, with appropriate parameters L and B, then Construction 1 in [JLLW23] is an ideal obfuscation under* Pr$\mathcal{O}$M.

Theorem 7. *Assuming H is a pseudo-random function, G_{sr}, G_v are pseudo-random generators, algorithm 5 is an ideal obfuscation under* Pr$\mathcal{O}$M.

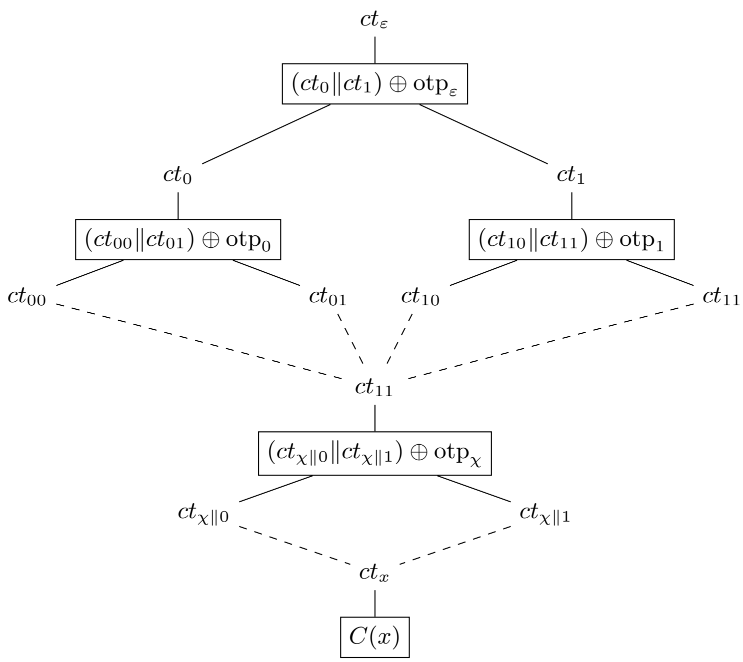

Fig. 2. The binary tree of ciphertexts [JLLW23] in Scheme 5

$$\mathrm{Expand}_{d,\mathrm{hyb}}[pk_{d+1}](\chi, \mathrm{info}_\chi)$$

Hardwired. pk_{d+1}, public key at level $(d+1)$.

Input. $x \in \{0,1\}^d$, input appropriate circuit;

$\mathrm{info}_\chi = (C, \{k_{i,j}\}_{i\in(d,D),j\in[1,B]}, s_\chi, \beta, \{\sigma_{\chi,j}\}_{j\in[0.\beta)}, w_\chi, \{k_{d,j}\}_{j\in(\sigma,B]})$:

C, circuit to be obfuscated.

$k_{i,j}$, keys of H at levels $(d+1, \ldots, D-1)$.

s_χ, seed of pseudo-random generator G_{sr}, related to χ.

β, mixing index.

$\sigma_{\chi,j}$, seed of pseudo-random generator G_v, related to χ.

w_χ, decryption result of the software module.

$k_{d,j}$, keys of H at level $(d+1)$.

Output. Calculated as follows.

$s_{\chi\|0} \| r_{\chi\|0} \| s_{\chi\|1} \| r_{\chi\|1} \leftarrow G_{sr}(s_\chi)$

For $\eta = 0, 1$**:**

$\mathrm{flag}_{\chi\|\eta} \leftarrow \mathrm{normal}$

$\mathrm{info}_{\chi\|\eta} \leftarrow (C, \{k_{i,j}\}_{i\in[d+1,D),j\in[1,B]}, s_{\chi\|\eta})$

$ct_\varepsilon \leftarrow \mathrm{Enc}(pk_{d+1}, \mathrm{flag}_{\chi\|\eta}, \chi\|\eta, \mathrm{info}_{\chi\|\eta})$

Output $\nu_\chi \leftarrow G_\nu(\sigma_\chi, 1) \| \cdots \| G_\nu(\sigma_\chi, \beta-1) \| w_\chi$

$\|([ct_{\chi\|0} \| ct_{\chi\|1}]_{\beta+1} \oplus H(k_{d,\beta+1}, \chi\|0^{D-d})) \| \cdots$

$\|([ct_{\chi\|0} \| ct_{\chi\|1}]_B \oplus H(k_{d,B}, \chi\|0^{D-d}))$

Fig. 3. Obfuscation circuit $(\widehat{C}^{\mathcal{O}}) \to \widehat{C}^{\bullet}[ct_x, \{sk_d\}_{d\in[0,D]}, \{h_{i,j}\}_{i\in[0,D),j\in[0,B]}]$

$\text{Expand}_d[pk_{d+1}](\text{flag}_\chi, \chi, \text{info}_\chi)$ – For Level $d \in [0, D)$

Hardwired. pk_{d+1}, $(d+1)$ level public key.

Input. $\text{flag}_\chi \in \{\text{normal}, \text{hyb}, \text{sim}\}$, flag matching with χ.

$\chi \in \{0,1\}^d$, input appropriate prefix for the circuit.

info_χ, information matching with χ, changes with flag_χ.

Output.

$$\begin{cases} \text{Expand}_{d,\text{normal}}[pk_{d+1}](\chi, \text{info}_\chi), & \text{if } \text{flag}_\chi = \text{normal}; \\ \left.\begin{array}{ll} \text{Expand}_{d,\text{hyb}}[pk_{d+1}](\chi, \text{info}_\chi), & \text{if } \text{flag}_\chi = \text{hyb}; \\ \text{Expand}_{d,\text{hyb}}(\chi, \text{sim}_\chi), & \text{if } \text{flag}_\chi = \text{sim}. \end{array}\right\} \text{Fig.7} \end{cases}$$

$\text{Eval}(\text{flag}_\chi, \chi, \text{info}_\chi)$ – For Level D

Hardwired. pk_{d+1}, $(d+1)$ level public key.

Input. $\text{flag}_\chi \in \{\text{normal}, \text{sim}\}$, flag matching with χ.

$\chi \in \{0,1\}^D$, input circuit.

info_χ, information matching with χ, changes with flag_χ.

Output.

$$\begin{cases} \text{Eval}_{d,\text{normal}}(\chi, \text{info}_\chi), & \text{if } \text{flag}_\chi = \text{normal}; \\ \text{Eval}_{d,\text{sim}}(\chi, \text{info}_\chi), & \text{if } \text{flag}_\chi = \text{sim}. \end{cases}$$

$\text{Expand}_{d,\text{normal}}[pk_{d+1}](\chi, \text{info}_\chi)$

Hardwired. pk_{d+1}, $(d+1)$ level public key.

Input. $\chi \in \{0,1\}^d$, input appropriate prefix for the circuit.

$\text{info}_\chi = (C, \{k_{i,j}\}_{i\in[0,D), j\in[0,B]}, s_\chi)$:

C, circuit to be obfuscated.

$k_{i,j}$, keys for hash functions $d, \ldots, D-1H$.

s_χ, seed for pseudo-random generator G_{sr} corresponding to χ.

Output. Perform the following calculations.

$s_{\chi\|0}\|r_{\chi\|0}\|s_{\chi\|1}\|r_{\chi\|1} \leftarrow G_{sr}(s_\chi)$.

for $\eta = 0, 1$:

$\text{flag}_{\chi\|\eta} \leftarrow \text{normal}$

$\text{info}_{\chi\|\eta} \leftarrow (C, \{k_{i,j}\}_{i\in[d+1,D), j\in[1,B]}, s_{\chi\|\eta})$.

$ct_{\chi\|\eta} \leftarrow \text{Enc}(pk_{d+1}, \text{flag}_{\chi\|\eta}, \chi\|\eta, \text{info}_{\chi\|\eta})$

$\text{otp}_\chi \leftarrow H(k_{d,1}, \chi\|0^{D-d}\| \cdots \|H(k_{d,B}, \chi\|0^{D-d})$

Output $\nu_\chi \leftarrow (ct_{\chi\|0}\|ct_{\chi\|1}) \oplus \text{otp}_\chi$

$\text{Eval}_{\text{normal}}(\chi, \text{info}_\chi)$

Input. $\chi \in \{0,1\}^D$, input circuit.

$\text{info}_\chi = (C, s_\chi)$:

C, circuit to be obfuscated.

s_χ, unused seed.

Output. $C(\chi)$, compute the evaluation of a generalized circuit (C, χ).

References

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Advances in Cryptology – CRYPTO 2015*, pages 308–326. Springer Berlin Heidelberg, 2015.

[AJLA+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold fhe. In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, page 483 - 501. Springer-Verlag, 2012.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited. In *Advances in Cryptology – CRYPTO 2013*, pages 57–74. Springer Berlin Heidelberg, 2013.

[BDGM19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In *Theory of Cryptography: 17th International Conference*, page 407 - 437. Springer-Verlag, 2019.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. In *Advances in Cryptology – EUROCRYPT 2020*, pages 79–109. Springer International Publishing, 2020.

[BDGM22] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and Pairings Are Not Necessary for IO: Circular-Secure LWE Suffices. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229, pages 1–20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology — CRYPTO 2001*, pages 1–18. Springer Berlin Heidelberg, 2001.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of ggh15: Provable security against zeroizing attacks. In *Theory of Cryptography*, pages 544–574, Cham, 2018. Springer International Publishing.

[BV18] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *Journal of the ACM*, 65(6):1 - 37, 2018.

[BZ17] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *Algorithmica*, 79:1233–1285, 2017.

[Ceg12] Andrzej Cegielski. *Iterative Methods for Fixed Point Problems in Hilbert Spaces*. 2012.

[Che24] Yilei Chen. Quantum algorithms for lattice problems. Cryptology ePrint Archive, Paper 2024/555, 2024. https://eprint.iacr.org/2024/555.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology – CRYPTO 2013*, pages 476–493. Springer Berlin Heidelberg, 2013.

[DS16] Léo Ducas and Damien Stehlé. Sanitization of fhe ciphertexts. In *Advances in Cryptology – EUROCRYPT 2016*, pages 294–310. Springer Berlin Heidelberg, 2016.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford, CA, USA, 2009.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, page 169 – 178. Association for Computing Machinery, 2009.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology – EUROCRYPT 2013*, pages 1–17. Springer Berlin Heidelberg, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 40–49, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography*, pages 498–527. Springer Berlin Heidelberg, 2015.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. In *Theory of Cryptography*, pages 74–94. Springer Berlin Heidelberg, 2014.

[GSM18] Fuchun Guo, Willy Susilo, and Yi Mu. *Introduction to Security Reduction*. 2018.

[Had00] Satoshi Hada. Zero-knowledge and code obfuscation. In *Advances in Cryptology — ASIACRYPT 2000*, pages 443–457. Springer Berlin Heidelberg, 2000.

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of ggh map. In *Advances in Cryptology – EUROCRYPT 2016*, pages 537–565. Springer Berlin Heidelberg, 2016.

[JLLW23] Aayush Jain, Huijia Lin, Ji Luo, and Daniel Wichs. The pseudorandom oracle model and ideal obfuscation. In *Advances in Cryptology – CRYPTO 2023*, pages 233–262, Cham, 2023. Springer Nature Switzerland.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, page 60 – 73. Association for Computing Machinery, 2021.

[Lin17] Huijia Lin. Indistinguishability obfuscation from sxdh on 5-linear maps and locality-5 prgs. In *Advances in Cryptology - CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 599–629. Springer, 2017.

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In *Advances in Cryptology – CRYPTO 2017*, pages 630–660. Springer International Publishing, 2017.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. In *Advances in Cryptology – CRYPTO 2016*, pages 629–658. Springer Berlin Heidelberg, 2016.

[Nor66] Levinson Norman. On the elementary proof of the prime number theorem. *Proceedings of the Edinburgh Mathematical Society*, 15(2):141 – 146, 1966.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *Journal of ACM*, 51:899–942, 2004.

[SW21] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *SIAM Journal on Computing*, 50(3):857 – 908, 2021.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-io from evasive lwe. In *Advances in Cryptology – ASIACRYPT 2022*, pages 195–221, Cham, 2022. Springer Nature Switzerland.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and cp-abe from evasive lattice assumptions. In *Advances in Cryptology – EUROCRYPT 2022*, pages 217–241, Cham, 2022. Springer International Publishing.

[XXZ12] Xiang Xie, Rui Xue, and Rui Zhang. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In *Proceedings of the 8th International Conference on Security and Cryptography for Networks*, page 1 – 18. Springer-Verlag, 2012.

[Yue20] Steven Yue. Introduction to io 01: What is indistinguishability obfuscation (io)?, 2020.

[YZ21] Yu Yu and Jiang Zhang. Smoothing out binary linear codes and worst-case sub-exponential hardness for lpn. In *Advances in Cryptology – CRYPTO 2021*, pages 473–501, Cham, 2021. Springer International Publishing.