Pseudo Flawed-Smudging Generators and Their Application to Indistinguishability Obfuscation

Huijia Lin Christian Matt

University of California, Santa Barbara
{rachel.lin, cmatt}@cs.ucsb.edu

Abstract

We construct indistinguishability obfuscation from subexponentially secure Learning With Errors (LWE), bilinear maps, a constant-locality Pseudo Random Generator (PRG), and a new tool called *Pseudo Flawed-smudging Generator* (PFG). A PFG is an expanding function whose outputs Y satisfy a weak form of pseudo randomness. Roughly speaking, for some polynomial bound B, and any B-bounded noise vector distribution χ, it guarantees that for $e \leftarrow \chi$, the distribution of $(e,\ Y + e)$ is indistinguishable from $(e', Y + e)$, where e' is a fresh random sample from χ conditioned on agreeing with e at *a few*, $o(\lambda)$, locations. In other words, Y "hides" e at *all but a few* locations. Our construction of indistinguishability obfuscation requires a PFG that is computable by a *degree 2* polynomial over the integers and has polynomially bounded outputs. We finally propose a candidate of such PFGs and formalize an assumption under which it satisfies the requirements of our construction.

Contents

1 Introduction **3**

2 Preliminaries **7**
- 2.1 Notation and Basic Definitions . . . 7
- 2.2 μ-Indistinguishability . . . 8
- 2.3 Learning with Errors . . . 8
- 2.4 Pseudorandom Generators and Pseudorandom Functions . . . 9
- 2.5 Indistinguishability Obfuscation . . . 9
- 2.6 Randomized Encodings . . . 10
- 2.7 Functional Encryption . . . 10
  - 2.7.1 Public-Key Functional Encryption . . . 11
  - 2.7.2 FE for $\mathsf{P/poly}$, NC^1, and Compactness . . . 12
- 2.8 (Fully) Homomorphic Encryption . . . 13
  - 2.8.1 Threshold Multi-Key FHE . . . 14

3 Definition of Pseudo Flawed-Smudging Generators **15**

4 Functional Encryption for Constant Degree Polynomials **16**
- 4.1 Base Case: $\mathsf{2FE}^{N,S}$ for Quadratic Polynomials . . . 17
- 4.2 Recursive Construction: $\mathsf{d{+}1FE}^{N,S}$ for degree $d+1$ Polynomials . . . 19
  - 4.2.1 Proof of Special Purpose Simulation Security of $\mathsf{d{+}1FE}$ . . . 24
  - 4.2.2 Property of DFE . . . 29

5 Functional Encryption for NC_1 **31**
- 5.1 Bit-fixing Homomorphic Sharing . . . 31
  - 5.1.1 Construction from Threshold Multi-Key FHE . . . 33
- 5.2 Construction of FE for NC_1 . . . 34

6 Construction of Candidate PFG **42**
- 6.1 Properties of (Flawed-)Smudging Distributions . . . 42
  - 6.1.1 Preservation Under Addition of Independent Value . . . 42
  - 6.1.2 Mixtures of (Flawed-)Smudging Distributions . . . 42
  - 6.1.3 Smudging and Independence Implies Flawed-Smudging . . . 43
- 6.2 Candidate Construction . . . 48
  - 6.2.1 Definitions . . . 48
  - 6.2.2 Flawed-Smudging Property of Candidate . . . 49
  - 6.2.3 Choice of Parameters . . . 54

Acknowledgments **54**

1 Introduction

Indistinguishability obfuscation (IO), first defined in the seminal work of Barak et. al. [BGI$^+$01], aims to obfuscate functionally equivalent programs into indistinguishable ones while preserving functionality. IO is an extraordinarily powerful object that has been shown to enable a large set of new cryptographic applications.

The first-generation IO constructions [GGH$^+$13, BR14, BGK$^+$14, PST14, AGIS14, GLSW15, Zim15, AB15, GMM$^+$16, DGG$^+$16] rely on polynomial-degree *multilinear maps* or *graded encodings*. An L-linear map [BS03] essentially allows to evaluate degree-L polynomials on secret encoded values, and to test whether the output of such polynomials is zero or not. While bilinear maps (i.e., $L = 2$) can be efficiently instantiated from elliptic curves, instantiation of L-linear maps for $L \geq 3$ has remained elusive — so far, vulnerabilities [CHL$^+$15, CGH$^+$15, MSZ16, CGH17, ADGM17] were demonstrated against all known candidates [CLT13, LSS14, GGH15, CLT15]. Of course, this does not mean that the resulting IO constructions are insecure; in particular, the construction of [GMM$^+$16] is formally shown to withstand all existing attacks.

A line of recent works [Lin16, LV16, Lin17, AS17] aimed at finding the minimal degree of multilinear maps sufficient for constructing IO, and has successfully reduced the required degree to $L = 3$. A key ingredient in these second-generation constructions is PRGs with simple structure, in particular, *small locality*. It was shown that the degree of multilinear maps needed matches exactly the locality of the PRG [Lin16, AS17], or even a relaxed notion of *block locality* [LT17]. These constructions essentially use degree-L multilinear maps to evaluate the simple PRG with (block)-locality L, and then bootstrap from there to hide arbitrary complex computation. Unfortunately, the locality of a PRG cannot be smaller than 5 [CM01, MST03], and recent attacks [LV17, BBKK18] showed that block-locality cannot be smaller than 3. This raises the following natural question:

> *Are there simple PRGs, with potentially weak security guarantees, that can be evaluated using bilinear maps and are useful for constructing IO?*

Flawed-Smudging Generators To this end, we propose *Pseudo Flawed-smudging Generators* (PFGs). They are polynomially stretching functions from n input elements to $m = n^{1+\alpha}$ output elements (where n, m are parameterized by the security parameter λ), satisfying a weak form of pseudo randomness called *pseudo flawed-smudging*. For IO construction, we need simple PFGs that are computable by degree 2 polynomials over $\mathbb{Z}$ (with no restriction on locality) and have polynomially bounded outputs (i.e., every output element is an integer of polynomial magnitude). As such, they can be computed in the exponent of bilinear pairing groups, and their outputs can be extracted via brute force discrete logarithm.

The *pseudo flawed-smudging* property ensures that the output of a PFG evaluated on an input from a specific distribution is able to "smudge" (or flood) a small noise vector, at *all but a few*, $o(\lambda)$, locations. More precisely, we require the output of a PFG to be indistinguishable to a, so-called, *flawed-smudging distribution* $\mathbf{Y} \leftarrow \mathcal{Y}$. It ensures that for some polynomial bound B, and every B-bounded noise vector distribution $\mathbf{e} \leftarrow \chi$, $\mathbf{Y} + \mathbf{e}$ "hides" the noise vector $\mathbf{e}$ at *all but a few* locations in the following sense. There is a random variable I correlated with $\mathbf{e}, \mathbf{Y}$, representing a small ($|I| = o(\lambda)$) subset of locations to be fixed, so that, the joint distribution of $(I, \mathbf{e}, \mathbf{Y} + \mathbf{e})$ is close to that of $(I, \mathbf{e}', \mathbf{Y} + \mathbf{e})$,

$$\{\ I,\ \mathbf{e},\ \mathbf{Y} + \mathbf{e}\ \} \approx \{\ I,\ \mathbf{e}',\ \mathbf{Y} + \mathbf{e}\ \}, \text{ where } \mathbf{e}' \leftarrow \chi|_{\mathbf{e}_I, I}\ ,$$

where $\mathbf{e}'$ is a fresh new sample from χ conditioned on agreeing with $\mathbf{e}$ at locations in I (i.e., $e'_i = e_i$ for all $i \in I$). In short, given $\mathbf{Y} + \mathbf{e}$, the noise vector remains random, up to a few locations being fixed! (The formal definition of flawed-smudging distribution is more complex, as discussed in our techniques and in Section 3.)

In the literature, noise smudging (or noise flooding) is a commonly used technique for hiding small noises in LWE samples, which is also our purpose. However, the smudging distributions used in the literature usually have *super-polynomially* large output elements (for instance, a discrete Gaussian with super-polynomial standard deviation, or a uniform distribution over a consecutive super-polynomial-sized support). A sample $\mathbf{Y}$ from such distributions can hide a small noise vector $\mathbf{e}$ *entirely* at all locations (with overwhelming probability), in the sense that $(\mathbf{e}, \mathbf{Y} + \mathbf{e}) \approx (\mathbf{e}', \mathbf{Y} + \mathbf{e})$ for a completely independent $\mathbf{e}' \leftarrow \chi$. In fact, to hide the noise vectors $\mathbf{e}$ entirely, it is *necessary* that $\mathbf{Y}$ is super-polynomially large. This highlights the key rationale behind the definition of flawed-smudging distributions — when the smudging distribution is polynomially bounded, it inevitably leads to "leakage" of the noise vector $\mathbf{e}$ at some locations with high probability (e.g., discrete Gaussian distributions with polynomial standard deviation, and uniform distributions with consecutive polynomial-sized supports, both behave as such).

Our Results Leveraging the simple structure of PFGs, we construct functional encryption schemes for computing NC_1 circuits that have sublinearly compact ciphertexts (ciphertext size grows polynomially in the security parameter and input-length, and sublinearly in the size of the computation), and satisfy standard 1-key fully-selective indistinguishability security.

Theorem 1.1. *There is a construction of public key functional encryption schemes for computing polynomial-sized circuits in NC_1, satisfying sublinearly compactness and 1-key fully selective indistinguishability security, from LWE, bilinear maps, a constant-locality PRG (with mild structural properties), and a degree-2 pseudo flawed-smudging generator over $\mathbb{Z}$ with polynomially bounded outputs.*

Previous works [AJ15, BV15, LPST16b, LPST16a, BNPW16, KNT18] showed that such functional encryption schemes with subexponential security imply indistinguishability obfuscation. Hence, we get the following corollary:

Corollary 1.2. *There is a construction of indistinguishability obfuscation for polynomial-sized circuits from LWE, bilinear maps, a constant-locality PRG (with mild structural properties), and a degree-2 pseudo flawed-smudging generator over $\mathbb{Z}$ with polynomially bounded outputs, all with sub-exponential security.*

We further analyze properties of flawed-smudging distributions. First, we show that any product distribution $\mathcal{D}_1 \times \cdots \times \mathcal{D}_m$, satisfying that every distribution $\mathcal{D}_i$ has sufficiently (but still polynomially) small statistical distance between $\mathcal{D}_i$ and $\mathcal{D}_i + e$ for every B-bounded constant e, is a flawed-smudging distribution. Second, the flawed-smudging property is preserved under addition with an independent distribution, and under convex combinations. More precisely, the distribution obtained by adding a sample from a flawed-smudging distribution $\mathcal{Y}$ and a sample from an arbitrary independent distribution $\mathcal{T}$ is flawed-smudging, and the convex combination of any number of flawed-smudging distributions is also flawed-smudging.

Based on above properties, we propose a candidate PFG, which is a variant of random Multivariate Quadratic (MQ) polynomials over $\mathbb{Z}$. The security of random MQ polynomials have been studied in finite rings such as $\mathbb{Z}_p$ [BGP06, HLY12]. We emphasize that MQ polynomials over the integers behave differently. In particular, for a random MQ polynomial over $\mathbb{Z}$, it is not even clear whether the marginal distribution $\mathcal{D}_i$ of an individual output element i can hide a small (scalar) noise with good probability (for instance, the range of an output element may be non-consecutive, and hence $SD(\mathcal{D}_i, \mathcal{D}_i + 1)$ is already large). Our candidate PFG samples MQ polynomials from a specific distribution which ensures that the marginal distribution of every output element indeed satisfies that $SD(\mathcal{D}_i, \mathcal{D}_i + e)$ is sufficiently small for all small e. Finally, we propose to apply the aforementioned operations that preserve the flawed-smudging property to make our candidate harder to attack, concretely, by adding it together with an independent instance of random MQ polynomials.

Our Techniques Towards constructing functional encryption schemes for NC_1, we follow the same two-step approach as previous works [Lin16, LV16, Lin17, AS17]: First construct functional encryption schemes for computing constant-degree polynomials, or constant-degree FE for short; then bootstrap constant-degree FE to FE for computing NC^1 circuits. The key difference of our technique is that we manage to construct constant-degree FE from LWE, bilinear maps, and a flawed-smudging generator, whereas previous constructions rely on constant-degree multilinear maps. However, relying on only bilinear maps comes at a price — our constant-degree FE is *weak and leaky*, and reveals the encrypted input vector at a few locations. As a result, the bootstrapping step from constant-degree FE to FE for NC_1 must be *robust* to such leakage.

Constant-Degree FE via Homomorphic Encryption. In order to avoid the use of multilinear maps with degree larger than 2, we need to use other tools to compute a constant-degree polynomial f over a secret input $\mathbf{x}$, and then extract the output $\mathbf{y} = f(\mathbf{x})$ in the clear. The natural approach is using Homomorphic Encryption (HE) schemes, which has been explored in [GKP$^+$13, Agr18a] as discussed in related work. At a high-level, our constant-degree FE scheme encrypts an input $\mathbf{x}$ using a HE scheme and a secret vector $\mathbf{s}$; one can then homomorphically evaluate any function f on $\mathbf{x}$ and obtain a ciphertext ct_f encrypting $\mathbf{y}$. The difficult part is how to decrypt a ciphertext ct_f associated with a "legitimate" function f (ones for which secret keys are generated) to recover $f(\mathbf{x})$ in the clear. We do so using a FE scheme for computing only *quadratic* polynomials (or quadratic FE for short). There are constructions of quadratic FE schemes from bilinear maps [Lin16, BCFG17, AS17], which however have the limitation that outputs are computed in the exponent of the target group of the bilinear map, and can only be recovered in the clear if they reside in a polynomial-sized range.

To decrypt HE ciphertext ct_f using quadratic FE, we observe that the decryption of certain HE schemes, such as [BV11, BGV12], is simple — in fact, a linear operation, such as $\langle \mathsf{ct}_f, \mathbf{s} \rangle$, already yields an approximate output, such as $\mathbf{y} + 2\mathbf{e}$, perturbed by a small noise vector $\mathbf{e}$. However, the noise $\mathbf{e}$ is *sensitive*, revealing information about the input $\mathbf{x}$, the HE secret $\mathbf{s}$, and the noises used for generating the original ciphertext encrypting $\mathbf{x}$. Therefore, we need a way to hide $\mathbf{e}$ using just degree 2 computation. This is where noise smudging comes in: We instead compute the approximate output $\mathbf{y} + 2\mathbf{e} + 2\mathbf{Y}$ further shifted by a large noise $\mathbf{Y}$ in order to hide $\mathbf{e}$. At a first glance, it seems that any degree 2 PRG (such as ones based on the MQ assumptions over $\mathbb{Z}_p$) can be used to generate $\mathbf{Y}$. However, a more careful examination reveals a dilemma: Current quadratic FE schemes can only evaluate quadratic polynomials whose outputs are polynomially bounded, and as we discussed earlier, polynomially-bounded $\mathbf{Y}$ cannot hide $\mathbf{e}$ entirely.

Weak and Leaky Constant-Degree FE. Therefore, we use our Pseudo-Flawed-smudging Generator, which has degree 2, polynomially-bounded outputs, and ensures that $\mathbf{e} + \mathbf{Y}$ hides $\mathbf{e}$ at *all but* *a few* locations. However, revealing $\mathbf{e}$ even at *a few* locations violates the standard security requirement of FE. Therefore, we turn to achieve what is the best possible. By using a HE scheme that is *robust to leakage*, we construct constant-degree FE with (1-key) *weak and leaky* simulation security. Roughly speaking, it guarantees that a tuple $(\mathsf{mpk}, \mathsf{sk}_f, \mathsf{ct}_x)$ consisting of an honestly generated master public key mpk, a secret key sk_f for a distributional function $f \leftarrow \mathcal{FN}$, and a ciphertext ct_x for a distributional input $x \leftarrow \mathcal{X}$, can be simulated by a simulator Sim using the output $y = f(x)$ of a randomly sampled x conditioned on its value being fixed at a few locations. More precisely, there is a distribution Fix over the fixed locations K and values x^*, such that, $|K| = o(\lambda)$, and

$$\{\ x,\ \mathsf{mpk}, \mathsf{sk}_f, \mathsf{ct}_x\ \} \ \approx\ \{\ x,\ \mathsf{Sim}\left((x^*, K),\ f,\ y = f(x)\right)\}\ ,$$
$$\text{where } (x^*, K) \leftarrow \mathsf{Fix},\ \text{ and } x \leftarrow \mathcal{X}|_{x^*,K}\ .$$

In other words, given $\mathsf{mpk}, \mathsf{sk}_f, \mathsf{ct}_x$, the encrypted input x appears random up to a few locations being fixed, and the output being y.

HE schemes robust to leakage can be instantiated using the [BV11, BGV12] schemes based on LWE, thanks to the robustness of LWE itself. When the LWE secret $\mathbf{s}$ comes from a small domain (e.g., $\mathbf{s}$ is binary), the hardness of LWE holds as long as $\mathbf{s}$ has sufficient entropy and does not necessarily need to be uniformly random [GKPV10, AKPW13]. Furthermore, for the construction of weak and leaky constant-degree FE to go through, we need a slightly stronger version of the flawed-smudging property: Consider a B-bounded noise vector distribution $\chi = e(\mathcal{R})$ where the noise $\mathbf{e}$ is a function over other distributional secret $\mathbf{w} \leftarrow \mathcal{R}$; there is again a correlated random variable I, such that,

$$\{\ I,\ \mathbf{w},\ \mathbf{Y} + e(\mathbf{w})\ \} \approx \{\ I,\ \mathbf{w}',\ \mathbf{Y} + e(\mathbf{w})\ \},\ \text{ where } \mathbf{w}' \leftarrow \chi|_{\mathbf{w}_I, I}\ .$$

This means given $\mathbf{Y} + E(\mathbf{w})$, only a few locations of the secret $\mathbf{w}$ get fixed and leaked. In our construction of constant-degree FE, we use this guarantee to bound what information of the HE input $\mathbf{x}$ and secret $\mathbf{s}$ is fixed and leaked through leakage of the noise $\mathbf{e}$ in the ciphertext obtained via homomorphic evaluation.

Bootstrapping from Weak and Leaky Constant-Degree FE. We next present a new bootstrapping technique to FE for NC_1 from weak and leaky constant-degree FE. Our bootstrapping follows the same paradigm as previous works [Lin16, LV16, Lin17, AS17, LV17] — it uses a randomized encoding [IK02, AIK04] to transform a NC_1 computation $g(v)$ into a simple constant-degree polynomial $\hat{g}(v; r)$, and uses a constant locality PRG to supply pseudorandom coins $r = \mathsf{PRG}(\text{seed})$ needed for the randomized encoding. The fact that the underlying constant-degree FE is weak and leaky means both the input v, as well as the PRG seed maybe fixed and leaked at a few locations. To deal with this, we introduce a new primitive called Bit-Fixing Homomorphic Sharing in order to make the original computation g robust.

Our bit-fixing homomorphic sharing resembles the recent new concept of Homomorphic Secret Sharing (HSS) [BGI15] in syntax, but differs in security and efficiency requirements. It enables compiling a single computation $g(v)$ into a collection of computations $o_1 = h_1(x_1), \cdots, o_\lambda = h_\lambda(x_\lambda)$ that operates on a secret sharing $x_1, \cdots, x_q$ of the original input v, and from the collection of outputs $o_1, \cdots, o_q$, the original output $g(v)$ can be reconstructed. Security ensures that the original input v remains hidden, given all output shares $o_1 \cdots o_q$ and a small subset of input shares (HSS only guarantees that the input remains hidden given a subset of input shares, without the output shares). Moreover, the security is robust to a few bits in the input shares being fixed.

We give a construction of bit-fixing homomorphic sharing from multi-key FHE with threshold decryption as constructed in [MW16].

Next, we use the weak and leaky constant-degree FE to compute the randomized encoding of the compiled computations $\left\{\hat{h}_i(x_i\ ;\ r_i)\right\}$. Through careful analysis, we show that the weak security of constant-degree FE only leads to a small subset of the computation $o_i = h_i(x_i)$ being "corrupted", meaning the input share x_i is revealed or some bits of x_i are fixed. It then follows from the security of bit-fixing homomorphic sharing that the original input v remains hidden.

Related Works As mentioned above, the approach of using a homomorphic encryption scheme to construct functional encryption has already been explored in the works of [GKP+13, Agr18a]. The challenge lies in designing ways to decrypt ciphertexts ct_f obtained from homomorphic evaluations. The work of [GKP+13] achieves this using attribute-based encryption and garbled circuits (with the HE secret $\mathbf{s}$ hardcoded), which, however, results in non-compact ciphertexts whose sizes scale with the complexity of the computation. Such non-compact FE schemes are insufficient for the construction of IO. To make the ciphertext size compact, Agrawal proposed the approach [Agr18b] of using a noise generator to generate a large noise $\mathbf{Y}$ to flood the noise $\mathbf{e}$ in ct_f, so that it is safe to reveal $y + \mathbf{e} + \mathbf{Y}$. In this work, we follow her approach and explore what happens when $\mathbf{Y}$ is polynomially bounded and $\mathbf{e}$ maybe leaked.

Outline of the paper We review some notation and standard cryptographic notions that we use in the paper in Section 2. In Section 3, we define pseudo flawed-smudging generators (PFGs). Section 4 describes the construction of a functional encryption scheme for constant degree polynomials, which makes use of a PFG. In Section 5, we construct a functional encryption scheme for NC_1 using our FE scheme from Section 4 and additional tools, including bit-fixing homomorphic sharing, which we introduce in Section 5.1. Finally, Section 6 discusses a candidate PFG and proves its security under an assumption we formalize there.

2 Preliminaries

2.1 Notation and Basic Definitions

We denote by $\mathbb{Z}$ the set of integers and by $\mathbb{N}$ the set of nonnegative integers. For $n \in \mathbb{N}$, $[n] := \{1, \ldots, n\}$. For a distribution D, $x \xleftarrow{\$} D$ denotes that x is sampled according to D, for a probabilistic algorithm A, $y \xleftarrow{\$} A(x)$ denotes running A on input x and assigning the output to y, and for a finite set S, $x \xleftarrow{\$} S$ denotes assigning a uniformly random value from S to x.

Definition 2.1 (Statistical Distance)**.** Let X and X' be random variables over a discrete set $\mathcal{X}$. The *statistical distance* between X and X' is defined as

$$\delta(X, X') := \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr[X = x] - \Pr[X' = x]|.$$

The *min-entropy* of a random variable X is defined as

$$H_\infty(X) := -\log\Big(\max_x \Pr[X = x]\Big).$$

We further define the *conditional min-entropy* of X given Z following Dodis et al. [DORS08] as

$$H_\infty(X \mid Z) := -\log\Big(\mathbb{E}_{z \xleftarrow{\$} Z}\Big[\max_x \Pr[X = x \mid Z = z]\Big]\Big).$$

We denote by PPT probabilistic polynomial time Turing machines. The term *negligible* is used for denoting functions that are (asymptotically) smaller than any inverse polynomial. More precisely, a function ν from $\mathbb{N}$ to reals is called *negligible* if for every constant $c > 0$ and all sufficiently large n, $\nu(n) < n^{-c}$.

2.2 μ-Indistinguishability

Definition 2.2 (μ-indistinguishability)**.** Let $\mu \colon \mathbb{N} \to [0,1]$ be a function. A pair of distribution ensembles $\{X_\lambda\}_{\lambda\in\mathbb{N}}$, $\{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are μ-indistinguishable if for every family of polynomial-sized distinguishers $\{D_\lambda\}_{\lambda\in\mathbb{N}}$, and every sufficiently large security parameter $\lambda \in \mathbb{N}$,

$$\Big|\Pr\Big[x \xleftarrow{\$} X_\lambda : D\big(1^\lambda, x, z\big) = 1\Big] - \Pr\Big[y \xleftarrow{\$} Y_\lambda : D\big(1^\lambda, y, z\big) = 1\Big]\Big| \le O(\mu(\lambda)).$$

2.3 Learning with Errors

We next state the decisional learning with errors (LWE) assumption, which was introduced by Regev [Reg05].

Definition 2.3. Let $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be integers and let $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}_{q(\lambda)}$ for $\lambda \in \mathbb{N}$. Then, the $\mathsf{LWE}_{n,m,q,\chi}$ assumption with μ-indistinguishability is that the following distributions are μ-indistinguishable:

$$\Big\{\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}; \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n; \mathbf{e} \xleftarrow{\$} \chi^m : (\mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e})\Big\}_{\lambda\in\mathbb{N}}$$
$$\Big\{\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}; \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m : (\mathbf{A}, \mathbf{u})\Big\}_{\lambda\in\mathbb{N}}$$

It has been shown that this assumptions holds when χ is a discrete Gaussian distribution if certain worst-case lattice problems are hard [Reg05, Pei09].

We further define LWE with weak and leaky secrets, as introduced in the full version of [AKPW13].

Definition 2.4 (LWE with Weak and Leaky Secrets)**.** Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\gamma = \gamma(\lambda) \in (0, q/2) \cap \mathbb{Z}$ be integers, let $k = k(\lambda)$ be a real, and let $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}_{q(\lambda)}$ for $\lambda \in \mathbb{N}$. Then, the $\mathsf{LWE}_{n,m,q,\chi}^{\mathsf{WL}(\gamma,k)}$ assumption with μ-indistinguishability states that for all efficiently samplable correlated random variables $(\mathbf{s}, \mathrm{aux})$, where the support of $\mathbf{s}$ is $[-\gamma, \gamma]^n \cap \mathbb{Z}^n$ and $H_\infty(\mathbf{s} \mid \mathrm{aux}) \ge k$, the following distributions are μ-indistinguishable

$$(\mathrm{aux}, \mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e}), \quad (\mathrm{aux}, \mathbf{A}, \mathbf{u}),$$

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\mathbf{e} \xleftarrow{\$} \chi^m$, and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$ are sampled independently of $(\mathbf{s}, \mathrm{aux})$.

2.4 Pseudorandom Generators and Pseudorandom Functions

We review the notion of a pseudorandom generator (PRG) family and its locality.

Definition 2.5 (Family of Pseudorandom Generators (PRGs))**.** Let n and m be polynomials. A family of (n, m)-PRGs is an ensemble of distributions $\mathsf{PRG} = \{\mathsf{PRG}_\lambda\}_{\lambda\in\mathbb{N}}$ satisfying the following properties:

Syntax: For every $\lambda \in \mathbb{N}$, every PRG in the support of PRG_λ defines a function $\{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}$, for which we also write PRG.

Efficiency: There is a uniform Turing machine M satisfying that for every $\lambda \in \mathbb{N}$, every PRG in the support of PRG_λ, and for every $x \in \{0,1\}^{n(\lambda)}$, $M(\mathrm{PRG}, x)$ runs in time $\mathrm{poly}(\lambda)$ and we have $M(\mathrm{PRG}, x) = \mathrm{PRG}(x)$.

μ-Indistinguishability: The following ensembles are μ-indistinguishable:

$$\left\{\mathrm{PRG} \xleftarrow{\$} \mathsf{PRG}_\lambda; s \xleftarrow{\$} \{0,1\}^{n(\lambda)} : (\mathrm{PRG}, \mathrm{PRG}(s))\right\}_\lambda,$$
$$\left\{\mathrm{PRG} \xleftarrow{\$} \mathsf{PRG}_\lambda; r \xleftarrow{\$} \{0,1\}^{m(\lambda)} : (\mathrm{PRG}, r)\right\}_\lambda.$$

Definition 2.6 (Locality of PRGs)**.** Let n, m, and ℓ be polynomials. We say a family of (n, m)-PRGs PRG has locality ℓ if for every λ and for every PRG in the support of PRG_λ, every output bit of PRG depends on at most $\ell(\lambda)$ input bits.

We next define pseudorandom function (PRF) families.

Definition 2.7. For $\lambda \in \mathbb{N}$, let $\mathcal{K} = \mathcal{K}(\lambda)$, $X = X(\lambda)$, and $Y = Y(\lambda)$ be finite sets, and let $\mathsf{PRF} = \mathsf{PRF}_\lambda \colon \mathcal{K} \times X \to Y$ be an efficiently computable function. We say $(\mathsf{PRF}_\lambda)_{\lambda\in\mathbb{N}}$ is a pseudorandom function family with μ-indistinguishability if for all PPT algorithms A with access to an oracle, which is either $\mathsf{PRF}(K, \cdot)$ for $K \xleftarrow{\$} \mathcal{K}$ or a truly uniform function $X \to Y$,

$$\left|\Pr\left[K \xleftarrow{\$} \mathcal{K} : A^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1\right] - \Pr\left[U \xleftarrow{\$} (X \to Y) : A^{U(\cdot)}(1^\lambda) = 1\right]\right| \le O(\mu(\lambda)).$$

2.5 Indistinguishability Obfuscation

We recall the notion of indistinguishability obfuscation for a class of circuit defined by [BGI+01].

Definition 2.8 (Indistinguishability Obfuscator ($i\mathcal{O}$) for a circuit class)**.** A uniform PPT machine $i\mathcal{O}$ is an indistinguishability obfuscator for a class of circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$, if the following conditions are satisfied:

Correctness: For all security parameters $\lambda \in \mathbb{N}$, for every $C \in \mathcal{C}_\lambda$, and every input x, we have that

$$\Pr[C' \leftarrow i\mathcal{O}(1^\lambda, C) \ : \ C'(x) = C(x)] = 1$$

where the probability is taken over the coin-tosses of the obfuscator $i\mathcal{O}$.

μ-Indistinguishability: For every ensemble of pairs of circuits $\{C_{0,\lambda}, C_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ satisfying that $C_{b,\lambda} \in \mathcal{C}_\lambda$, $|C_{0,\lambda}| = |C_{1,\lambda}|$, and $C_{0,\lambda}(x) = C_{1,\lambda}(x)$ for every x, the following ensembles of distributions are μ-indistinguishable:

$$\left\{C_{1,\lambda}, C_{2,\lambda}, i\mathcal{O}(1^\lambda, C_{1,\lambda})\right\}_{\lambda \in \mathbb{N}}$$
$$\left\{C_{1,\lambda}, C_{2,\lambda}, i\mathcal{O}(1^\lambda, C_{2,\lambda})\right\}_{\lambda \in \mathbb{N}}$$

Definition 2.9 (IO for $\mathsf{P/poly}$)**.** A uniform PPT machine $i\mathcal{O}_{\mathsf{P/poly}}(\star, \star)$ is an indistinguishability obfuscator for $\mathsf{P/poly}$ if it is an indistinguishability obfuscator for the class $\{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ of circuits of size at most λ.

2.6 Randomized Encodings

In this section, we recall the traditional definition of randomized encodings with simulation security [IK02, AIK06].

Definition 2.10 (Randomized encoding scheme for circuits)**.** A randomized encoding scheme RE consists of two PPT algorithms,

- $\hat{C}_x \xleftarrow{\$} \mathsf{REnc}(1^\lambda, C, x)$: On input a security parameter 1^λ, circuit C, and input x, REnc generates an encoding $\hat{C}_x$.
- $y = \mathsf{REval}(\hat{C}_x)$: On input $\hat{C}_x$ produced by REnc, REval outputs y.

Correctness: The two algorithms REnc and REval satisfy the following correctness condition: For all security parameters $\lambda \in \mathbb{N}$, circuit C, input x, it holds that,

$$\Pr\left[\hat{C}_x \xleftarrow{\$} \mathsf{REnc}(1^\lambda, C, x) : \ \mathsf{Eval}(\hat{C}_x) = C(x)\right] = 1$$

μ-Simulation Security: There exists a PPT algorithm RSim, such that, for every ensemble $\{C_\lambda, x_\lambda\}_\lambda$ where $|C_\lambda|, |x_\lambda| \leq \mathrm{poly}(\lambda)$, the following ensembles are μ-indistinguishable for all $\lambda \in N$.

$$\left\{\hat{C}_x \xleftarrow{\$} \mathsf{REnc}(1^\lambda, C, x) : \hat{C}_x\right\}_{\lambda \in \mathbb{N}}$$
$$\left\{\hat{C}_x \xleftarrow{\$} \mathsf{RSim}(1^\lambda, C(x), 1^{|C|}, 1^{|x|}) : \hat{C}_x\right\}_{\lambda \in \mathbb{N}}$$

where $C = C_\lambda$ and $x = x_\lambda$.

Furthermore, let $\mathcal{C}$ be a complexity class, we say that randomized encoding scheme RE is in $\mathcal{C}$, if the encoding algorithm REnc can be implemented in that complexity class.

2.7 Functional Encryption

We provide the indistinguishability-based security definition of a public-key functional encryption (FE) scheme, which originally appeared in [BSW11, O'N10].

2.7.1 Public-Key Functional Encryption

Syntax Let $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be ensembles of sets. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, where every function in the set $\mathcal{F}_\lambda$ maps inputs in $\mathcal{X}_\lambda$ to outputs in $\mathcal{Y}_\lambda$.

A public-key functional encryption scheme FE for $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ consists of four PPT algorithms $(\mathsf{FE.Setup}, \mathsf{FE.KeyGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$.

- *Setup:* $\mathsf{FE.Setup}(1^\lambda, \mathsf{pp})$ is an algorithm that on input a security parameter and some public parameter (e.g., description of bilinear pairing groups) outputs a master public key and a master secret key $(\mathsf{mpk}, \mathsf{msk})$.
- *Key Generation:* $\mathsf{FE.KeyGen}(\mathsf{msk}, f)$ on input the master secret key msk and the description of a function $f \in \mathcal{F}_\lambda$, outputs a secret key sk_f.
- *Encryption:* $\mathsf{FE.Enc}(\mathsf{mpk}, x)$ on input the master public key mpk and a message $x \in \mathcal{X}_\lambda$, outputs an encryption ct of x.
- *Decryption:* $\mathsf{FE.Dec}(\mathsf{sk}, \mathsf{ct})$ on input the secret key associated with f and an encryption of x, outputs $y \in \mathcal{Y}_\lambda$.

Correctness: We define perfect correctness here. For every λ, $f \in \mathcal{F}_\lambda$, $x \in \mathcal{X}_\lambda$, it holds that,

$$\Pr\left[\begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{ct} \xleftarrow{\$} \mathsf{FE.Enc}(\mathsf{mpk}, x) \\ \mathsf{sk} \xleftarrow{\$} \mathsf{FE.KeyGen}(\mathsf{msk}, f) \end{array} : f(x) = \mathsf{FE.Dec}(\mathsf{sk}, \mathsf{ct})\right] = 1$$

Indistinguishability Security. Indistinguishability security of a functional encryption requires that no adversary can distinguish the FE encryption of one input x_0 from that of another x_1, if the adversary only obtains secret keys for functions that yield the same outputs on x_0 and x_1, that is, for every secret key sk_f, it holds that $f(x_0) = f(x_1)$. In this work, we consider the restricted setting where only a single function key is released, though the associated function may have multiple output bits.

Definition 2.11 (1-key $\mathsf{Full\text{-}Sel\text{-}Ind}$-security)**.** A public-key functional encryption scheme $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KeyGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ for $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is *1-key* μ-$\mathsf{Full\text{-}Sel\text{-}Ind}$*-secure*, if for every sequence of functions $\{f_\lambda\}_{\lambda\in\mathbb{N}}$ where $f_\lambda \in \mathcal{F}_\lambda$, and every sequence of pairs of inputs $\{x^0_\lambda, x^1_\lambda\}_{\lambda\in\mathbb{N}}$ where $x^0_\lambda, x^1_\lambda \in \mathcal{X}_\lambda$ and $f_\lambda(x^0_\lambda) = f_\lambda(x^1_\lambda)$, the following distributions are μ-indistinguishable.

$$\left\{\begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, f_\lambda) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, x^0_\lambda) \end{array} : (\mathsf{mpk},\ \mathsf{sk},\ \mathsf{ct})\right\}_{\lambda\in\mathbb{N}}$$

$$\left\{\begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, f_\lambda) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, x^1_\lambda) \end{array} : (\mathsf{mpk},\ \mathsf{sk},\ \mathsf{ct})\right\}_{\lambda\in\mathbb{N}}$$

Note that our notion of fully-selective security is weaker than the notion of selective security in some papers in the literature (e.g., [GKP+13, ABSV15]), which only requires the adversaries to choose challenge inputs x_0, x_1 statically, but allows the adversaries to choose challenge function inputs adaptively. Intuitively, the notion of fully-selective security is sufficient for applications that are non-interactive, for instance, building IO from FE as in [AJ15, BV15].

Simulation Security We will also consider 1-key simulation-based security, which requires that the master public key, secret key for a function f, and ciphertext for an input x, can be simulated via a simulator receiving only f and $f(x)$.

Definition 2.12 (1-key Full-Sel-Sim-security)**.** A public-key functional encryption scheme $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KeyGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ for $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is *1-key* μ-Full-Sel-Sim*-secure*, if for every sequence of efficiently samplable distributions $\{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ over pairs (f, x) where $f \in \mathcal{F}_\lambda$ and x is in the domain of f, there exists a simulator Sim, such that, the following distributions are μ-indistinguishable.

$$\left\{ \begin{array}{c} (f,x) \leftarrow \mathcal{D}_\lambda \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, f) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, x) \end{array} : (\mathsf{mpk},\ \mathsf{sk},\ \mathsf{ct}) \right\}_{\lambda\in\mathbb{N}}$$

$$\left\{ (f,x) \leftarrow \mathcal{D}_\lambda \ : \ \mathsf{Sim}(f, f(x)) \right\}_{\lambda\in\mathbb{N}}$$

2.7.2 FE for P/poly, NC^1, and Compactness

Definition 2.13 (FE schemes for families of function classes)**.** Let $\{\mathcal{F}^I\}_{I\in\mathcal{I}}$ be a family of function classes. We say that $\mathcal{FE} = \{\mathsf{FE}^I\}_{I\in\mathcal{I}}$ is a family of (1-key) FE schemes for $\{\mathcal{F}^I\}_{I\in\mathcal{I}}$ with μ-Adap-security or μ-Full-Sel-security if for every function class $\mathcal{F}^I = \{\mathcal{F}^I_\lambda\}_{\lambda\in\mathbb{N}}$, FE^I is a (1-key) FE scheme for $\mathcal{F}^I$ with μ-Adap-security or μ-Full-Sel-security.

Moreover, define the following special cases: Let $\{\mathcal{F}^{N,S}\}_{\mathcal{N},\mathcal{S}}$ be a family of function classes indexed by arbitrary polynomials $N \in \mathcal{N}$ and $S \in \mathcal{S}$, such that, $\mathcal{F}^{N,S}_\lambda$ includes a subset of functions that can be computed by circuits with $N(\lambda)$-bit inputs and $S(\lambda)$ size. (For instance, $\mathsf{P/poly}^{N,S}_\lambda$ includes all circuits with $N(\lambda)$ input bits and $S(\lambda)$ size).

Compactness In the above definition of families of FE schemes for $\{\mathcal{F}^{N,S}\}$, algorithms in the FE schemes could run in polynomial time depending on the polynomial parameters, such as, N, S. In the literature, stronger efficiency requirements have been considered. In particular, the works of [AJ15, BV15] defined compact FE schemes, which requires the encryption time to be independent of the circuit size S of the functions.

Definition 2.14 (Compactness of FE schemes)**.** Let $\mathcal{FE} = \left\{\mathsf{FE}^{N,S}\right\}$ be a family of FE schemes for $\{\mathcal{F}^{N,S}\}$.

Compactness: We say that the functional encryption scheme $\mathcal{FE}$ is compact if there exists a polynomial p, such that, for every polynomials N, S, ciphertexts of $\mathsf{FE}^{N,S}$ have size $p(\lambda, N(\lambda), \log S(\lambda))$.

$(1-\varepsilon)$**-Sublinear Compactness (a.k.a.** $(1-\varepsilon)$**-Weakly Compactness):** We say that $\mathcal{FE}$ is $(1-\varepsilon)$-sublinearly compact, if there exists a polynomial p, such that, for every polynomials N, S, ciphertexts of $\mathsf{FE}^{N,S}$ have size $p(\lambda, N(\lambda)) \cdot S(\lambda)^{1-\varepsilon}$.

2.8 (Fully) Homomorphic Encryption

We give a definition of secret key fully homomorphic encryption following [Gen09a, Gen09b].

Definition 2.15 ((Fully) Homomorphic Encryption)**.** A fully homomorphic secret-key encryption (FHE) scheme FHE consists of the following four PPT algorithms:

Key generation: The algorithm $\mathsf{FHE.KeyGen}$ on input a security parameter 1^λ, outputs a key s.

Encryption: The algorithm $\mathsf{FHE.Enc}$ on input a key s and a message x, outputs a ciphertext ct.

Evaluation: The algorithm $\mathsf{FHE.Eval}$ on input a circuit C and a tuple of ciphertexts $(\mathsf{ct}_1, \ldots, \mathsf{ct}_t)$, outputs a ciphertext ct'.

Decryption: The algorithm $\mathsf{FHE.Dec}$ on input a key s and a ciphertext ct, outputs a message y.

A FHE scheme is required to have to following two properties:

Correctness: For all s output by $\mathsf{FHE.KeyGen}(1^\lambda)$, all circuits C, all messages $x_1, \ldots, x_t$, and all ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_t$ output by $\mathsf{FHE.Enc}(s, x_1), \ldots, \mathsf{FHE.Enc}(s, x_t)$, we have

$$\Pr\Big[\mathsf{ct}' \overset{\$}{\leftarrow} \mathsf{FHE.Eval}(C, (\mathsf{ct}_1, \ldots, \mathsf{ct}_t)) : \mathsf{FHE.Dec}(s, \mathsf{ct}') = C(x_1, \ldots, x_t)\Big] = 1 - \mathrm{negl}(\lambda).$$

Compactness: There is a polynomial p such that for all s output by $\mathsf{FHE.KeyGen}(1^\lambda)$, all circuits C, all messages $x_1, \ldots, x_t$, all ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_t$ output by $\mathsf{FHE.Enc}(s, x_1), \ldots,$ $\mathsf{FHE.Enc}(s, x_t)$, and all ct' output by $\mathsf{FHE.Eval}(C, (\mathsf{ct}_1, \ldots, \mathsf{ct}_t))$, we have that $|\mathsf{ct}'| \le p(\lambda)$ and $\mathsf{FHE.Dec}(s, \mathsf{ct}')$ runs in time bounded by $p(\lambda)$ (independently of C).

If the evaluation only allows circuits from a certain class (and correctness and compactness holds for such circuits), we call the scheme homomorphic for that class (instead of fully homomorphic).

Definition 2.16 (Leveled Homomorphic Encryption)**.** A family of homomorphic encryption schemes $\{\mathsf{HE}^{(d)}\}_{d \in \mathbb{N}}$, where for all $d \in \mathbb{N}$, $\mathsf{HE}^{(d)}$ is homomorphic for all circuits of depth at most d, is called leveled fully homomorphic if all $\mathsf{HE}^{(d)}$ use the same decryption circuit, and the computational complexity of all algorithms in $\mathsf{HE}^{(d)}$ is polynomial in λ, d, and (for the evaluation algorithm) the size of the circuit.

The definition of CPA-security for (fully) homomorphic encryption is identical to the definition for ordinary secret-key encryption:

Definition 2.17 (CPA-Security)**.** Let $\mathsf{FHE} = (\mathsf{FHE.KeyGen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ be a FHE scheme. We say FHE is μ-CPA-secure if for every PPT adversary A and for every sufficiently large λ, the advantage of A in the following game is bounded by $O(\mu(\lambda))$:

- The challenger runs $s \overset{\$}{\leftarrow} \mathsf{FHE.KeyGen}(1^\lambda)$.
- The adversary A with access to an encryption oracle $\mathsf{FHE.Enc}(s, \cdot)$ chooses a pair of messages x_0, x_1 of equal length and sends them to the challenger.
- The challenger samples a bit $b \overset{\$}{\leftarrow} \{0, 1\}$, computes $\mathsf{ct} \overset{\$}{\leftarrow} \mathsf{FHE.Enc}(s, x_b)$, and sends ct to A.

- The adversary A again has access to an encryption oracle $\mathsf{FHE.Enc}(s, \cdot)$, and finally outputs a bit b'.

The advantage of A is defined as

$$\mathsf{Advt}_A^{\mathsf{FHE}} := \left|2 \cdot \Pr[b' = b] - 1\right|.$$

2.8.1 Threshold Multi-Key FHE

We next give definitions for multi-key FHE [LATV12] and threshold multi-key FHE following [MW16].

Definition 2.18 (Multi-Key (Leveled) FHE)**.** A multi-key (leveled) FHE scheme consists of the following PPT algorithms:

- $\mathsf{MFHE.Setup}$ on input a security parameter 1^λ and circuit depth 1^d, outputs the system parameters params.
- $\mathsf{MFHE.KeyGen}$ on input the system parameters params, outputs a public key pk and a secret key sk.
- $\mathsf{MFHE.Enc}$ on input pk and a message x, outputs a ciphertext ct.
- $\mathsf{MFHE.Expand}$ on input a sequence of public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_N$, an index $i \in [N]$, and a fresh ciphertext c under the ith key pk_i, outputs an "expanded" ciphertext $\widehat{\mathsf{ct}}$.
- $\mathsf{MFHE.Eval}$ on input params, a boolean circuit C of depth at most d, and expanded ciphertexts $\widehat{\mathsf{ct}}_1, \ldots, \widehat{\mathsf{ct}}_\ell$, outputs an evaluated ciphertext $\widehat{\mathsf{ct}}$.
- $\mathsf{MFHE.Dec}$ on input params, a sequence of secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$, and a ciphertext $\widehat{\mathsf{ct}}$, outputs a message.

We require the following properties:

μ-Semantic Security: For any polynomial d and any two messages x_0, x_1, the following two distributions are μ-indistinguishable:

$$\left\{ \begin{array}{r} \mathsf{params} \xleftarrow{\$} \mathsf{MFHE.Setup}(1^\lambda, 1^{d(\lambda)}) \\ (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{MFHE.KeyGen}(\mathsf{params}) \end{array} : \big(\mathsf{params}, \mathsf{pk}, \mathsf{MFHE.Enc}(\mathsf{pk}, x_0)\big) \right\}_{\lambda \in \mathbb{N}}$$

$$\left\{ \begin{array}{r} \mathsf{params} \xleftarrow{\$} \mathsf{MFHE.Setup}(1^\lambda, 1^{d(\lambda)}) \\ (\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{MFHE.KeyGen}(\mathsf{params}) \end{array} : \big(\mathsf{params}, \mathsf{pk}, \mathsf{MFHE.Enc}(\mathsf{pk}, x_1)\big) \right\}_{\lambda \in \mathbb{N}}$$

Correctness and Compactness: Let $\mathsf{params} \xleftarrow{\$} \mathsf{MFHE.Setup}(1^\lambda, 1^d)$, let for all $i \in \{1, \ldots, N\}$, $(\mathsf{pk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{MFHE.KeyGen}(\mathsf{params})$, and let $x_1, \ldots, x_\ell$ be messages. For $I_1 \in [N], \ldots, I_\ell \in [N]$, let $\mathsf{ct}_i \xleftarrow{\$} \mathsf{MFHE.Enc}(\mathsf{pk}_{I_i}, x_i)$ and let $\widehat{\mathsf{ct}}_i \xleftarrow{\$} \mathsf{MFHE.Expand}((\mathsf{pk}_1, \ldots, \mathsf{pk}_N), I_i, \mathsf{ct}_i)$ for $i \in [\ell]$. Further let C be a circuit of depth at most d and let $\widehat{\mathsf{ct}} := \mathsf{MFHE.Eval}(C, (\widehat{\mathsf{ct}}_1, \ldots, \widehat{\mathsf{ct}}_\ell))$. Then the following holds:

Correctness of Expansion: $\mathsf{MFHE.Dec}\big(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N), \widehat{\mathsf{ct}}_i\big) = x_i$ for all $i \in [\ell]$.

Correctness of Evaluation: $\mathsf{MFHE.Dec}\big(\mathsf{params}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N), \widehat{\mathsf{ct}}\big) = C(x_1, \ldots, x_\ell)$.

Compactness: $\big|\widehat{\mathsf{ct}}\big|$ is polynomial in λ, d, and N (independently of C and ℓ).

Definition 2.19 (Threshold Multi-Key FHE)**.** A threshold multi-key FHE scheme is a multi-key FHE scheme with the following two additional algorithms:

- $\mathsf{MFHE.PartDec}$ on input an expanded ciphertext $\widehat{\mathsf{ct}}$, public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_N$, an index $i \in [N]$, and the ith secret key sk_i, outputs a partial decryption p_i.

- $\mathsf{MFHE.FinDec}$ on input partial decryptions $p_1, \ldots, p_N$, outputs a message.

We further require correctness and simulatability of partial decryptions defined as follows: Let $\mathsf{params} \overset{\$}{\leftarrow} \mathsf{MFHE.Setup}\big(1^\lambda, 1^d\big)$, $(\mathsf{pk}_i, \mathsf{sk}_i) \overset{\$}{\leftarrow} \mathsf{MFHE.KeyGen}(\mathsf{params})$ for $i \in [N]$, and let $x_1, \ldots, x_\ell$ be messages. Further let $I_1 \in [N], \ldots, I_\ell \in [N]$, and let $\mathsf{ct}_i \overset{\$}{\leftarrow} \mathsf{MFHE.Enc}(\mathsf{pk}_{I_i}, x_i)$ and $\widehat{\mathsf{ct}}_i \overset{\$}{\leftarrow} \mathsf{MFHE.Expand}\big((\mathsf{pk}_1, \ldots, \mathsf{pk}_N), I_i, \mathsf{ct}_i\big)$ for $i \in [\ell]$. Finally let C be a circuit of depth at most d and let $\widehat{\mathsf{ct}} := \mathsf{MFHE.Eval}\big(\mathsf{params}, C, \big(\widehat{\mathsf{ct}}_1, \ldots, \widehat{\mathsf{ct}}_\ell\big)\big)$. We then require:

Correctness of Decryption: For $p_i \overset{\$}{\leftarrow} \mathsf{MFHE.PartDec}\big(\widehat{\mathsf{ct}}, (\mathsf{pk}_1, \ldots, \mathsf{pk}_N), i, \mathsf{sk}_i\big)$, $i \in [N]$, we have $\mathsf{MFHE.FinDec}(p_1, \ldots, p_N) = C(x_1, \ldots, x_\ell)$ with probability 1.

μ-Simulatability of Partial Decryptions: There exists a PPT simulator Sim that on input $i \in [N]$, $(\mathsf{sk}_j)_{j \in [N] \setminus \{i\}}$, $\widehat{\mathsf{ct}}$, and $C(x_1, \ldots, x_\ell)$, produces a simulated partial decryption p_i' such that

$$\delta(p_i, p_i') \leq O(\mu(\lambda)),$$

where $p_i \overset{\$}{\leftarrow} \mathsf{MFHE.PartDec}\big(\widehat{\mathsf{ct}}, (\mathsf{pk}_1, \ldots, \mathsf{pk}_N), i, \mathsf{sk}_i\big)$. Here, the randomness for the statistical distance is only over the coins of Sim and $\mathsf{MFHE.PartDec}$, and all other values are fixed.

3 Definition of Pseudo Flawed-Smudging Generators

Definition 3.1. Let B, ℓ be positive integers. We say that a distribution $\mathcal{D}$ over $\mathbb{Z}^\ell$ is B-bounded, if $\mathrm{Support}(\mathcal{D}) \subseteq [-B, B]^\ell$.

Definition 3.2 (Smudging Distributions)**.** Let ℓ and B be positive integers and $\varepsilon \in [0, 1]$. We say a distribution $\mathcal{X}$ over $\mathbb{Z}^\ell$ is (B, ε)-smudging if for $X \overset{\$}{\leftarrow} \mathcal{X}$ and for all $e \in [-B, B]^\ell \cap \mathbb{Z}^\ell$, we have $\delta(X, X + e) \leq \varepsilon$.

Definition 3.3 (Bit-fixing distributions)**.** Let $\mathcal{D}$ be a distribution over strings in $\Delta^{\leq \ell}$ for some integer ℓ. Let $I \subseteq [\ell]$ be a set of indices, and x an arbitrary string in $\Delta^{|I|}$. Define $\mathcal{D}|_{x,I}$ to be the distribution of sampling $\overline{x}$ from $\mathcal{D}$ conditioned on $\overline{x}_I = x$. For convenience, we sometimes also write I as its characteristic vector v, where $v_i = 1$ iff $i \in I$.

We say that $\mathcal{D}$ is bit-fixing efficiently samplable, if $\mathcal{D}|_{x,I}$ is efficiently samplable for any x, I.

Definition 3.4 (Flawed-Smudging Distributions)**.** Let ℓ and m be positive integers and let $\mathcal{X}$ and $\mathcal{V}$ be distributions over $\mathbb{Z}^\ell$ and $\mathbb{Z}^m$, respectively. Further let for $i \in [\ell]$, $\Phi_i \subseteq [m]$ and let $E_i \colon \mathbb{Z}^{|\phi_i|} \to \mathbb{Z}$ be functions. For $I \subseteq [\ell]$, let $\Phi_I := \bigcup_{i \in I} \Phi_i$. Define $E \colon \mathbb{Z}^m \to \mathbb{Z}^\ell$ as

$$E(v_1, \ldots, v_m) = \big(E_1\big((v_j)_{j \in \phi_1}\big), \ldots, E_\ell\big((v_j)_{j \in \phi_\ell}\big)\big).$$

We say that $\mathcal{X}$ is (K,μ)-flawed-smudging for $(E,\mathcal{V})$, if there exist randomized predicates $\big\{\mathrm{BAD}_i : \mathbb{Z}^{\ell+1} \to \{0,1\}\big\}_{i\in[\ell]}$ such that the following two distributions are identical,

$$\mathcal{D}_1 = \left\{ \begin{array}{c} V \leftarrow \mathcal{V} \\ X \leftarrow \mathcal{X} \\ \mathrm{bad} = \big(\mathrm{bad}_i \leftarrow \mathrm{BAD}_i(E_i(V_{\Phi_i}), X)\big)_{i\in[\ell]} \end{array} \quad : \ (V,\ E(V)+X,\ \mathrm{bad}) \right\},$$

$$\mathcal{D}_2 = \left\{ \begin{array}{c} V \leftarrow \mathcal{V} \\ X \leftarrow \mathcal{X} \\ \mathrm{bad} = \big(\mathrm{bad}_i \leftarrow \mathrm{BAD}_i(E_i(V_{\Phi_i}), X)\big)_{i\in[\ell]} \\ \overline{V} \leftarrow \mathcal{V}|_{V_{\Phi_{\mathrm{bad}}},\Phi_{\mathrm{bad}}} \end{array} \quad : \ \big(\overline{V},\ E(V)+X,\ \mathrm{bad}\big) \right\},$$

and in addition, with probability at least $1-\mu$, the 1-norm of bad is bounded by $|\mathrm{bad}|_1 \le K$.

We say that $\mathcal{X}$ is (K,μ)-flawed-smudging for B-bounded distributions, if it is (K,μ)-flawed-smudging for every $(E,\mathcal{V})$ such that $E(\mathcal{V})$ is B-bounded.

Definition 3.5. (Pseudo Flawed-Smudging Generator) Let n, m, K, B be polynomials. A family of (n,m)-pseudo flawed-smudging generator (PFG) is an ensemble of distributions $\mathcal{PFG} = \{\mathcal{PFG}_\lambda\}_{\lambda\in\mathbb{N}}$ satisfying the following properties:

Syntax: For every $\lambda \in \mathbb{N}$, every $(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}})$ in the support of $\mathcal{PFG}_\lambda$ defines a function $\mathrm{PFG} : \mathbb{Z}^{n(\lambda)} \to \mathbb{Z}^{m(\lambda)}$ and a poly(λ)-*bounded* distribution $\mathcal{D}_{\mathrm{seed}}$ over seeds.

Efficiency: There is a uniform Turing machine M satisfying that for every $\lambda \in \mathbb{N}$, every $(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}}) \in \mathsf{Support}(\mathcal{PFG}_\lambda)$ and seed $\in \mathsf{Support}(\mathcal{D}_{\mathrm{seed}})$, $M(\mathrm{PFG}, \mathrm{seed})$ runs in time poly(λ) and we have $M(\mathrm{PFG}, \mathrm{seed}) = \mathrm{PFG}(\mathrm{seed})$. Furthermore, $\mathcal{PFG}$ and all $\mathcal{D}_{\mathrm{seed}}$ in the support of $\mathcal{PFG}_\lambda$ are efficiently samplable.

(K,μ)-pseudo-flawed-smudging for B-bounded distributions: There exists an ensemble $\{\mathcal{X}_\lambda\}$ of distributions where $\mathcal{X}_\lambda$ is $(K(\lambda),\mu(\lambda))$-flawed-smudging for all $B(\lambda)$-bounded distributions, and the following ensembles are μ-indistinguishable:

$$\Big\{(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}}) \overset{\$}{\leftarrow} \mathcal{PFG}_\lambda; \mathrm{seed} \overset{\$}{\leftarrow} \mathcal{D}_{\mathrm{seed}} : (\mathrm{PFG}, \mathrm{PFG}(\mathrm{seed}))\Big\}_{\lambda\in\mathbb{N}},$$

$$\Big\{(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}}) \overset{\$}{\leftarrow} \mathcal{PFG}_\lambda; X \overset{\$}{\leftarrow} \mathcal{X}_\lambda : (\mathrm{PFG}, X)\Big\}_{\lambda\in\mathbb{N}}.$$

4 Functional Encryption for Constant Degree Polynomials

Fix an arbitrary maximum degree D, an appropriate sequence of modulus $p_2 \gg p_3 > \cdots p_d > p_{d+1} \cdots > p_D$ where $p_2 = \lambda^{w(1)}$, $\{p_3, \cdots, p_D\}$ are polynomially large as set in Remark 4.3 and (p_d, p_{d+1}) for every $3 \le d \le D-1$ are coprime, and an appropriate sequence of component-size bounds $s_2 > s_3 > \cdots > s_d > s_{d+1} > \cdots > s_D$ all poly(λ)-bounded. We construct special-purpose FE schemes for the following families of constant-degree polynomials:

- $\underline{\textit{Family } \mathsf{d}\mathcal{F}^{N,S} = \{\mathsf{d}\mathcal{F}^{N,S}_\lambda\}}$ indexed by arbitrary polynomials N, S, contains degree d polynomials f over $\mathbb{Z}_{p_d}$, with input-length $N = N(\lambda)$, size $S = S(\lambda)$. $f = \{f_i\}$ may have many output elements $\{f_i(x)\}$ and satisfies that each component f_i has size at most $s_d = s_d(\lambda)$.

Our FE schemes are constructed recursively, starting from the base case for quadratic $d = 2$ polynomials. We will show that the constructed schemes satisfies *special-purpose* sub-linear compactness, correctness and simulation-security.

4.1 Base Case: $2\mathsf{FE}^{N,S}$ for Quadratic Polynomials

We start with constructing a special-purpose FE scheme $2\mathsf{FE}^{N,S}$ for computing *quadratic* polynomials in $2\mathcal{F}^{N,S}$. Our FE scheme $2\mathsf{FE} = 2\mathsf{FE}^{N,S}$ relies on the following building blocks. Let λ be the global security parameter, $B < \overline{B}$ are appropriately set polynomials in λ specified below.

- A polynomial-stretch $(S^{1-\alpha}, S)$-PFG $(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}})$, whose output is indistinguishable to a $(\lambda^{\varepsilon_2}, \mu)$-flawed-smudging distribution $\mathcal{X}$ for B-bounded distributions. Importantly, PFG must be a quadratic polynomial over $\mathbb{Z}$, and its output to be $\overline{B}$-bounded, where $\overline{B} = \mathrm{poly}(B, \lambda)$. In Section 6.2, we provide a candidate of such generators.

  (Here for convenience, we use a single PFG $(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}})$. It can be replaced with a family of PFGs $\mathcal{PFG}$ by sampling $(\mathrm{PFG}, \mathcal{D}_{\mathrm{seed}}) \leftarrow \mathcal{PFG}_\lambda$ in the FE setup algorithm.)

- FE schemes $\{\mathsf{QFE}^{N,S}\}$ for computing *quadratic* functions f over $\mathbb{Z}_{p_2}$ of input-length N and size S, with the following properties:

  - *Linear compactness*: Ciphertext size is $\mathrm{poly}(\lambda)N$, depending linearly on the input length and is independent of the size of computation.
  - *Simulation Security*: The scheme satisfies 1-key $2^{-\lambda^{\varepsilon_1}}$-$\mathsf{Full}$-$\mathsf{Sel}$-$\mathsf{Sim}$-security.

  Such a scheme can be constructed from bilinear maps over $\mathbb{Z}_p$ as in [Lin17, BCFG17], together with a degree-2 PRG over $\mathbb{Z}_p$. However, these schemes achieves only weak correctness, in the sense that the outputs are computed in the target group of the bilinear map and can only be extracted in plaintext if the outputs resides some in polynomially-sized set.

  - *Weak Correctness*: For any correctly generated qsk for a quadratic function f and ciphertext qct for an input x, $\mathsf{QFE.Dec}(\mathsf{qsk}, \mathsf{qct}) = [f(x)]_T$, where $[f(x)]_T = g_T^{f(x)}$ is encoding of the output in the target group of the underlying bilinear map.

Construction of $2\mathsf{FE}^{N,S}$: Our special-purpose FE scheme is parameterized by a public parameter $\mathsf{pp} = (p_2, p_3, \cdots, p_D, s_2, \cdots s_D, n_{\mathrm{pad}})$ consisting all fixed parameters and a sufficiently large polynomial $n_{\mathrm{pad}} = \mathrm{poly}(s_2) > \mathrm{poly}(s_d)$.

- $2\mathsf{FE.Setup}(1^\lambda, \mathsf{pp})$: On input the security parameter and public parameter pp, generate a pair of QFE keys $(\mathsf{qmpk}, \mathsf{qmsk}) \leftarrow \mathsf{QFE.Setup}(1^\lambda)$, and output $\mathsf{mpk} = (\mathsf{pp}, \mathsf{qmpk}), \mathsf{msk} = \mathsf{qmsk}$.

- $2\mathsf{FE.Enc}(\mathsf{mpk}, x)$: On input $\mathsf{mpk} = (\mathsf{pp}, \mathsf{qmpk})$ and $x \in \mathbb{Z}_{p_2}^N$, do:

  - For every $3 \le \beta \le D$ and every $k \in [n_{\mathrm{pad}}]$, sample $\mathrm{seed}_{\beta,k} \leftarrow \mathcal{D}_{\mathrm{seed}}$.
  - Encrypt $\mathsf{qct} \leftarrow \mathsf{QFE.Enc}(\mathsf{qmpk}, (x, \{\mathrm{seed}_{\beta,k}\}_{\beta,k}))$.

  Output $2\mathsf{ct} = \mathsf{qct}$.

- $2\mathsf{FE.KeyGen}(\mathsf{msk}, f)$: On input msk and a quadratic polynomial f over $\mathbb{Z}_{p_2}^N$, do:

  - Let $g(x, \{\mathrm{seed}_{\beta,k}\}_{\beta,k})$ denote the following quadratic function: For every output element i of f,

$$g_i(x, \{\mathrm{seed}_{\beta,k}\}_{\beta,k}) := f_i(x) + \sum_{3 \le \beta \le D} p_\beta \times \left(\sum_{k \in [n_{\mathrm{pad}}]} \mathrm{PFG}_i(\mathrm{seed}_{\beta,k}) \right) .$$

For simplicity of notation, we suppress the subscript i and write

$$g(x, \{\mathrm{seed}_{\beta,k}\}_{\beta,k}) = f(x) + \sum_{3\leq\beta\leq D} p_\beta \times \sum_{k\in[n_{\mathrm{pad}}]} \mathrm{PFG}(\mathrm{seed}_{\beta,k}),$$ [1]

Note that since PFG is a quadratic polynomial over $\mathbb{Z}$, g is a quadratic polynomial over $\mathbb{Z}_{p_2}$.

– Generate key $\mathsf{qsk} \leftarrow \mathsf{QFE.KeyGen}(\mathsf{qmsk}, g)$.

Output $2\mathsf{sk} = \mathsf{qsk}$.

- $2\mathsf{FE.Dec}(2\mathsf{sk}, 2\mathsf{ct})$: Output $y = \mathsf{QFE.Dec}(\mathsf{qsk}, \mathsf{qct})$.

It is easy to see that the above construction satisfies the following properties.

- *Special-Purpose Sublinear Compactness:* It follows from the fact that PFG has polynomial-stretch, every $\mathrm{seed}_{\beta,k}$ has length $S^{1-\alpha}$. Then, by the linear compactness of QFE, we have that the size of ciphertext is

$$|2\mathsf{ct}| = |\mathsf{qct}| = \mathrm{poly}(\lambda)(N + n_{\mathrm{pad}} \times S^{1-\alpha}) = \mathrm{poly}(\lambda)(N + S^{1-\alpha}) \ ,$$

where the last equality follows from the fact that $n_{\mathrm{pad}} = \mathrm{poly}(\lambda)$. Note that unlike standard sublinear compactness which allows ciphertext size to grow polynomially with the length of the input $\mathrm{poly}(\lambda, N)S^{1-\alpha}$, the ciphertext size of $2\mathsf{FE}$ grows linearly in N. We refer to this as special-purpose $(1-\alpha)$-sublinear compactness.

- *Special-Purpose Weak Correctness:* By construction of g, we have

$$\begin{aligned} y &= g(x, \{\mathrm{seed}_{\beta,k}\}) \\ &= f(x) + \sum_{3\leq\beta\leq D} p_\beta \times \left(\sum_{k\in[n_{\mathrm{pad}}]} \mathrm{PFG}(\mathrm{seed}_{\beta,k}) \right) \\ &= f(x) + \sum_{3\leq\beta\leq D} p_\beta \times \Lambda_\beta \ , \end{aligned}$$

where $\Lambda_\beta = \sum_k \mathrm{PFG}(\mathrm{seed}_{\beta,k})$, and since the output of PFG_i is $\overline{B}$-bounded, Λ_β is $(\overline{B}n_{\mathrm{pad}} = \mathrm{poly}(\lambda))$-bounded. It then follows from the weak correctness of QFE that

$$2\mathsf{FE.Dec}(2\mathsf{sk}, 2\mathsf{ct}) = \left[y = (f(x) + \sum_{3\leq\beta\leq D} p_\beta \times \Lambda_\beta) \right]_T \ . \tag{1}$$

- *Special-Purpose Simulation Security:* It follows from the 1-key μ-$\mathsf{Full\text{-}Sel\text{-}Sim}$-security of QFE that if for every sequence of distributions $\{\mathcal{FN}_\lambda\}$ over $f \in 2\mathcal{F}^{N,S}_\lambda$, and every sequence

[1] This notation ignores the fact that PFG may have longer outputs than f. More precisely, g is defined component-wise as above.

of efficiently samplable input distributions $\{\mathcal{D}_\lambda\}$ over $\mathbb{Z}_{q_2}^N$, there exists a simulator Sim, such that, the following distributions are μ-indistinguishable.

$$\left\{ \begin{array}{c} f \leftarrow \mathcal{FN}_\lambda,\ x \leftarrow \mathcal{D}_\lambda \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow 2\mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ 2\mathsf{sk} \leftarrow 2\mathsf{FE.KeyGen}(\mathsf{msk}, f) \\ 2\mathsf{ct} \leftarrow 2\mathsf{FE.Enc}(\mathsf{mpk}, x) \end{array} : \ (\mathsf{mpk},\ 2\mathsf{sk},\ 2\mathsf{ct}) \right\}_{\lambda \in \mathbb{N}}$$

$$\left\{ \begin{array}{c} f \leftarrow \mathcal{FN}_\lambda,\ x \leftarrow \mathcal{D}_\lambda \\ \{\mathrm{seed}_{\beta,k} \leftarrow \mathcal{D}_{\mathrm{seed}}\}_{3 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \ \mathsf{Sim}(f,\ y = f(x) + \sum_{\beta,k} p_\beta \mathrm{PFG}(\mathrm{seed}_{\beta,k})) \right\}_{\lambda \in \mathbb{N}}$$

It further follows from the pseudo-flawed-smudging property of $\mathcal{PFG}$ that the second ensemble above is indistinguishable to the following, where the outputs of the PFG are replaced with samples from the $(\lambda^{\varepsilon_2}, \mu)$-flawed-smudging distribution $\mathcal{X}$ for B-bounded distributions.

$$\left\{ \begin{array}{c} f \leftarrow \mathcal{FN}_\lambda,\ x \leftarrow \mathcal{D}_\lambda \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}_\lambda\}_{3 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \ \mathsf{Sim}(f,\ y = f(x) + \sum_{\beta,k} p_\beta \Delta_{\beta,k}) \right\}_{\lambda \in \mathbb{N}} \tag{2}$$

4.2 Recursive Construction: $\mathsf{d{+}1FE}^{N,S}$ for degree $d+1$ Polynomials

We recursively construct our special-purpose FE scheme $\mathsf{d{+}1FE} = \mathsf{d{+}1FE}^{N,S}$ for computing degree $d+1$ polynomials over $\mathbb{Z}_{p_{d+1}}$ from our special purpose FE scheme $\mathsf{dFE} = \mathsf{dFE}^{N',S'}$ for computing degree d polynomials over $\mathbb{Z}_{p_d}$. The construction also makes use of a secret-key encryption scheme $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Eval}, \mathsf{HE.Dec})$ supporting homomorphic evaluation of *constant-degree* polynomials with special properties.

Secret-key Homomorphic Encryption for Constant-Degree Polynomials We describe a specific instantiation of HE using the simple secret-key encryption scheme based on the hardness of LWE — a ciphertext of $x \in \mathbb{Z}_{p_{d+1}}$ is of form $\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + p_{d+1}e + x$ over $\mathbb{Z}_{p_d}$. It was shown in [BV11] that this simple scheme supports homomorphic evaluation of constant-degree polynomials. In addition, as shown in [GKPV10, AKPW13] that when the secret $\mathbf{s}$ is binary from $\{-1, 1\}$, the hardness of LWE holds even if the secret $\mathbf{s}$ is only entropic instead of uniformly random. This means the semantic security of the scheme holds as long as $\mathbf{s}$ has sufficient entropy. Below we describe the scheme, and highlight the special properties that are useful for our construction of FE schemes.

- $\mathsf{HE.KeyGen}(1^n, \mathsf{hpp} = (p_{d+1}, p_d))$ on input a security parameter 1^n, outputs a random binary vector $\mathbf{s} \leftarrow \{-1, 1\}^n$.
- $\mathsf{HE.Enc}(\mathbf{s}, x)$, on input message $x \in \mathbb{Z}_{p_{d+1}}^N$, samples a vector $\mathbf{a} \leftarrow \mathbb{Z}_{p_d}^n$, and noise $e \leftarrow \chi$ where χ is a $poly(n)$-bounded noise distribution, and outputs

$$\mathsf{hct} = (\mathbf{a}, c = \langle \mathbf{a}, \mathbf{s} \rangle + p_{d+1}e + x) \pmod{p_d} .$$

Property 1 — Encryption with partial information of secret: Note that encryption depends only $\langle \mathbf{a}, \mathbf{s} \rangle$ instead of $\mathbf{s}$ entirely. This means if information related to generating a few ciphertexts is revealed, $\mathbf{s}$ still has high entropy. To make this explicit, we write

$$\mathsf{hct} = \mathsf{HE.Enc}'(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle, x, e) ,$$

In addition, note that $\mathsf{HE.Enc}'$ *has constant locality in its input.*

- $\mathsf{HE.Eval}(f, \mathsf{hct}_1, \cdots \mathsf{hct}_t)$ on input a degree $d+1$ polynomial $f : \mathbb{Z}_{p_{d+1}}^N \to \mathbb{Z}_{p_{d+1}}$, and a list of ciphertexts, expand f as a sum of monomials
$$f(\mathbf{x}) = \sum_j \mathrm{mnl}_j(\mathbf{x}) \ , \ \text{where } \mathrm{mnl}_j(\mathbf{x}) = \prod_{\gamma \in [d+1]} x_{j_\gamma}$$
output ciphertext
$$\mathsf{hct}^f = \sum_j \mathsf{hct}^{\mathrm{mnl}_j} \ , \ \text{where } \mathsf{hct}^{\mathrm{mnl}_j} = \otimes_{\gamma \in [d+1]} \mathsf{hct}_{j_\gamma} \pmod{p_d} \ ,$$
where $\otimes$ stands for vector tensor product.

- $\mathsf{HE.Dec}(\mathbf{s}, \mathsf{hct})$ learns the degree $d+1$ of homomorphic evaluation from $|\mathsf{hct}|$, and outputs
$$y = \langle \mathbf{s}^{\otimes d+1}, \mathsf{hct} \rangle \pmod{p_d} \ , \ \text{where } \mathbf{s}^{\otimes d} = \otimes_{\gamma \in [d+1]} \mathbf{s} \ .$$
Note that the actual message can be extracted as $y \bmod p_{d+1}$. For convenience of notation, we let $\mathsf{HE.Dec}$ output directly y.

Property 2–Additive and Local Structure of Noise By construction, for hct^f *derived from ciphertexts* $\mathsf{hct}_1 \cdots \mathsf{hct}_t$ *described above,*
$$\begin{aligned} \mathsf{HE.Dec}(\mathbf{s}, \mathsf{hct}^{\mathrm{mnl}_j}) &= \mathrm{mnl}_j(\mathbf{x}) + e^{\mathrm{mnl}_j}(x_{j_1}, \cdots, x_{j_{d+1}}, e_{j_1}, \cdots, e_{j_{d+1}}) \ , \\ \mathsf{HE.Dec}(\mathbf{s}, \mathsf{hct}^f) &= f(\mathbf{x}) + e^f = f(\mathbf{x}) + \sum_j e^{\mathrm{mnl}_j} \end{aligned}$$
where x_i, e_i *are respectively the input and noise underlying the i'th input ciphertext* hct_i*, and* $e^{\mathrm{mnl}_j}(x_{j_1}, \cdots, x_{j_{d+1}}, e_{j_1}, \cdots, e_{j_{d+1}})$ *is the noise underlying* $\mathsf{hct}^{\mathrm{mnl}_j}$*, which depends only on the inputs and noises underlying the ciphertexts* $\mathsf{hct}_{j_1}, \cdots, \mathsf{hct}_{j_{d+1}}$ *involved in computing this monomial.*

Since the noise distribution is $\mathrm{poly}(n)$*-bounded, every* e^{mnl_j} *is also* $\mathrm{poly}(n)$*-bounded, where the polynomial depends on* $d+1$*.*

Property 3 Decomposing homomorphic evaluation and decryption: For every polynomial $f(\mathbf{x})$ *of degree* $d+1$*, we define the following polynomials*
$$\mathsf{HE.Dec}(\mathbf{s}, \mathsf{HE.Eval}(f, (\mathsf{hct}_1, \cdots, \mathsf{hct}_t))) =: \hat{f}_1(\mathbf{s}, (\mathsf{hct}_1, \cdots, \mathsf{hct}_t)) + \hat{f}_2(\mathsf{hct}_1, \cdots, \mathsf{hct}_t) \ ,$$
The total degree of $\hat{f}_1$ *is* $d+1$ *and its degree in* $(\mathsf{hct}_1, \cdots, \mathsf{hct}_t)$ *is d. Hence there is a degree d function* $\tilde{f}_1$ *s.t.*
$$\tilde{f}_1((\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}_1, \cdots, \mathsf{hct}_t, 1)) := \hat{f}_1(\mathbf{s}^{\otimes d+1}, (\mathsf{hct}_1, \cdots, \mathsf{hct}_t)) \ .$$
When having multiple functions $f = \{f_i\}$*, one can perform homomorphic evaluation and decryption component-wise.*
$$\begin{aligned} &\mathsf{HE.Dec}(\mathbf{s}, \mathsf{HE.Eval}(f_i, (\mathsf{hct}_1, \cdots, \mathsf{hct}_t))) \\ &\qquad = \hat{f}_{2,i}(\mathsf{hct}_1, \cdots, \mathsf{hct}_t) + \tilde{f}_{1,i}((\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}_1, \cdots, \mathsf{hct}_t, 1)) \ . \qquad (3) \end{aligned}$$
For convenience, below we often suppress the subscript i. By construction of HE*,* $\tilde{f}_2$ *is a degree d polynomial over* $\mathbb{Z}_{p_d}$*. For each component* $\tilde{f}_{2,i}$*, its size and input-length are only a multiplicative* $\mathrm{poly}(\lambda)$ *factor larger than the size and input-length of* f_i*.*

Property 4 — Robustness under entropic secrets *Semantic security of* HE *follows directly from the hardness of LWE with binary secrets. Therefore, by the robustness of LWE [GKPV10, AKPW13], semantic security holds as long as the binary LWE secret has sufficient entropy. More precisely,*

Lemma 4.1 (Robustness of HE)**.** *Let $p_d > p_{d+1}$ be co-prime integers. By the μ-indistinguishability of $\mathsf{LWE}^{\mathsf{WL}(1,k)}_{n,m,p_d,\chi}$ (Definition 2.4), it holds that for all efficiently samplable correlated random variables $(\mathbf{s}, \mathrm{aux})$, where the support of $\mathbf{s}$ is $\{-1,1\}^n$ and $H_\infty(\mathbf{s} \mid \mathrm{aux}) \geq k$, the following distributions are μ-indistinguishable*

$$(\mathrm{aux}, \{\mathsf{hct} \leftarrow \mathsf{HE.Enc}(\mathbf{s}, x_i)\}_{i\in[m]}) \quad (\mathrm{aux}, \{\mathsf{hct} \leftarrow \mathsf{HE.Enc}(\mathbf{s}, 0)\}_{i\in[m]}) \ ,$$

where $\{x_i\}_{i\in m}$ are arbitrary messages in $\mathbb{Z}_{p_{d+1}}$ and the encryption randomness is independent of $(\mathbf{s}, \mathrm{aux})$.

It was shown first in [GKPV10] that the weak and leaky LWE assumption is implied by standard LWE. However, their result requires super-polynomial modulus (p_d here) and modulus-to-noise ratio, which is insufficient for our purpose. Fortunately, a later work by [AKPW13] improved the result to work with polynomial modulus and modulus-to-noise ratio. We recall their theorem.

Theorem 4.2 ([AKPW13, Theorem B.5])**.** *Let k, ℓ, m, n, β, γ, σ, and p_d be integers, and let Ψ be a distribution (all parameterized by λ) such that $\Pr_{x \overset{\$}{\leftarrow} \Psi}[|x| \geq \beta] \leq \mathrm{negl}(\lambda)$ and $\sigma \geq \beta\gamma nm$. Further let χ_σ be either the discrete Gaussian distribution with standard deviation σ, or the uniform distribution over $[-\sigma, \sigma] \cap \mathbb{Z}$. Assuming that the $\mathsf{LWE}_{\ell,m,p_d,\Psi}$-assumption holds, the weak and leaky $\mathsf{LWE}^{\mathsf{WL}(\gamma,k)}_{n,m,p_d,\chi_\sigma}$-assumption holds if $k \geq (\ell + \Omega(\lambda))\log(p_d)$, with polynomial security loss.*

Remark 4.3. In particular, the modulus p_d, which needs to be larger than the noises from χ_σ, can be set to a polynomial, depending on n, γ, and β all bounded by fixed polynomials in λ, and the number m of LWE samples. Since in our construction of FE, the number of LWE samples generated is polynomial (looking ahead, it depends on the input length N and size S of the computation), p_d is polynomial.

Construction of d+1FE Using our special purpose FE scheme $\mathsf{dFE} = \mathsf{dFE}^{N',S'}$ for degree d polynomials with appropriate N', S' set below, and the homomorphic encryptions scheme HE for constant degree computation above, our $\mathsf{d{+}1FE} = \mathsf{d{+}1FE}^{N,S}$ scheme proceeds as follows:

- $\mathsf{d{+}1FE.Setup}(1^\lambda, \mathsf{pp})$: On input the security parameter 1^λ and public parameter pp, generate a pair of dFE keys $(\mathsf{dmpk}, \mathsf{dmsk}) \leftarrow \mathsf{dFE.Setup}(1^\lambda, \mathsf{pp})$, and output $\mathsf{mpk} = (\mathsf{pp}, \mathsf{dmpk}), \mathsf{msk} = \mathsf{dmsk}$.
- $\mathsf{d{+}1FE.Enc}(\mathsf{mpk}, x)$: On input $\mathsf{mpk} = (\mathsf{pp}, \mathsf{dmpk})$ and $x \in \mathbb{Z}^N_{p_{d+1}}$, do:
  - Sample $\mathbf{s} \leftarrow \mathsf{HE.KeyGen}(1^n, \mathsf{hpp} = (p_{d+1}, p_d))$, where $n(\lambda) = \mathrm{poly}(\lambda)$ is an appropriate polynomial in the global security λ.
  - Encrypt $\mathsf{hct} = \{\mathsf{hct}_i = \mathsf{HE.Enc}'(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle, x_i \ ; \ e_i)\}_{i\in[N]}$.
  - Generate message $X = (\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}, 1)$, and encrypt $\mathsf{dct} \leftarrow \mathsf{dFE.Enc}(\mathsf{dmpk}, X)$.

  Output $\mathsf{ct} = (\mathsf{hct}, \mathsf{dct})$.

  (Note that $|\mathsf{hct}| = \mathrm{poly}(\lambda)N$, $X \in \mathbb{Z}^{|X|}_{p_d}$, and $|X| = \mathrm{poly}(\lambda)N$.)

- $\mathsf{d{+}1FE.KeyGen}(\mathsf{msk}, f)$: On input $\mathsf{msk} = \mathsf{dmsk}$, and a degree $d+1$ polynomial f over $\mathbb{Z}_{p_{d+1}}^N$, output a key for the polynomial $\tilde{f}_1(X)$ associated with f (Equation 3), $\mathsf{dsk} \leftarrow \mathsf{dFE.KeyGen}(\mathsf{dmsk}, \tilde{f}_1)$.

  (Note that $\tilde{f}_1$ is a degree d polynomial over $\mathbb{Z}_{p_d}$; it has size $S' = \mathrm{poly}(\lambda)S$, input length $N' = |X| = \mathrm{poly}(\lambda)N$, and every component $\tilde{f}_{1i}$ has size $s_d = \mathrm{poly}(\lambda)s_{d+1}$.)

- $\mathsf{d{+}1FE.Dec}(\mathsf{sk}, \mathsf{ct})$: On input $\mathsf{sk} = \mathsf{dsk}$ and $\mathsf{ct} = (\mathsf{hct}, \mathsf{dct})$, output

  $$y = \hat{f}_2(\mathsf{hct}) \boxplus \mathsf{dFE.Dec}(\mathsf{dsk}, \mathsf{dct}) \ .$$

  where $\boxplus$ denotes component-wise addition between a scalar and an encoding in the target group of the bilinear map.

We first argue the succinctness and special-purpose correctness of the schemes, and then show that they satisfy special-purpose simulation-security. All proofs go by induction on d, and the base case $d = 2$ was proven in Section 4.1.

- Special-Purpose Sublinear Compactness: Assume that $\mathsf{dFE}^{N',S'}$ satisfies special-purpose $(1-\alpha)$-sublinear compactness, that is,

  $$|\mathsf{dct}| = \mathrm{poly}(\lambda)(N' + S'^{1-\alpha}) \ .$$

  Then, $\mathsf{d{+}1FE}^{N,S}$ constructed above also satisfies that

  $$|\mathsf{ct}| = \mathrm{poly}(\lambda)(N + S^{1-\alpha}) \ .$$

  By construction, $\mathsf{ct} = (\mathsf{hct}, \mathsf{dct})$, and we have

  $$\begin{aligned} |\mathsf{hct}| &= \mathrm{poly}(\lambda)|x| \ , \\ |\mathsf{dct}| &= \mathrm{poly}(\lambda)(N' + S'^{1-\alpha}) = \mathrm{poly}(\lambda)(N + S^{1-\alpha}) \ , \end{aligned}$$

  where the latter follows from the fact that $N' = \mathrm{poly}(\lambda)N$ and $S' = \mathrm{poly}(\lambda)S$. This concludes the proof.

- Special-Purpose Weak Correctness: Assume that $\mathsf{dFE}^{N',S'}$ satisfies special-purpose weak correctness that for any correctly generated secret key dsk for function $\tilde{f}$ and ciphertext dct for input X,

  $$\mathsf{dFE.Dec}(\mathsf{dsk}, \mathsf{dct}) = \left[\tilde{f}(X) + \sum_{d+1 \le \beta \le D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \le \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T ,$$

  where every $\tilde{\Lambda}_\beta$ is $(\overline{B} n_{\mathrm{pad}})$-bounded, and every $\tilde{Y}_\beta$ is $(\overline{B} + B)n_{\mathrm{pad}}$-bounded.

  Then $\mathsf{d{+}1FE}^{N,S}$ also satisfies that for any correctly generated secret key sk for function f and ciphertext ct for input x,

  $$\mathsf{d{+}1FE.Dec}(\mathsf{sk}, \mathsf{ct}) = \left[f(x) + \sum_{d+2 \le \beta \le D} p_\beta \times \Lambda_\beta + \sum_{3 \le \beta < d+2} p_\beta Y_\beta \right]_T ,$$

where again every Λ_β is $(\overline{B}n_{\text{pad}})$-bounded, and every Y_β is $(\overline{B}+B)n_{\text{pad}}$-bounded.

Since $\mathsf{sk} = \mathsf{dsk}$ for function $\tilde{f}_1$ associated with f, $\mathsf{ct} = (\mathsf{hct}, \mathsf{dct})$ encrypting respectively x and X, and

$$\begin{aligned}
\mathsf{d{+}1FE.Dec}(\mathsf{sk}, \mathsf{ct}) &= \hat{f}_2(\mathsf{hct}) \boxplus \mathsf{dFE.Dec}(\mathsf{dsk}, \mathsf{dct}) \\
&= \left[\hat{f}_2(\mathsf{hct}) + \tilde{f}_1(X) + \sum_{d+1 \le \beta \le D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \le \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T \\
&= \left[f(x) + e^f + \sum_{d+1 \le \beta \le D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \le \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T \\
&= \left[f(x) + \sum_{d+2 \le \beta \le D} p_\beta \times \tilde{\Lambda}_\beta + p_{d+1}(Y_{d+1} = \tilde{\Lambda}_{d+1} + e^f) + \sum_{3 \le \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T .
\end{aligned}$$

We claim that e^f is (Bn_{pad})-bounded, when B and n_{pad} are set to sufficiently large polynomial in λ. Then $Y_{d+1} = \tilde{\Lambda}_{d+1} + e^f$ is $(\overline{B}+B)n_{\text{pad}}$-bounded as desired.

For any $f = \{f_i\}$, $e^f = \{e^{f_i}\}$, where e^{f_i} is the noise in the ciphertext produced by homomorphically evaluating f_i. By the additive and local structure of noise of HE (Property 2), $e^{f_i} = \sum_j e^{\text{mnl}_{i,j}}$ where $\text{mnl}_{i,j}$ is the j'th monomial in function f_i, and $e^{\text{mnl}_{i,j}}$ is poly(λ)-bounded, depending on the degree $d+1$ and size of LWE noises from χ.

> *Set $B = B(\lambda)$ to be larger than $e^{\text{mnl}_{i,j}}$ for maximum degree D and noise distribution χ.*

In addition, as f_i has size s_{d+1}, it contains at most poly(s_{d+1}) monomials.

> *Set n_{pad} to be larger than the number of monomials in maximum degree D polynomials with size at most s_2 (recall $s_2 > s_{d+1}$).*

Therefore, e^{f_i} is Bn_{pad}-bounded.

- *Special-Purpose Simulation Security:* We start with defining special-purpose simulation security.

Definition 4.4. We say that $\mathsf{dFE}^{N,S}$ satisfies special-purpose μ-simulation security if for every distribution $\{\mathcal{FN}\}_\lambda$ over $f \in \mathsf{d}\mathcal{F}^{N,S}$, and every distribution $\{\text{inp}(\mathcal{R})\}_\lambda$ where inp is a *constant locality* function with range $\mathbb{Z}_{q_d}^N$ and $\mathcal{R}$ is a distribution over the domain of I, there exist correlated random variables (r_K, K, st) sampled by $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$, and an efficient simulator dSim, such that,

– the following distributions dReal (top) and dIdeal (bottom) are μ-indistinguishable, and

$$\left\{\begin{array}{c} f \leftarrow \mathcal{FN},\ r \leftarrow \mathcal{R},\ x \leftarrow \mathrm{inp}(r) \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{dFE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{dFE.KeyGen}(\mathsf{msk}, f) \\ \mathsf{ct} \leftarrow \mathsf{dFE.Enc}(\mathsf{mpk}, x) \end{array} : \ r,\ (\mathsf{mpk},\ \mathsf{sk},\ \mathsf{ct}) \right\}_{\lambda \in \mathbb{N}}$$

$$\left\{\begin{array}{c} f \leftarrow \mathcal{FN}, \\ (r_K, K, \mathrm{st}) \leftarrow \mathsf{d}\mathcal{D}_{\mathsf{Sim}}(f), \\ \bar{r} \leftarrow \mathcal{R}|_{r_K,K},\ \bar{x} = \mathrm{inp}(\bar{r}) \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \ \bar{r},\ \mathsf{dSim}\left(\begin{array}{c} (r_K, K, \mathrm{st}),\ f, \\ y = f(\bar{x}) + \sum_{\substack{d+1 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k} \end{array} \right) \right\}_{\lambda \in \mathbb{N}}$$

– with probability $1-\mu$, $|K| \le O(\lambda^{\varepsilon_2})$.

More intuitively, the special-purpose simulation security states that honestly generated $(\mathsf{mpk}, \mathsf{sk}, \mathsf{ct})$ of dFE can be simulated using *i)* the output $f(\bar{x})$ "perturbed" by some "noise" $\Delta_{\beta,k}$ (sampled from the $(\lambda^{\varepsilon_2}, \mu)$-flawed-smudging distribution $\mathcal{X}$ for B-bounded distributions associated with the PFG PFG), and *ii)* the distribution under which the input $\bar{x}$ is sampled — instead of sampling x randomly from $\mathrm{inp}(\mathcal{R})$, $\bar{x}$ is sampled using $\bar{r}$ that is partially fixed at locations K to r_K, where (r_K, K) are sampled from a distribution $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ together with a state st. Importantly, the set of fixed locations is small, bounded sublinear in λ.

As analyzed in Section 4.1, $\mathsf{2FE}^{N,S}$ satisfies the above definition w.r.t. a distribution $\mathsf{2}\mathcal{D}_{\mathsf{Sim}}$ that always outputs null, and a simulator $\mathsf{2Sim}$ established in Section 4.1. Next, we show inductively that special-purpose simulation security holds.

Lemma 4.5 (Induction)**.** *If for every polynomial N', S', $\mathsf{dFE}^{N',S'}$ satisfies $O(\mu)$-special-purpose simulation security, then for every polynomial N, S, $\mathsf{d{+}1FE}^{N,S}$ also satisfies $O(\mu)$-special-purpose simulation security.*

Therefore, all our schemes satisfies special-purpose simulation security.

Next, we first prove Lemma 4.5 and then based on the proof show a structural property of $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$.

4.2.1 Proof of Special Purpose Simulation Security of d+1FE

Fix $\mathsf{d{+}1FE}^{N,S}$, $\mathcal{FN}$, and $\mathrm{inp}(\mathcal{R})$ as in Definition 4.4. We want to show that there exists correlated random variables (r_K, K, st) sampled by $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$, and an efficient simulator $\mathsf{d{+}1Sim}$, such that, $\mathsf{d{+}1Real}$ (top) and $\mathsf{d{+}1Ideal}$ (bottom) are $O(\mu)$-indistinguishable, and

$$\left\{\begin{array}{c} f \leftarrow \mathcal{FN},\ r \leftarrow \mathcal{R},\ x = \mathrm{inp}(r) \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{d{+}1FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{d{+}1FE.KeyGen}(\mathsf{msk}, f) \\ \mathsf{ct} \leftarrow \mathsf{d{+}1FE.Enc}(\mathsf{mpk}, x) \end{array} : \ r,\ (\mathsf{mpk},\ \mathsf{sk},\ \mathsf{ct}) \right\}$$

$$\left\{\begin{array}{c} f \leftarrow \mathcal{FN}, \\ (r_K, K, \mathrm{st}) \leftarrow \mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}(f), \\ \bar{r} \leftarrow \mathcal{R}|_{r_K,K},\ \bar{x} = \mathrm{inp}(\bar{r}) \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \ \bar{r},\ \mathsf{d{+}1Sim}\left(\begin{array}{c} (r_K, K, \mathrm{st}),\ f, \\ y = f(\bar{x}) + \sum_{\substack{d+2 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k} \end{array} \right) \right\}$$

We show this via a sequence of hybrids $H_0, \cdots H_4$, where $H_0 = \mathsf{d{+}1Real}$ and $H_4 = \mathsf{d{+}1Ideal}$ for some $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$ and $\mathsf{d{+}1Sim}$ constructed along the way.

Hybrid H_0 is identical to $\mathsf{d{+}1Real}$. By construction of $\mathsf{d{+}1FE}$, H_0 is identical to:

$$\left\{\left(r,\ \left(\mathsf{hct}, \mathsf{dmpk},\ \mathsf{dsk}(\tilde{f}_1),\ \mathsf{dct}(\mathbf{X})\ \right)\right) \leftarrow \mathsf{Samp}_0\right\},$$ [2]

where Samp_0 does:

1. Sample $f \leftarrow \mathcal{FN},\ r \leftarrow \mathcal{R},\ x \leftarrow \mathrm{inp}(r)$.
2. Generate $\mathbf{s} \leftarrow \mathsf{HE.KeyGen}(1^n, \mathsf{hpp})$ and encrypt $\mathsf{hct} = \{\mathsf{HE.Enc}'(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle, x_i\ ;\ e_i)\}_{i \in [N]}$ using random $\mathbf{a}_i$'s.
3. Generate $(\mathsf{dmpk}, \mathsf{dmsk}) \leftarrow \mathsf{dFE.Setup}(1^\lambda, \mathsf{pp})$, $\mathsf{dsk} \leftarrow \mathsf{dFE.KeyGen}(\mathsf{msk}, \tilde{f}_1)$ for $\tilde{f}_1$ related to f, and $\mathsf{dct} \leftarrow \mathsf{dFE.Enc}(\mathsf{mpk}, X)$ for $X = (\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}, 1)$.

In order to invoke the special-purpose simulation security of dFE for $(\mathsf{dmpk}, \mathsf{dsk}, \mathsf{dct})$, we observe that the above distribution can be efficiently generated from the following distribution

We define $\mathsf{d}\mathcal{FN}$ and $\mathsf{dinp}(\mathsf{d}\mathcal{R})$ to be the distributions that sample the function $\tilde{f}_1$ and input X used by the dFE scheme:

$\mathsf{d}\mathcal{FN}$ samples $f \leftarrow \mathcal{FN}$, output $\tilde{f}_1$ associated with f

$\mathsf{d}\mathcal{R}$ samples $(\ r,\ \mathbf{s}, \mathbf{A} = \{\mathbf{a}_i\}, \mathbf{As} = \{\langle \mathbf{a}_i, \mathbf{s}\rangle\}, \mathbf{e} = \{e_i\})$ as in the steps 1, 2 of Samp_0

$\mathsf{dinp}\,(r, \mathbf{s}, \mathbf{A}, \mathbf{As}, \mathbf{e})$ computes $X = (\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}, 1)$ as in step 2, 3 of Samp_0

We observe that $\mathsf{d}\mathcal{R}$ is bit fixing efficiently samplable (i.e., for any subset of indexes K', and any $R_{K'}$, one can efficiently sample $\bar{R}$ conditioned on $\bar{R}_{K'} = R_{K'}$). Moreover, since both inp and $\mathsf{HE.Enc}'$ have constant locality, dinp also has constant locality. Consider the following dReal distribution w.r.t. the degree d scheme dFE.

$$\left\{\begin{array}{c} \tilde{f}_1 \leftarrow \mathsf{d}\mathcal{FN} \\ (R = (r, \mathbf{s}, \mathbf{A}, \mathbf{As}, \mathbf{e})) \leftarrow \mathsf{d}\mathcal{R} \\ X = \mathsf{dinp}(R) \\ (\mathsf{dmpk}, \mathsf{dmsk}) \leftarrow \mathsf{dFE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{dsk} \leftarrow \mathsf{dFE.KeyGen}(\mathsf{msk}, \tilde{f}_1) \\ \mathsf{dct} \leftarrow \mathsf{dFE.Enc}(\mathsf{mpk}, X) \end{array} \ :\ R,\ (\mathsf{dmpk},\ \mathsf{dsk}(\tilde{f}_1),\ \mathsf{dct}(X))\right\}$$

It is easy to see that Real can be efficiently reconstructed from dReal, as given R one can obtain r and compute $\mathsf{hct}(x)$.

It follows from the special-purpose simulation security of dFE that there exist a distribution $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ and an efficient simulator dSim, such that, dReal is $O(\mu)$-indistinguishable to the following dIdeal distribution

$$\left\{\begin{array}{c} \tilde{f}_1 \leftarrow \mathsf{d}\mathcal{FN} \\ ((R_{K'}, K', \mathrm{st}'), \bar{R}, \bar{X}) \leftarrow \mathsf{Samp}_1(\tilde{f}_1) \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} \ :\ \bar{R},\ \mathsf{dSim}\left(\begin{array}{c} (R_{K'}, K', \mathrm{st}'),\ \tilde{f}_1, \\ y' = \tilde{f}_1(\bar{X}) + \sum_{\substack{d+1 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k} \end{array}\right)\right\}$$

[2] We changed the order of components, without loss of generality.

where $\mathsf{Samp}_1(\tilde{f}_1)$ does:

1. Sample $(R_{K'}, K', \mathrm{st}') \leftarrow \mathsf{d}\mathcal{D}_{\mathsf{Sim}}(\tilde{f}_1)$
2. Sample $\bar{R} \leftarrow \mathcal{R}|_{R_{K'},K'}$ where $\bar{R}$ contains $\big(\bar{r}, \bar{\mathbf{e}}, \bar{\mathbf{s}}, \bar{\mathbf{A}}, \bar{\mathbf{A}}\bar{\mathbf{s}}\big)$.
3. Compute $\bar{x} = \mathrm{inp}(\bar{r})$, and $\bar{X} = \mathsf{dinp}(\bar{R}) = (\bar{\mathbf{s}}^{\otimes d+1}, 1) \otimes (\overline{\mathsf{hct}}, 1)$, where $\overline{\mathsf{hct}} = \{\mathsf{HE.Enc}'(\bar{\mathbf{a}}_i, \langle \bar{\mathbf{a}}_i, \bar{\mathbf{s}} \rangle, \bar{x}_i, \bar{e}_i)\}$.

Since H_0 can be reconstructed from dReal, it is $O(\mu)$-indistinguishable to the following hybrid H_1 reconstructed from dIdeal.

Hybrid H_1, based on dIdeal, samples as follows:

$$\left\{ \begin{array}{c} \tilde{f}_1 \leftarrow \mathsf{d}\mathcal{FN} \\ ((R_{K'}, K', \mathrm{st}'), \bar{R}, \bar{X}) \leftarrow \mathsf{Samp}_1(\tilde{f}_1), \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \right.$$
$$\left. \bar{r},\ \overline{\mathsf{hct}},\ \mathsf{dSim}\left(\begin{array}{c} (R_{K'}, K', \mathrm{st}'),\ \tilde{f}_1, \\ y' = \tilde{f}_1(\bar{X}) + \sum_{\substack{d+1 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k} \end{array} \right) \right\}$$

where $\bar{r}$ is contained in $\bar{R}$ and $\overline{\mathsf{hct}}$ in $\bar{X}$.

By property of HE, homomorphic evaluation of f followed by decryption (Property 3) can be decomposed into

$$\mathsf{HE.Dec}(\bar{\mathbf{s}}, \mathsf{HE.Eval}(f, \overline{\mathsf{hct}})) = \hat{f}_2(\overline{\mathsf{hct}}) + \tilde{f}_1(\bar{X}) = f(\bar{x}) + p_{d+1}\mathbf{e}^f\ ,$$
$$\text{where } \mathbf{e}^f = \{e^{f_i}\} \text{ and } \forall i,\ e^{f_i} = \sum_j e^{\mathrm{mnl}_{ij}}(\bar{x}, \bar{\mathbf{e}})$$

The second line follows from the additive and local structure of noise of HE (Property 2). By the same property, we also know that $e^{\mathrm{mnl}_{ij}}(\bar{x}, \bar{\mathbf{e}})$ depends only on variables $(\bar{x}_l, \bar{e}_l)$ s.t. $\bar{x}_l$ appears in mnl_{ij}. Thus, $e^{\mathrm{mnl}_{ij}}$ has constant locality $2(d+1)$.

Therefore, y' above can be written as:

$$y' = f(\bar{x}) - \hat{f}_2(\overline{\mathsf{hct}}) + p_{d+1}\left(\mathbf{e}^f + \sum_{k \in [n_{\mathrm{pad}}]} \Delta_{d+1,k} \right) + \sum_{\substack{d+2 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k}$$
$$\text{where } \forall i,\ e^{f_i} + \sum_k \Delta_{d+1,k,i} = \sum_k (e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i})\ .$$

(Recall that n_{pad} upper bounds the number of monomials contained in every component f_i; if for some $k \in [n_{\mathrm{pad}}]$, monomial mnl_{ik} does not exist, let $e^{\mathrm{mnl}_{ik}} = 0$.)

For every k, $\{e^{\mathrm{mnl}_{ik}}\}$ is B-bounded. On the other hand, each $\Delta_{d+1,k}$ is a sample from the $(\lambda^{\varepsilon_2}, \mu)$-flawed-smudging distribution $\mathcal{X}$ for exactly B-bounded distributions. This means, there are randomized predicates $\{\mathrm{BAD}_{k,i}\}$, such that, the following two distributions are identical for any $\tilde{f}_1$,

$$\mathcal{D}_1(\tilde{f}_1) = \left\{ \begin{array}{c} (\star, \bar{r}, \bar{\mathbf{e}}, \star) \leftarrow \mathsf{Samp}_1(\tilde{f}_1) \\ \{\ e^{\mathrm{mnl}_{ik}} = e^{\mathrm{mnl}_{ik}}(\mathrm{inp}(\bar{r}), \bar{\mathbf{e}})\ \}_{i,k} \\ \{\Delta_{d+1,k} \leftarrow \mathcal{X}\}_k \\ \{\ \mathrm{bad}_{k,i} \leftarrow \mathrm{BAD}_{k,i}(e^{\mathrm{mnl}_{ik}}, \Delta_{d+1,k})\ \}_{i,k} \end{array} \right. : \quad (\bar{r}, \bar{\mathbf{e}}),\ \left\{ e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i} \right\}_{k,i},\ \mathrm{bad}) \Bigg\}\ ,$$

$$\mathcal{D}_2(\tilde{f}_1) = \left\{ \begin{array}{c} \text{same sampling as in } \mathcal{D}_1 \\ (\star, \tilde{r}, \tilde{\mathbf{e}}, \star) \leftarrow \mathsf{Samp}_1(\tilde{f}_1)|_{(\bar{r},\bar{\mathbf{e}})_I, I} \end{array} \right. : \ (\tilde{r}, \tilde{\mathbf{e}}),\ \left\{ e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i} \right\}_{k,i},\ \mathrm{bad}) \Bigg\}\ . \tag{4}$$

where I is the set of indices l of variables in $\bar{r}, \bar{\mathbf{e}}$ that $e^{\mathrm{mnl}_{i,k}}$ for with $\mathrm{bad}_{k,i} = 1$ depend on. In addition, with probability $1 - \mu$, $|\mathrm{bad}|_1 = O(\lambda^{\varepsilon_2})$. Since both $e^{\mathrm{mnl}_{i,k}}$ and inp have constant locality, we have that $|I| = O(\lambda^{\varepsilon_2})$.

Hybrid H_2: Apply the above equality of distribution to hybrid H_1, we have that it is identical to distribution H_2 below, where $\bar{r}, \bar{\mathbf{e}}$ are replaced with $\tilde{r}, \tilde{\mathbf{e}}$ as sampled above everywhere, except in the sums $\{e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i}\}_{k,i}$.

$$\left\{ \begin{array}{c} \tilde{f}_1 \leftarrow \mathsf{d}\mathcal{FN}, \\ \text{Same sampling as in } \mathcal{D}_1(\tilde{f}_1) \\ ((R_{K''}, K'', \mathrm{st}''), \tilde{R}, \tilde{X}) \leftarrow \mathsf{Samp}_1(\tilde{f}_1)|_{(\bar{r},\bar{\mathbf{e}})_I, I} \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \ \tilde{r},\ \widetilde{\mathsf{hct}},\ \mathsf{dSim}\left((R_{K''}, K'', \mathrm{st}''),\ \tilde{f}_1, \tilde{y}\right) \right\}$$

$$\text{where } \tilde{y} = f(\tilde{x}) - \hat{f}_2(\widetilde{\mathsf{hct}}) + p_{d+1} \left\{ \sum_k \left(e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i} \right) \right\}_i + \sum_{\substack{d+2 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k}$$

By construction of $\mathsf{Samp}_1(\tilde{f}_1)$, the bit-fixing distribution $\mathsf{Samp}_1(\tilde{f}_1)|_{(\bar{r},\bar{\mathbf{e}})_I, I}$ proceeds as follows:

1. Sample $(R_{K''}, K'', \mathrm{st}'') \leftarrow \mathsf{d}\mathcal{D}_{\mathsf{Sim}}(\tilde{f}_1)$
2. Sample $\tilde{R} \leftarrow \left(\mathcal{R}|_{R_{K''}, K''}\right)|_{(\bar{r},\bar{\mathbf{e}})_I, I}$.

   Note that the bit fixing distribution simply composes, requiring sampling $\tilde{R} = (\tilde{r}, \tilde{\mathbf{e}}, \tilde{\mathbf{s}}\tilde{\mathbf{A}}, \tilde{\mathbf{A}}, \tilde{\mathbf{s}})$ at random conditioned on $\tilde{R}_{K''} = R_{K''}$ and $(\tilde{r}, \tilde{\mathbf{e}})_I = (\bar{r}, \bar{\mathbf{e}})_I$. Since $\mathcal{R}$ is bit-fixing efficient, this sampling can be done efficiently.
3. Compute $\tilde{x} = \mathrm{inp}(\tilde{r})$, and $\tilde{X} = \mathsf{dinp}(\tilde{R}) = (\tilde{\mathbf{s}}^{\otimes d+1}, 1) \otimes (\widetilde{\mathsf{hct}}, 1)$, where $\widetilde{\mathsf{hct}} = \{\mathsf{HE.Enc}'(\tilde{\mathbf{a}}_i, \langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle, \tilde{x}_i, \tilde{e}_i)\}$.

Consider the HE ciphertexts in two cases:

- Case 1: $\widetilde{\mathsf{hct}}_i = \mathsf{HE.Enc}'(\tilde{\mathbf{a}}_i, \langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle, \tilde{x}_i, \tilde{e}_i)$ has that *i)* $\tilde{\mathbf{a}}_i$ is fixed in $R_{K''}$, or *ii)* $\langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle$ is fixed in $R_{K''}$, or *iii)* $\tilde{e}_i$ is fixed in $(\bar{r}, \bar{\mathbf{e}})_I$.

  Let $\mathcal{S} = \mathcal{S}(K'', I)$ denote the indices of these ciphertexts, efficiently computable from K'' and I. We have that $|\mathcal{S}| \le |K''| + |I|$. With probability $1 - O(\mu)$, both $|K''|$ and $|I|$ are bounded by $O(\lambda^{\varepsilon_2})$, and so is $|\mathcal{S}|$.

- Case 2: for every $i \notin \mathcal{S}$, $\widetilde{\mathsf{hct}}_i$ is generated using randomly sampled $\tilde{\mathbf{a}}_i$ and $\tilde{e}_i$, and $\langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle$ is not leaked nor fixed.

Intuitively, ciphertext in $\mathcal{S}$ are generated using fixed values, and are potentially not hiding. However, since there are only a few $O(\lambda^{\varepsilon_2})$ of them, even if all information used for generating them are revealed, the secret key $\tilde{\mathbf{s}}$ still have high entropy. Therefore, by robustness of HE, ciphertexts in $\neg\mathcal{S}$, generated using random $\tilde{\mathbf{a}}_i$ and $\tilde{e}_i$ and entropy $\tilde{\mathbf{s}}$, remain semantically secure. This yields the next hybrid H_3.

Hybrid H_3 is the same as H_2 except that HE ciphertexts in $\neg\mathcal{S}$ encrypt 0 instead of $\tilde{x}_{\mathcal{S}}$.

$$\left\{ \begin{array}{c} \tilde{f}_1 \leftarrow \mathsf{d}\mathcal{FN}, \\ \text{Same sampling as in } \mathcal{D}_1(\tilde{f}_1) \\ ((R_{K''}, K'', \mathrm{st}''), \tilde{R}, \tilde{X}) \leftarrow \mathsf{Samp}_1(\tilde{f}_1)|_{(\bar{r}, \bar{\mathbf{e}})_I, I} \\ \mathcal{S} = \mathcal{S}(K'', I) \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} : \begin{array}{c} \\ \\ \\ \\ \tilde{r},\ \left(\widetilde{\mathsf{hct}}_S(\tilde{x}_S)\widehat{\mathsf{hct}}_{\neg\mathcal{S}}(\mathbf{0})\right),\ \mathsf{dSim}\left((R_{K''}, K'', \mathrm{st}''),\ \tilde{f}_1, \tilde{y}\right) \end{array} \right\},$$

where

$$\begin{aligned} \widetilde{\mathsf{hct}}_{\mathcal{S}}(x_{\mathcal{S}}) &= \{\widetilde{\mathsf{hct}}_i = \mathsf{HE.Enc}'(\tilde{\mathbf{a}}_i, \langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle, \tilde{x}_i, \tilde{e}_i)\}_{i \in \mathcal{S}}, \\ \widehat{\mathsf{hct}}_{\bar{\mathcal{S}}}(\mathbf{0}) &= \{\widehat{\mathsf{hct}}_i = \mathsf{HE.Enc}'(\tilde{\mathbf{a}}_i, \langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{s}} \rangle, 0, \tilde{e}_i)\}_{i \notin \mathcal{S}}, \\ \tilde{y} &= f(\tilde{x}) - \hat{f}_2(\left(\widetilde{\mathsf{hct}}_S(\tilde{x}_S)\widehat{\mathsf{hct}}_{\neg\mathcal{S}}(\mathbf{0})\right)) \\ &\qquad + p_{d+1}\left\{\sum_k \left(e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i}\right)\right\}_i + \sum_{\substack{d+2 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k}. \end{aligned}$$

We show that H_3 and H_4 are $O(\mu)$-indistinguishable following the μ-robustness of HE. We already argued that ciphertexts in $\neg\mathcal{S}$ are generated using random $\tilde{\mathbf{a}}_i$ and noise $\tilde{e}_i$. It remains to argue that secret $\tilde{\mathbf{s}}$ is efficiently samplable and has high-entropy.

Note that H_3 and H_4 make the same potentially *inefficient* sampling steps, namely, sampling from $\mathcal{D}_1$, $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ (first step in $\mathsf{Samp}_1|_{(\bar{r},\bar{\mathbf{e}})_I, I}$), and $\mathcal{X}$. The other sampling and computing steps are all efficient, including sampling from $\left(\mathcal{R}|_{R_{K''}, K''}\right)|_{(\bar{r},\bar{\mathbf{e}})_I, I}$ and computing $\tilde{x}, \tilde{X}$ (the second and third step in $\mathsf{Samp}_1|_{(\bar{r},\bar{\mathbf{e}})_I, I}$), and computing $\mathcal{S}$.

Let $\mathcal{T}$ be the set of possible values sampled from $\mathcal{D}_1$, $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ and $\mathcal{X}$. We know that for an $1 - O(\mu)$ fraction of $\mathcal{T}$, it holds that $|K''|$ and $|I|$ are bounded by $O(\lambda^{\varepsilon_2})$. Take any $t \in \mathcal{T}$ satisfying this. H_3 and H_4 conditioned on t being sampled are efficient. Furthermore, $\tilde{\mathbf{s}}$ has high entropy conditioned on t, $\tilde{r}$, and ciphertexts in $\mathcal{S}$,

$$H_\infty\left(\tilde{\mathbf{s}} \mid t\ \tilde{r}, \widetilde{\mathsf{hct}}_{\mathcal{S}}\right) \ge n - O(\lambda^{\varepsilon_2})\ .$$

Observe that $t, \tilde{r}$ contains information for computing the entire output of H_3 and H_4 except for $\widehat{\mathsf{hct}}_{\bar{\mathcal{S}}}(x_{\neg\mathcal{S}})$ in H_3 and $\widehat{\mathsf{hct}}_{\neg\mathcal{S}}(\mathbf{0})$ in H_4. Conditioned on t, by the robustness of

HE, these ciphertexts are μ-indistinguishable. Therefore, conditioned on t, H_3 and H_4 are μ-indistinguishable. Since this holds for a $1 - O(\mu)$ fraction of t, we conclude that H_3 and H_4 are $O(\mu)$-indistinguishable.

Next, we re-write H_3 in the form of the ideal distribution $\mathsf{d{+}1Ideal}$ for $\mathsf{d{+}1FE}$ scheme.

Hybrid H_4 is the $\mathsf{d{+}1Ideal}$ distribution with $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$ and $\mathsf{d{+}1Sim}$ specified below.

$$\left\{ \begin{array}{c} f \leftarrow \mathcal{FN},\ (r^*_K, K, \mathrm{st}) \leftarrow \mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}(f), \\ \tilde{r} \leftarrow \mathcal{R}|_{r^*_K, K},\ \tilde{x} = \mathrm{inp}(\tilde{r}) \\ \{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \le \beta \le D, k \in [n_{\mathrm{pad}}]} \end{array} \quad : \quad \tilde{r},\ \mathsf{d{+}1Sim}\left(\begin{array}{c} (r^*_K, K, \mathrm{st}),\ f, \\ y = f(\tilde{x}) + \sum_{\substack{d+2 \le \beta \le D \\ k \in [n_{\mathrm{pad}}]}} p_\beta \Delta_{\beta,k} \end{array} \right) \right\},$$

The distribution $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}(f)$ works as follows:

1. Sample $\Big((\bar{r}, \bar{\mathbf{e}}),\ \big\{e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i}\big\}_{k,i},\ \mathrm{bad}\Big) \leftarrow \mathcal{D}_1(\tilde{f}_1)$, where bad determines I.
2. Sample $(R_{K''}, K'', \mathrm{st}'') \leftarrow \mathsf{d}\mathcal{D}_{\mathsf{Sim}}(\tilde{f}_1)$.
3. Compute $\mathcal{S} = \mathcal{S}(K'', I)$ and let $\mathcal{S}'$ be the set of indices of randomness that determine inputs at location $\mathcal{S}$, that is, $u_{\mathcal{S}} = \mathrm{inp}(v_{\mathcal{S}})$.
4. Sample $(\tilde{R} = (\tilde{r}, \tilde{\mathbf{e}}, \tilde{\mathbf{s}}\tilde{\mathbf{A}}, \tilde{\mathbf{A}}, \tilde{\mathbf{s}})) \leftarrow \big(\mathcal{R}|_{R_{K''}, K''}\big)|_{(\bar{r}, \bar{\mathbf{e}})_I, I}$.
5. Set K to be the set of indices in $\tilde{r}$ that are in $(\mathcal{S}', K'', I)$, set $r^*_K = \tilde{r}_K$, and set st to include $(\tilde{\mathbf{e}}, \tilde{\mathbf{s}}\tilde{\mathbf{A}}, \tilde{\mathbf{A}}, \tilde{\mathbf{s}})$, $\big\{e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i}\big\}_{k,i}$, $(R_{K''}, K'', \mathrm{st}'')$.

The simulator $\mathsf{d{+}1Sim}((r^*_K, K, \mathrm{st}), f, y)$ samples the following components in H_3:

1. It generates $\Big(\widetilde{\mathsf{hct}}_S(\tilde{x}_S)\widehat{\mathsf{hct}}_{\bar{\mathcal{S}}}(\mathbf{0})\Big)$ by first computing $\tilde{x}_S = \mathrm{inp}(r^*_{\mathcal{S}'})$, generating ciphertexts $\widetilde{\mathsf{hct}}_{\mathcal{S}}(\tilde{x}_S)$ and $\widehat{\mathsf{hct}}_{\neg\mathcal{S}}(\mathbf{0})$ using $(\tilde{\mathbf{e}}, \tilde{\mathbf{s}}\tilde{\mathbf{A}}, \tilde{\mathbf{A}}, \tilde{\mathbf{s}})$ contained in st.
2. It generates $\mathsf{dSim}\Big((R_{K''}, K'', \mathrm{st}''),\ \tilde{f}_1, \tilde{y}\Big)$, using $(R_{K''}, K'', \mathrm{st}'')$ contained in st and

$$\tilde{y} = y - \hat{f}_2(\Big(\widetilde{\mathsf{hct}}_S(\tilde{x}_S)\widehat{\mathsf{hct}}_{\neg\mathcal{S}}(\mathbf{0})\Big)) + p_{d+1}\left\{\sum_k \Big(e^{\mathrm{mnl}_{ik}} + \Delta_{d+1,k,i}\Big)\right\}_i$$

It is easy to see that $\mathsf{d{+}1Sim}$ is efficiently computable.

By construction of $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$ and $\mathsf{d{+}1Sim}$, hybrids H_3 and H_4 are identically distributed.

By a hybrid argument, we have that $\mathsf{d{+}1Real}$ and $\mathsf{d{+}1Ideal}$ are $O(\mu)$-indistinguishable, which concludes that $\mathsf{d{+}1FE}$ satisfies special-purpose simulation security.

4.2.2 Property of DFE

In Section 5, we will use DFE, the scheme for the maximum degree $d = D$ set by the public parameter pp, to construct FE schemes for NC^1 computation. Here, we describe the special-purpose function and input distributions that will be used and prove a structural property about the distribution $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$ when using such distributions.

Special-Purpose Function Distribution We will use DFE for function distribution $\mathcal{FN}$ that samples f of form:

- $f[G] : \{0,1\}^n \to \{0,1\}^n$ is defined by *i)* a *fixed* collection of predicates $\{g_i\}_{i \in [m]}$, one for every output bit, and *ii)* an input-output dependency graph $G \leftarrow \mathcal{G}$ sampled from a distribution $\mathcal{G}$.

$$\forall\ (x, \mathrm{seed}) \in \{0,1\}^n,\ i, \qquad f_i(x, \mathrm{seed}) = g_i(x, \mathrm{seed}_{G(i)})\ .$$

  We refer to the second input as the seed.

- $f[G]$ has constant locality ℓ in seed, that is, for every i, $|G(i)| \leq \ell$.

In other words, the actual functions to be computed $\{g_i\}$ are fixed, but which seed bits are fed into each g_i is distributional.

Correspondingly, we will use an input distribution $\mathrm{inp}(\mathcal{R} \times \mathcal{U}) = \mathcal{R} \times \mathcal{U}$ (i.e., inp is the identity function), where $x \leftarrow \mathcal{R}$ is sampled arbitrarily and seed $\leftarrow \mathcal{U}$ uniformly randomly.

For every such special-purpose distributions, the special-purpose $O(\mu)$-simulation security of DFE implies the existence of $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$ and Sim, such that, the real DReal distribution is indistinguishable to the following ideal DIdeal distribution.

$$\mathsf{DIdeal} = \left\{ \begin{array}{c} f \leftarrow \mathcal{FN}, \\ ((x, \mathrm{seed})_K, K, \mathrm{st}) \leftarrow \mathsf{D}\mathcal{D}_{\mathsf{Sim}}(f), \\ (\bar{x}, \overline{\mathrm{seed}}) \leftarrow \mathcal{R}|_{(x,\mathrm{seed})_K, K}, \end{array} \quad : \quad (\bar{x}, \overline{\mathrm{seed}}),\ \mathsf{DSim}\left(((x, \mathrm{seed})_K, K, \mathrm{st}),\ f,\ y = f(\bar{x}, \overline{\mathrm{seed}})\right) \right\}_{\lambda \in \mathbb{N}}$$

With probability $1 - O(\mu)$, $|K| = O(\lambda^{\varepsilon_2})$. Let K_{seed} be the random variable representing the indexes of seed bits $\mathrm{seed}_{K_{\mathrm{seed}}}$ that are fixed in $(x, \mathrm{seed})_K$.

The following lemma says that the positions K_{seed} of fixed seed bits only "weakly depends" on the dependency graph G, in the sense that, there is a set $\mathcal{K}$ independent of G, s.t. $G(\mathcal{K})$ contains K_{seed}. This property will be very instrumental later in our construction of FE for NC_1.

Lemma 4.6. *For every λ, every superpolynomially small $\mu = \lambda^{-\omega(1)}$, every special-purpose distribution $\mathcal{FN}$, and every $\mathcal{R}$, there exists a random variable $\mathcal{K}$, such that,* i) *$\mathcal{K}$ is correlated with $(f[G] \leftarrow \mathcal{FN},\ (x_K, K, \mathrm{st}) \leftarrow \mathsf{D}\mathcal{D}_{\mathsf{Sim}}(f[G]))$, and* ii) *$\mathcal{K}$ is independent of G,* iii) *$K_{\mathrm{seed}} \subseteq G(\mathcal{K})$ and $|\mathcal{K}| = O(\lambda^{\varepsilon_2})$ with probability $1 - O(\sqrt{\mu})$.*

Proof. Let us analyze how $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$ decides which seed bits to fix. By construction, $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$ recursively calls $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ for smaller degree d. In the base case, $2\mathcal{D}_{\mathsf{Sim}}$ does not fix any bits. In each recursion step, $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$ fixes the bits fixed by $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ and adds new bits to be fixed according to a collection $\{\mathrm{BAD}^{d+1}_{k,i}\}$ of randomized predicates (see Equation (4)). Each predicate $\mathrm{BAD}^{d+1}_{k,i}(e^{\mathrm{mnl}_{ik}}, \Delta_{d+1,k})$ is evaluated on the noises in the HE ciphertext produced by homomorphic evaluation of the k'th monomial mnl_{ik} in function f^{d+1}_i computed using $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$. Thus, $\mathrm{BAD}^{d+1}_{k,i}$ depends only on input bits that mnl_{ik} depends on; if $\mathrm{BAD}^{d+1}_{k_i}$ evaluates to 1, these input bits would be fixed.

By construction, the function f^d that $\mathsf{d}\mathcal{D}_{\mathsf{Sim}}$ evaluates is determined by f^{d+1} that $\mathsf{d{+}1}\mathcal{D}_{\mathsf{Sim}}$ evaluates. In particular, $f^d = \tilde{f}^{d+1}_1$, operating on $(\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}, 1)$.

$$\begin{aligned} &\textit{If} \quad & f^{d+1}_i(x^{d+1}, \mathrm{seed}^{d+1}) &= g^{d+1}_i(x^{d+1}, \mathrm{seed}^{d+1}_{G^{d+1}_i})\ ,\ \text{for fixed } g^{d+1}_i \\ &&& \qquad\qquad\qquad\qquad \text{and distributional } G^{d+1}\ . \\ &\textit{Then} \quad & \tilde{f}^d_i((\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}, 1)) &= \tilde{g}^{d+1}_{i1}((\mathbf{s}^{\otimes d+1}, 1) \otimes (\mathsf{hct}(x^{d+1}), \mathsf{hct}(\mathrm{seed}^{d+1}_{G^{d+1}(i)}), 1)) \\ &&&= h^d_i(x^{d+1}, \mathrm{seed}^{d+1}_{G^{d+1}(i)})\ ,\ \text{for fixed } \tilde{h}^d_i\ . \end{aligned}$$

The last equality simply rewrites the second last line as evaluating a randomized function h_i^d on $x^{d+1}, \mathrm{seed}_{G^{d+1}(i)}$ (that samples other variables such as $\mathbf{s}$, $\mathbf{A}$, and $\mathbf{e}$ that g_i^{d+1} uses internally). In other words, f_i^d depends ("recursively") only on seed bits $\mathrm{seed}_{G^{d+1}(i)}$. Thus, $\mathrm{BAD}_{k,i}^d$ can also be written as a fixed randomized predicate depending only on seed bits $\mathrm{seed}_{G^{d+1}(i)}$. Given that the starting-point function $f_i^D = g_i(x, \mathrm{seed}_{G(i)})$ indeed has this structure, we have that every $\mathrm{BAD}_{k,i}^d$ can be written as a fixed randomized predicate $P_{k,i}^d(x, \mathrm{seed}_{G(i)})$, and if it evaluates to 1, a subset of x, seed_{G_i} is fixed.

In the analysis of the special purpose $O(\mu)$-simulation security of dFE, we showed that with probability $1 - O(\mu)$, the number of $\mathrm{BAD}_{k,i}^d$ that evaluates to 1 is bounded by $O(\lambda^{\varepsilon_2})$. Therefore, for a $1 - O(\sqrt{\mu})$ fraction of randomness r of $\{P_{k,i}^d\}$ and input x, conditioned on them being sampled, the number of $P_{k,i}^d(x, \mathrm{seed}_{G(i)}\ ;\ r)$ that evaluates to 1 is bounded by $O(\lambda^{\varepsilon_2})$, with probability $1 - O(\sqrt{\mu})$ over the choice of seed and G. For every such x, r, we have that the expectation of the sum of outputs of $P_{k,i}^d$ is bounded.

$$\mathop{\mathbb{E}}_{\mathrm{seed},G}\left[\sum_{i,k} P_{k,i}^d\left(x, \mathrm{seed}_{G(i)}\ ;\ r\right)\right] = O(\lambda^{\varepsilon_2}) \qquad \left(\text{as } \mu = \lambda^{-\omega(1)}\right) .$$

For any x, r, let $E_{k,i}^b = \mathbb{E}[P_{k,i}^d(x, \mathrm{seed}_{G(i)}\ ;\ r)]$ be the expectation of $P_{k,i}^d$ over the choice of seed and G. Observe that since G has locality ℓ and the marginal distribution of $\mathrm{seed}_{G(i)}$ is uniform, the expectation $E_{k,i}^b$ is either zero if $P_{k,i}^d(x, \star\ ;\ r)$ is a zero predicate, or at least $1/2^\ell$. Let $\mathcal{K}$ be the set of i s.t., for some d and i, $P_{k,i}^d(x, \star\ ;\ r)$ is non-zero and $E_{k,i}^b \geq 1/2^\ell$. We have that for an $1 - O(\sqrt{\mu})$ fraction of x, r, the number of non-zero predicates is bounded by $O(2^\ell \lambda^{\varepsilon_2})$.

We observe that $\mathcal{K}$ satisfy the conditions in the lemma statement: *i)* $\mathcal{K}$ is correlated f, $((x, \mathrm{seed})_K, K, \mathrm{st}) \leftarrow \mathsf{D}\mathcal{D}_{\mathsf{Sim}}$, but *ii)* $\mathcal{K}$ is independent of G (it depends on fixed randomized predicates $P_{k,i}^d$, their randomness r, and input x). *iii)* $K_{\mathrm{seed}} \subseteq G(\mathcal{K})$ as only non-zero predicate $P_{k,i}^d$ can lead to fixed seed bits $\mathrm{seed}_{G(i)}$; in addition, with probability $1 - O(\sqrt{\mu})$, $|\mathcal{K}| = O(2^\ell \lambda^{\varepsilon_2})$. □

5 Functional Encryption for NC_1

In this section, we construct sublinearly compact FE schemes for NC_1 with standard fully selective indistinguishability security, using a constant-locality PRG, the AIK randomized encoding [AIK04], our special-purpose FE scheme dFE constructed in Section 4, and a new primitive called bit-fixing homomorphic sharing. Below, we start with introducing the new primitive and constructing it from multi-key FHE in Section 5.1, and then move to the construction of FE for NC_1 in Section 5.2.

5.1 Bit-fixing Homomorphic Sharing

A bit-fixing homomorphic sharing scheme has the same syntax as a Homomorphic Secret Sharing (HSS) scheme introduced by [BGI15, BGI16] — it enables generating a secret sharing $x_1, \cdots, x_\lambda$ of an input v, and homomorphically evaluating a circuit C on each share separately to obtain a set of output shares $o_1, \cdots, o_\lambda$, from which the final output $y = C(v)$ can be reconstructed. However, the similarity stops here, and the efficiency and security requirements are different.

- Security: similar to homomorphic encryption, HSS assumes that the output shares are *private*, and the the shared value v is hidden when the adversary sees only a subset of t

input shares. In contrast, bit-fixing homomorphic sharing require v to be hidden even to adversaries knowing all output shares and t input shares. Furthermore, security needs to hold not just for honestly generated input shares, as the name suggests, but also for shares where a few bits are fixed to arbitrary values.

- Efficiency: HSS (and homomorphic encryption) schemes have trivial constructions if reconstruction from the output shares can be as complex as evaluating the circuit itself (the output share can be an additive sharing of v together with C). Therefore, HSS is only meaningful when the reconstruction is "simpler" than the computation itself. This is not the case for bit-fixing homomorphic sharing — the trivial construction of HSS does not satisfy the above security requirement — and the sharing and reconstruction procedure can take time proportional the circuit size.

We now present the formal definition:

Definition 5.1. A (t, μ)-bit-fixing homomorphic sharing scheme $\mathsf{BF} = (\mathsf{BFsetup}, \mathsf{BFshare}, \mathsf{BFeval}, \mathsf{BFdec})$ consists of four efficient algorithms satisfying the following.

Syntax: The four algorithms have the following syntax

- $\mathsf{BFsetup}(1^\lambda, 1^s, 1^d)$ is a randomized algorithm that, on input a security parameter λ and bounds s and d on the function size and depth, outputs a CRS crs.
- $\mathsf{BFshare}(\mathrm{crs}, v)$ is a randomized algorithm that on input crs and $v \in \{0,1\}^{\mathrm{poly}(\lambda)}$, outputs input shares $x_1, \cdots, x_\lambda$ of v. Let $n = n(\lambda)$ be the length of all input shares.
- $\mathsf{BFeval}(\mathrm{crs}, x_i, i, f)$ is a deterministic algorithm that on input crs, a share x_i and its index i, and a function f, represented as a circuit of size s and depth d, outputs an output share o_i.

  Without loss of generality, if f has multiple output bits, BFeval is invoked separately for each output bit. The collection of output shares form o_i.
- $\mathsf{BFdec}(\mathrm{crs}, \{o_i\}, f)$ is a deterministic algorithm that, on input crs, all output shares, and the function f, outputs a string y.

Correctness: For every $\lambda, s, d \in \mathbb{N}$, every input v, and every function f of size s and depth d,

$$\Pr\left[\begin{array}{c} \mathrm{crs} \leftarrow \mathsf{BFsetup}(1^\lambda, 1^s, 1^d) \\ (x_1, \cdots, x_\lambda) \leftarrow \mathsf{BFshare}(\mathrm{crs}, v) \\ \forall\ i \in [\lambda],\ o_i \leftarrow \mathsf{BFeval}(\mathrm{crs}, x_i, i, f) \end{array} : \mathsf{BFdec}(\mathrm{crs}, \{o_i\}, f) = f(v)\right] = 1$$

(t, μ)-bit-fixing-security For every polynomials s and d, every sufficiently large $\lambda \in \mathbb{N}$, $s = s(\lambda)$, and $d = d(\lambda)$, every pair of poly(λ)-bit inputs (v_0, v_1), every t-bit string $x^* \in \{0,1\}^t$, every set of t indexes $J \subseteq [n(\lambda)]$,[3], every set of t indexes $K \subseteq [\lambda]$, every function f of size s and depth d that does not separate v_0 and v_1 (i.e., $f(v_0) = f(v_1)$), and every PPT adversary A,

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ \mathrm{crs} \leftarrow \mathsf{BFsetup}(1^\lambda, 1^s, 1^d) \\ (x_1, \cdots, x_\lambda) \leftarrow \mathsf{BFshare}(\mathrm{crs}, v_b)|_{x^*,J} \\ \forall\ i \in [\lambda],\ o_i \leftarrow \mathsf{BFeval}(\mathrm{crs}, x_i, i, f) \end{array} : \begin{array}{l} A\big(\mathrm{Binary}(x)_J, J, \\ \quad \{x_i\}_{i \in K}, \{o_i\}_{i \in [\lambda]}\big) = b \end{array}\right] \leq \frac{1}{2} + \mu(\lambda),$$

[3] Formally, we should only quantify over x^* and J for which $\mathsf{BFshare}(\mathrm{crs}, v_b)|_{x^*,J}$ is a valid distribution, i.e., for which the support of $\mathsf{BFshare}(\mathrm{crs}, v_b)$ contains a bit string which coincides with x^* on the indices in J. We ignore this subtlety for ease of presentation.

where Binary(x) denotes the binary representation of $(x_1, \dots, x_\lambda)$ and Binary$(x)_J$ are the bits at positions in J.

5.1.1 Construction from Threshold Multi-Key FHE

A bit-fixing homomorphic sharing scheme BF scheme can be constructed from a threshold multi-key FHE scheme MFHE for functions outputting a bit, and a pseudorandom function PRF as follows:

- $\mathsf{BFsetup}(1^\lambda, 1^s, 1^d)$ runs $\mathsf{params} \xleftarrow{\$} \mathsf{MFHE.Setup}(1^\lambda, 1^d)$ and outputs $\mathrm{crs} := \mathsf{params}$.
- $\mathsf{BFshare}(\mathrm{crs}, v)$ on input $v \in \{0,1\}^{\mathrm{poly}(\lambda)}$, computes the following:
  - $(\mathsf{pk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{MFHE.KeyGen}(\mathrm{crs})$ for $i \in [\lambda]$,
  - an additive sharing $\mathrm{ss}_1, \dots, \mathrm{ss}_\lambda$ of v, i.e., sample uniform bit strings $\mathrm{ss}_1, \dots, \mathrm{ss}_{\lambda-1}$ of length $|v|$ each and set $\mathrm{ss}_\lambda = v \oplus \bigoplus_{i=1}^{\lambda-1} \mathrm{ss}_i$,
  - $\mathsf{ct}_i \xleftarrow{\$} \mathsf{MFHE.Enc}(\mathsf{pk}_i, \mathrm{ss}_i)$ for $i \in [\lambda]$,
  - $\widehat{\mathsf{ct}}_i \xleftarrow{\$} \mathsf{MFHE.Expand}((\mathsf{pk}_1, \dots, \mathsf{pk}_\lambda), i, \mathsf{ct}_i)$ for $i \in [\lambda]$.

  It finally samples PRF keys K_i for $i \in [\lambda]$ and outputs $(x_1, \dots, x_\lambda)$, where

  $$x_i := \Big(\big(\widehat{\mathsf{ct}}_j, \mathsf{pk}_j\big)_{j \in \lambda}, \mathsf{sk}_i, K_i\Big).$$

- $\mathsf{BFeval}(\mathrm{crs}, x_i, i, f)$ on input crs, $x_i = \big((\widehat{ct}_j, \mathsf{pk}_j)_{j\in\lambda}, \mathsf{sk}_i, K_i\big)$, i, and a function f with ℓ output bits, where f_j denotes the function computing the jth output bit, computes for all $j \in [\ell]$:
  - $\widehat{ct}^{(j)} = \mathsf{MFHE.Eval}(\mathrm{crs}, f'_j, \widehat{\mathsf{ct}}_1, \dots, \widehat{\mathsf{ct}}_\lambda)$, with $f'_j(\mathrm{ss}_1, \dots, \mathrm{ss}_\lambda) := f_j\big(\bigoplus_{i=1}^{\lambda} \mathrm{ss}_i\big) = f_j(v)$,
  - $r_{i,j} = \mathsf{PRF}(K_i, j)$,
  - $o_{i,j} = \mathsf{MFHE.PartDec}\big(\widehat{ct}^{(j)}, (\mathsf{pk}_1, \dots, \mathsf{pk}_\lambda), i, \mathsf{sk}_i; r_{i,j}\big)$, which means $r_{i,j}$ is used as the randomness for the algorithm $\mathsf{MFHE.PartDec}$.

  It outputs the share $o_i = (o_{i,1}, \dots, o_{i,\ell})$.

- $\mathsf{BFdec}(\mathrm{crs}, \{o_i\}_{i\in[\lambda]}, f)$ on input crs, $\{o_i\}_{i\in[\lambda]} = \{o_{i,j}\}_{i\in[\lambda], j\in[\ell]}$, and a function f, computes $y_j = \mathsf{MFHE.FinDec}(o_{1,j}, \dots, o_{\lambda,j})$ for $j \in [\ell]$, and outputs $y = (y_1, \dots, y_\ell)$.

Correctness of the construction follows directly from the correctness of decryption of MFHE. We next show that the scheme is also bit-fixing secure.

Theorem 5.2. *If* MFHE *is μ-semantically secure, has μ-simulatability of partial decryptions, and* PRF *is μ-pseudorandom, then* BF *is $\big(t, O(L \cdot \mu)\big)$-bit-fixing secure for $t \le (\lambda - 1)/2$, where L is an upper bound on the number of output bits of the function f.*

Proof. Let s and d be polynomials, $\lambda \in \mathbb{N}$, $s = s(\lambda)$, $d = d(\lambda)$, let v_0 and v_1 be poly(λ)-bit inputs, let $x^* \in \{0,1\}^t$, let $J \subseteq [n(\lambda)]$ and $K \subseteq [\lambda]$ with $|J| = |K| = t$, let f be a function of size s and depth d with $\ell \le L$ output bits and $f(v_0) = f(v_1)$, and let A be a PPT adversary. Now let H_0 be the $\big(t, O(L \cdot \mu)\big)$-bit-fixing-security experiment, i.e., $b \leftarrow \{0,1\}$, crs $\leftarrow$

$\mathsf{BFsetup}(1^\lambda, 1^s, 1^d)$, $(x_1, \cdots, x_\lambda) \leftarrow \mathsf{BFshare}(\mathrm{crs}, v_b)|_{x^*, J}$, $\forall\ i \in [\lambda]$, $o_i \leftarrow \mathsf{BFeval}(\mathrm{crs}, x_i, i, f)$, $b' \leftarrow A\big(\mathrm{Binary}(x)_J, J, \{x_i\}_{i \in K}, \{o_i\}_{i \in [\lambda]}\big)$, and let the output of H_0 be 1 if $b' = b$, and 0 otherwise. Note that since $|K| \leq t$ and x^* fixes only $t \leq (\lambda - 1)/2$ bits, there exists an $i_0 \in [\lambda]$ such that no bit within x_{i_0} gets fixed and $i_0 \notin K$.

Define H_1 to be identical to H_0 except that for $j \in [\ell]$, $r_{i_0, j}$ in the execution of $\mathsf{BFeval}(\mathrm{crs}, x_{i_0}, i_0, f)$ gets replaced by a truly random value. Let A_{PRF} be an adversary with access to an oracle, which is either PRF or a truly random function, and let A_{PRF} emulate the experiment toward A, where instead of running $r_{i_0, j} = \mathsf{PRF}(K_{i_0}, j)$, A_{PRF} obtains $r_{i_0, j}$ by querying its oracle on input j. Note that this emulation can be done without K_{i_0} since no bits in x_{i_0} get fixed and A does not get x_{i_0} as an input. Finally let A_{PRF} output 1 if A guesses b correctly, and 0 otherwise. If the oracle of A_{PRF} is PRF, the view of A is identical to its view in H_0, and if the oracle is a truly random function, its view is identical to the one in H_1. Hence μ-indistinguishability of PRF implies

$$|\Pr[H_0 = 1] - \Pr[H_1 = 1]| \leq O(\mu(\lambda)).$$

Next, we define for $k \in \{0, \ldots, \ell\}$ the hybrid $H_{2,k}$ to be identical to H_1 except that $o_{i_0, j}$ for $j \leq k$ is produced by the simulator Sim from μ-simulatability of partial decryptions, where Sim is given i_0, all secret keys except sk_{i_0}, $\widehat{\mathsf{ct}}^{(j)}$, and $f_j(v_0) = f_j(v_1)$. Then, $H_{2,0}$ is identical to H_1 and since the statistical distance between the $o_{i_0, j}$ in different hybrids is bounded by $O(\mu(\lambda))$, we have

$$\forall k \in [\ell] \quad |\Pr[H_{2,k-1} = 1] - \Pr[H_{2,k} = 1]| \leq O(\mu(\lambda)).$$

Finally let H_3 be identical to $H_{2,\ell}$ except that $\mathsf{BFshare}$ encrypts $\perp$ instead of ss_{i_0} to obtain ct_{i_0}. Consider the attacker A_{sem} against the semantic security of MFHE that is given params, a public key pk^*, and an encryption ct^* of either ss_{i_0} or $\perp$, and then emulates $H_{2,\ell}$ or H_3 by using pk^* as pk_{i_0} and instead of encrypting ss_{i_0} or $\perp$, it uses its input ct^*. Note that this can be done without sk_{i_0} since it is only used in the scheme to compute $\{o_{i_0, j}\}_j$ and these are simulated in both hybrids without sk_{i_0}. Finally let A_{sem} output 1 if A guesses b correctly, and 0 otherwise. We then have that the advantage of A_{sem} is bounded by $O(\mu(\lambda))$ and thus,

$$|\Pr[H_3 = 1] - \Pr[H_{2,\ell} = 1]| \leq O(\mu(\lambda)).$$

Note that by the property of the additive sharing, the view of A in H_3 is independent of v_b and thus independent of b. Hence, $\Pr[H_3 = 1] = 1/2$. Combined with the results above, this yields $\Pr[H_0 = 1] \leq 1/2 + O(\ell \cdot \mu)$ and concludes the proof. □

5.2 Construction of FE for NC_1

For any fixed logarithmic function $\mathrm{Dep}(\lambda) = O(\log \lambda)$, we construct a family of FE schemes $\{\mathsf{FE}^{N,S,\mathrm{Dep}}\}$ for computing the class of NC_1 circuits C with depth $\mathrm{Dep} = \mathrm{Dep}(\lambda)$, arbitrary polynomial input length $N = N(\lambda)$, and size $S = S(\lambda)$. We will show that our schemes are sublinearly compact, and satisfies 1-key fully-selective indistinguishability security. Our scheme will make use of the following building blocks w.r.t. $\alpha \in (0, 1/2)$, $\varepsilon_2 \in (0, 1)$, and $\mu = 2^{-o(\lambda^{\varepsilon_2})}$.

- A polynomial-stretch $(\mathrm{poly}(\lambda)S^{1-\alpha}, \mathrm{poly}(\lambda)S)$-PRG PRG with μ-indistinguishability. Importantly, it has constant locality $\ell = O(1)$, and the function $\mathsf{PRG} = (P, G)$ is specified by a predicate $P : \{0,1\}^\ell \to \{0,1\}$ and an input-output dependency graph G, such that,

  $$\forall i,\ \mathsf{PRG}_i(\mathrm{seed}) = P(\mathrm{seed}_{G(i)}).$$

  Moreover, all input nodes in the graph have degree bounded by $o(S^{1-\alpha})$.

- A $(O(\lambda^{\varepsilon_2}), \lambda^{\omega(1)}\mu)$-bit-fixing homomorphic sharing $\mathsf{BF} = (\mathsf{BFsetup}, \mathsf{BFshare}, \mathsf{BFeval}, \mathsf{BFdec})$.
- The AIK randomized encoding [AIK04] $\mathsf{RE} = (\mathsf{REnc}, \mathsf{REval})$ whose encoding algorithm for any function f, $\mathsf{REnc}(f, x\ ;\ r)$ has locality 1 in input bits and locality 3 in the random bits. The AIK randomized encoding are unconditionally and perfectly secure.
- Our special-purpose FE scheme $\mathsf{DFE}^{N',S'} = (\mathsf{DFE.Setup}, \mathsf{DFE.Enc}, \mathsf{DFE.KeyGen}, \mathsf{DFE.Dec})$ for computing degree D polynomials with appropriate input length N' and size S' set below, that is special-purpose $(1-\alpha)$-sublinearly compact and special-purpose $O(\mu)$-simulation secure.

We now describe our FE scheme $\mathsf{FE}^{N,S}$ for NC_1 computation. We in-line the analysis of its correctness and $(1-\alpha)$-sublinear compactness in italic font.

- $\mathsf{FE.Setup}(1^\lambda)$: Generate a pair of DFE keys $(\mathsf{Dmpk}, \mathsf{Dmsk}) \leftarrow \mathsf{DFE.Setup}(1^\lambda, \mathsf{pp})$, using public parameter $\mathsf{pp} = (p_2, p_3, \cdots, p_D, s_2, \cdots s_D, n_{\text{pad}})$, where D is a fixed constant set below, $p_D = 2$, and $s_D = O(1)$ is a fixed constant set below. The other parameters are set as in Section 4. (In particular, p_d is the HE ciphertext space modulus when the message space has modulus p_{d+1}, and $p_D < \cdots < p_3 < \text{poly}(\lambda) \ll p_2$, $s_d = \text{poly}(\lambda)s_{d+1}$, and $n_{\text{pad}} = \text{poly}(s_2)$.) Generate a CRS for the bit-fixing homomorphic sharing scheme $\text{crs} \leftarrow \mathsf{BFsetup}(1^\lambda)$.

  Output $\mathsf{mpk} = (\mathsf{Dmpk}, \text{crs})$, and $\mathsf{msk} = (\mathsf{Dmsk}, \text{crs})$.

- $\mathsf{FE.KeyGen}(\mathsf{msk}, C)$: Input circuit C has input-length N, size S, and depth Dep. For convenience, we assume that C has exactly S output bits, and every output bit is computed by C_i with some fixed polynomial size $s(\lambda) = \text{poly}(\lambda)$[4].

  - Generate a polynomial f as follows:
    - ∗ Divide the output bits of C into $M = S^{1-\alpha}$ (assume for convenience that M divides S) consecutive chunks $I_1, \cdots, I_M$, where chunk I_j includes outputs bits $(j-1)S/M + 1, \cdots, jS/M$. For every $j \in [m]$, let $C_{I_j}(v) = \{C_k\}_{k \in I_j}$ denote collection of circuits that computes output bits in chunk I_j.
    - ∗ For every $j \in [M]$ and $i \in [\lambda]$, let D_i^j be the circuit that on input the i'th share x_i^j of v, homomorphically evaluates C_{I_j}

      $$D_i^j(x_i^j) = \mathsf{BFeval}(\text{crs}, x_i^j, i, C_{I_j}) = o_i^j\ .$$

      Since BFeval performs homomorphic evaluation of each C_i separately, the component size of D_i^j is $\text{poly}(\lambda, s)$. Moreover the size of each input share x_i^j is also bounded by $\text{poly}(\lambda, s)$.
    - ∗ Choose a random permutation $\pi : [\lambda] \times [M] \times [\phi] \to [\lambda M \phi]$. For every $j \in [M]$ and $i \in [\lambda]$, let f_i^j be the following function

      $$f_i^j(x_i^j, \text{seed}) = \mathsf{REnc}(D_i^j, x_i^j\ ;\ \mathsf{PRG}_{\Pi_i^j}(\text{seed}))\ .$$

[4] If not, one can always use garbled circuits to turn C into another circuit where every output bit is computable in size $\text{poly}(\lambda)$.

Encoding D_i^j, x_i^j takes time $\mathrm{poly}(\lambda, s)S/M = \mathrm{poly}(\lambda)S/M$ and hence uses at most $\phi = \mathrm{poly}(\lambda)S/M$ random coins. $\mathsf{PRG}_{\Pi_i^j}(\mathrm{seed})$ contains ϕ PRG output bits at locations $\{\pi(i,j,k)\}_{k\in[\phi]}$ determined by the random permutation π. Overall, $|\mathsf{PRG}(\mathrm{seed})| = \mathrm{poly}(\lambda, s)S = \mathrm{poly}(\lambda)S$ and $|\mathrm{seed}| = \mathrm{poly}(\lambda)S^{1-\alpha}$.

Finally, set

$$f(\{\mathbf{x}^j = \{x_i^j\}_{i\in[\lambda]}\}_{j\in[M]},\ \mathrm{seed}) := \{f_i^j(x_i^j, \mathrm{seed})\}_{j,i}$$

The input length and size of f is $N' = |\{x_i^j\}| + |\mathrm{seed}| = \mathrm{poly}(\lambda, s)\lambda M + \mathrm{poly}(\lambda)S^{1-\alpha} = \mathrm{poly}(\lambda)S^{1-\alpha}$ and $S' = |f_i^j|\lambda M = \mathrm{poly}(\lambda)S$. Since REnc has constant locality $1 + 3\ell$, where ℓ is the locality of PRG, f also has constant locality (and hence constant degree); let D be this locality, and s_D the constant component-size of f.

Remark 5.3. *Let $\mathcal{FN}$ be the distribution of f. We observe that $\mathcal{FN}$ is special-purpose (as defined in Section 4.2.2), that is, f is defined by a* fixed *collection of predicate $\{g_i\}$ and an input-output dependency graph G_π sampled from some distribution $\mathcal{G}$. To see this, recall that every output bit i of* PRG *is computed using the same predicate P on a subset of seeds $G(i)$. Thus, f_i^j can be written as*

$$f_i^j(x_i^j, \mathrm{seed}) = \mathsf{REnc}(D_i^j, x_i^j\ ;\ \{P(\mathrm{seed}_{G(\pi(i,j,k))})\}_{k\in[\phi]}\ )$$

This means the collection of predicates $\{g_i\}$ depends on D_i^j, P, and REnc*, and the input-output dependency graph G_π depends on the random permutation π, and the dependency graph G associated with* PRG *and that associated with* REnc*. As* PRG, REnc *have constant locality, so is f.*

- Generate a DFE secret key of f, $\mathsf{Dsk} \leftarrow \mathsf{DFE.KeyGen}(\mathsf{Dmsk}, f)$

Output $\mathsf{sk} = \mathsf{Dsk}$.

- $\mathsf{FE.Enc}(\mathsf{mpk}, v)$: On input $\mathsf{mpk} = (\mathsf{Dmpk}, \mathrm{crs})$ and $v \in \{0,1\}^\lambda$, do:
  - For every $j \in [M]$, share v using BF, $\mathbf{x}^j = \{x_i^j\}_{i\in[\lambda]} \leftarrow \mathsf{BFshare}(\mathrm{crs}, v, 1^s, 1^{\mathrm{Dep}})$.
  - Sample a random seed seed of length $\mathrm{poly}(\lambda)S^{1-\alpha}$.
  - Encrypt $X = (\{\mathbf{x}^j\}_j, \mathrm{seed})$ using DFE, $\mathsf{Dct} \leftarrow \mathsf{DFE.Enc}(\mathsf{Dmpk}, X)$.

  Output $\mathsf{ct} = \mathsf{Dct}$.

  (Sublinear Compactness Analysis: It follows from the special-purpose $(1-\alpha)$-sublinear compactness of DFE *that $|\mathsf{Dct}| = \mathrm{poly}(\lambda)(N' + S'^{1-\alpha}) = \mathrm{poly}(\lambda)S^{1-\alpha}$. Therefore,* FE *is $(1-\alpha)$-sublinearly compact.)*

- $\mathsf{FE.Dec}(\mathsf{sk}, \mathsf{ct})$: On input $\mathsf{sk} = \mathsf{Dsk}$ and $\mathsf{ct} = \mathsf{Dct}$, do
  - Decrypt DFE ciphertext and secret key $[z]_T = \mathsf{DFE.Dec}(\mathsf{Dsk}, \mathsf{Dct})$.

    (Correctness analysis: It follows from the special-purpose correctness of DFE *that*

$$\mathsf{DFE.Dec}(\mathsf{Dsk}, \mathsf{Dct}) = [z]_T = \left[f(X) + \sum_{3\le\beta\le D} p_\beta Y_\beta \right]_T ,$$

where every Y_β is $(\overline{B}+B)n_{\mathrm{pad}}$-bounded, and z is bounded by $\mathcal{B} = 1 + (\sum_{3\le\beta\le D} p_\beta)(\overline{B} + B)n_{\mathrm{pad}} = \mathrm{poly}(\lambda)$.)

- For every $z' \in [-\mathcal{B}, \mathcal{B}] \cap \mathbb{Z}$, test whether $[z]_T = [z']_T$. If the test succeeds with z', output bit $y = z' \bmod p_3 \bmod p_4 \cdots \bmod p_D$; otherwise, output $\perp$.

  (Correctness analysis, continued: we have that

$$\begin{aligned} y &= f(X) = \{y_i^j = f_i^j(x_i^j, \text{seed})\}_{j,i} \ , \\ y_i^j &= \mathsf{REnc}(D_i^j, x_i^j \ ; \ \mathsf{PRG}_{\pi(j,i)}(\text{seed})) \ . \end{aligned}$$

)

- Parse $y = \{y_i^j\}$, for every $j \in [M]$ and $i \in [\lambda]$, decode y_i^j using REval to obtain $o_i^j = \mathsf{REval}(y_i^j)$.

  (Correctness analysis, continued: By the correctness of $\mathfrak{R}$, we have that

$$o_i^j = \mathsf{REval}(y_i^j) = D_i^j(x_i^j) = \mathsf{BFeval}(\text{crs}, x_i^j, i, C_{I_j}) = o_i^j \ .$$

)

- For every $j \in [M]$, decode the output shares $\{o_i^j\}_{i \in [\lambda]}$ to obtain the actual output $u^j = \mathsf{BFdec}(\text{crs}, \{o_i^j\}, C_{I_j})$.

  (Correctness analysis, continued: By the correctness of BF*, we have that $u^j = C_{I_j}(v)$. This concludes the correctness analysis.)*

Output $u = \{u^j\}$.

Next, we show that FE satisfies 1-key fully-selective indistinguishability-security.

Theorem 5.4. *For any μ and ε_2 such that $\mu = 2^{-o(\lambda^{\varepsilon_2})}$. Assume that* PRG *satisfies μ-indistinguishability,* BF *satisfies $(O(\lambda^{\varepsilon_2}), \lambda^{\omega(1)}\mu)$-bit-fixing security,* $\mathsf{DFE}^{N',S'}$ *satisfies $O(\mu)$-special-purpose simulation security. Then,* $\mathsf{FE}^{N,S,\text{Dep}}$ *above satisfies $O(\mu)$-*Full-Sel-Ind*-security.*

Proof. Fix any NC_1 circuit $\{C\}_\lambda$ with input-length N, size S, depth Dep, component-size s, any pair of inputs $\{v_0, v_1\}_\lambda$ s.t. $C(v_0) = C(v_1)$, it suffices to show that for any efficient adversary A, the following probability is bounded by $1/2 + O(\mu)$ for sufficiently large λ and $\mu = \mu(\lambda)$.

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, C) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, v_b) \end{array} : \ A(\mathsf{mpk}, \ \mathsf{sk}, \ \mathsf{ct}) = b\right] \leq \frac{1}{2} + O(\mu) \ .$$

We prove the above via a sequence of hybrids that gradually change the distribution of $\mathsf{view}_A = (\mathsf{mpk}, \mathsf{sk}, \mathsf{ct})$ and argue that in the final hybrid, A cannot guess b with probability larger than the above bound.

Hybrid H_0: This hybrid generates view_A honestly as above, and additionally outputs X encrypted by DFE.

$$\mathsf{Real} = \left\{\begin{array}{c} b \leftarrow \{0,1\} \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda, \mathsf{pp}) \\ \mathsf{sk} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, f) \\ \mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, v_b) \end{array} : \ \{\mathbf{x}^j\}, \ \ \mathsf{view}_A = (\mathsf{mpk}, \ \mathsf{sk}, \ \mathsf{ct})\right\}_\lambda$$

By construction of DFE,

$$\mathsf{mpk} = (\mathsf{Dmpk}, \mathrm{crs}),\ \mathsf{sk} = \mathsf{Dsk}(f),\ \mathsf{ct} = \mathsf{Dct}(X),\ X = (\{\mathbf{x}^j\}, \mathrm{seed})\ ,$$

where every $\mathbf{x}^j$ is a random and independent sharing of v_b.

Consider the distribution $\{\mathcal{FN}\}$ of f, and the distribution $\mathrm{inp}(\mathcal{R})$ for sampling X, where inp is the identity function (and hence has locality 1) and $\mathcal{R}$ samples X honestly as in the $\mathsf{FE.Enc}$ algorithm. It follows from the special-purpose simulation security of DFE that there exist correlated random variables (X_K, K, st) sampled by $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$, and an efficient simulator DSim, such that, Real is $O(\mu)$-indistinguishable to the following $H_1 = \mathsf{Ideal}$ distribution.

Hybrid H_1: This hybrid samples view_A using the simulator of DFE as follows:

$$\mathsf{Ideal} = \left\{ \begin{array}{c} f \leftarrow \mathcal{FN}, \\ (X_K, K, \mathrm{st}) \leftarrow \mathsf{D}\mathcal{D}_{\mathsf{Sim}}, \\ \bar{X} \leftarrow \mathcal{R}|_{X_K, K}, \end{array} \quad : \quad \{\bar{\mathbf{x}}^j\},\ \mathsf{view}_A = \mathsf{DSim}\left((X_K, K, \mathrm{st}),\ f,\ y = f(\bar{X})\right) \right\}_\lambda\ ,$$

where

$$\bar{X} = (\{\bar{\mathbf{x}}^j\}, \overline{\mathrm{seed}}), \quad y_i^j = \mathsf{REnc}(D_i^j, \bar{x}_i^j\ ;\ \mathsf{PRG}_{\pi(j,i)}(\overline{\mathrm{seed}}))\ ,$$

where with probability $1 - O(\mu)$, $|K| \le O(\lambda^{\varepsilon_2})$.[5]

Fix any f and any (X_K, K, st) in the support of $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$. Since in $\mathcal{R}$, variables in seed and $\{\mathbf{x}^j\}$ are sampled independently and randomly, it is also the case in the conditional distribution. Formally,

$$\begin{aligned} &\mathcal{R}|_{X_K,K} = \left(\{\bar{\mathbf{x}}\} \leftarrow \mathcal{D}_{\mathbf{x}}|_{\mathbf{x}_{K_x}, K_x}\right) \ \times\ \left(\overline{\mathrm{seed}} \leftarrow \mathcal{U}|_{\mathrm{seed}_{K_{\mathrm{seed}}}, K_{\mathrm{seed}}}\right)\ , \\ &\mathcal{D}_{\mathbf{x}}\ :\ b \leftarrow \{0,1\}, (\mathbf{x} = \{\mathbf{x}^j \leftarrow \mathsf{BFshare}(\mathrm{crs}, v_b, 1^s, 1^{\mathrm{Dep}})\}\ , \quad \text{output } \mathbf{x}\ , \end{aligned}$$

where K_x and K_{seed} are the indexes of variables in $\{\mathbf{x}^j\}$ and seed that are fixed in K.

This means $\overline{\mathrm{seed}}$ is uniformly random at locations in $\neg K_{\mathrm{seed}}$, and fixed to $\mathrm{seed}_{K_{\mathrm{seed}}}$ at locations in K_{seed}. Recall that PRG is has constant locality ℓ and every output bit PRG_i is computed by $P(\mathrm{seed}_{G(i)})$ where G is the input-output dependency graph. Let K_{prg} be the set of indexes of PRG output bits that depends on seed bits in K_{seed}, that is,

$$K_{\mathrm{prg}} = G^{-1}(K_{\mathrm{seed}}) := \{l\ :\ G(l) \cap K_{\mathrm{seed}} \neq \emptyset\ \}$$

Since $\overline{\mathrm{seed}}_{\neg K_{\mathrm{seed}}}$ is uniformly random, we have by the security PRG that $\mathsf{PRG}_{\neg K_{\mathrm{prg}}}(\overline{\mathrm{seed}})$ is μ-indistinguishable to random. That is,

$$\mathsf{PRG}_{\neg K_{\mathrm{prg}}}(\overline{\mathrm{seed}}) \quad \approx_\mu \quad \mathcal{U}\ .$$

On the other hand, the distribution of output bits $\mathsf{PRG}_{K_{\mathrm{prg}}}(\overline{\mathrm{seed}})$ may not be random.

[5] Applying the special-purpose simulation security of DFE directly would imply the indistinguishability of Real and Ideal where the distributions output the entire X and $\bar{X}$. Since they contain $\{\mathbf{x}^j\}$ and $\{\bar{\mathbf{x}}^j\}$ respectively, H_0 and H_1 are also indistinguishable.

Furthermore, recall that the (i,j)'th randomized encoding y_i^j is computed using PRG output bits with indexes $\Pi_i^j = \{\pi(i,j,k)\}_{k\in[\phi]}$. Let $K_{\rm re}$ denote the set of (i,j) s.t. y_i^j depends on $\mathsf{PRG}_{K_{\rm prg}}(\overline{\rm seed})$, that is,

$$K_{\rm re} := \left\{(i,j) \ : \ \Pi_i^j \cap K_{\rm prg} \neq \emptyset \ \right\}$$

Observe that $K_{\rm re} = K_{\rm re}(K_{\rm seed})$ can be computed efficiently from $K_{\rm seed}$.

All randomized encodings y_i^j with $(i,j) \notin K_{\rm re}$ uses coins computed from truly random seed bits; call them the good encodings, and ones in $K_{\rm re}$ the bad encodings. By the security of RE, the good encodings are perfectly simulatable using the corresponding outputs:

$$\left\{ \ \left\{ y_i^j = \mathsf{REnc}(D_i^j, \bar{x}_i^j \ ; \ \mathsf{PRG}_{\pi(j,i)}(\overline{\rm seed})) \right\}_{(i,j)\in\neg K_{\rm re}} \ \right\}$$
$$\left\{ \ \left\{ \ \tilde{y}_i^j \leftarrow \mathsf{RSim}(\bar{o}_i^j = D_i^j(\bar{x}_i^j)) \right\}_{(i,j)\in\neg K_{\rm re}} \ \right\} .$$

where the output $\bar{o}_i^j$ is the output share obtained by homomorphically evaluating circuit C_{I_j} on input share $\bar{x}_i^j$. Therefore, H_1 is $O(\mu)$-indistinguishable to the following hybrid H_2.

Hybrid H_2: This hybrid samples view_A identically to H_1, except that randomized encodings at location $K_{\rm re}$ are now simulated.

$$\left\{ \begin{array}{c} f \leftarrow \mathcal{FN}, \\ (X_K, K, {\rm st}) \leftarrow \mathsf{D}\mathcal{D}_{\mathsf{Sim}}(f) \\ \bar{X} \leftarrow \mathcal{R}|_{X_K,K} \end{array} \quad : \right.$$
$$\left. \{\bar{\mathbf{x}}^j\}, \mathsf{view}_A = \mathsf{DSim}\left((X_K, K, {\rm st}), \ f, \ y' = \begin{cases} y_i^j & (i,j) \in K_{\rm re} \\ \tilde{y}_i^j & (i,j) \in \neg K_{\rm re} \end{cases} \right) \right\}_\lambda ,$$

where

$$\bar{X} = (\{\bar{\mathbf{x}}^j\}, \overline{\rm seed}), \qquad \begin{cases} y_i^j = \mathsf{REnc}(D_i^j, \bar{x}_i^j \ ; \ \mathsf{PRG}_{\pi(j,i)}(\overline{\rm seed})) \\ \tilde{y}_i^j \leftarrow \mathsf{RSim}(\bar{o}_i^j = D_i^j(\bar{x}_i^j)). \end{cases}$$

Next, we argue that bad encodings in $K_{\rm re}$ are well spread around among different chunks of randomized encodings $\mathbf{y}^j = \{y'^j_i\}_i$ that depend on the same sharing $\mathbf{x}^j = \{x_i^j\}$.

Lemma 5.5. *Let f and K be sampled from $\mathcal{FN}$ and $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}(f)$, and let $K_{\rm re} = K_{\rm re}(K_{\rm seed})$ as in H_2. It holds that there exists a constant c_0 such that for any $\mu = 2^{-o(\lambda^{\varepsilon_2})}$,*

$$\Pr\left[\ \exists j, \ \left|\{(i,j)\}_{i\in[\lambda]} \ \cap \ K_{\rm re}(K_{\rm seed})\right| \geq c_0 \cdot \lambda^{\varepsilon_2} \ \right] \leq O(\mu) \ .$$

Proof. Recall that (see Remark 5.3) $\mathcal{FN}$ is a special-purpose distribution — f sampled from $\mathcal{FN}$ has constant locality D, and depends on a fixed collection of predicates and a input-output dependency graph G_π from a distribution, where G_π is parameterized by a random permutation $\pi : [\lambda] \times [M] \times [\phi]$.

It follows from the property of $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}$ as shown in Lemma 4.6 that there exists a random variable $\mathcal{K}$ representing a subset of output bits of f, such that, *i)* $\mathcal{K}$ is correlated with

$(f[G_\pi], (r_K, K, \mathrm{st})$, *ii)* it is independent of G_π and hence π, and *iii)* the set of fixed seed bits satisfies $K_{\text{seed}} \subseteq G_\pi(\mathcal{K})$, and $|\mathcal{K}| = O(\lambda^{\varepsilon_2})$ with probability $1 - O(\mu)$. Therefore, it suffices to show that

$$\Pr\left[\ \exists j,\ \left|\{(i,j)\}_{i\in[\lambda]}\ \cap\ K_{\text{re}}(G_\pi(\mathcal{K}))\right| \geq c_0 \cdot \lambda^{\varepsilon_2}\ \right] \leq O(\mu)\ .$$

$J = G_\pi(\mathcal{K})$ is the collection of seed bits that output bits $\mathcal{K}$ in f depends on; by construction of f, $J = G(\pi(\mathcal{K}'))$ for some subset $\mathcal{K}'$ (determined by the dependency graph of REnc), with $|\mathcal{K}'| = O(|\mathcal{K}|)$. Recall that $K_{\text{re}}(J)$ includes all (i,j) s.t., encoding y_i^j depends on seed bits in J, that is, all (i,j), s.t. $G(\Pi_i^j = \{\pi(i,j,k)\}_{k\in[\phi]}) \cap J \neq \emptyset$. Let $G^{-1} \circ G(i)$ represent the set of two-hop neighbors of node i in graph G. The above inequality is equivalent to the following:

$$\Pr\left[\ \exists j,\ \left|\pi\left(\{(i,j,k)\}_{i\in[\lambda],k\in[\phi]}\right)\ \cap\ \left(G^{-1}\circ G\left(\pi(\mathcal{K}')\right)\right)\right| \geq c_0 \cdot \lambda^{\varepsilon_2}\ \right]$$
$$\leq 2^{-\Omega(\lambda^{\varepsilon_2})} + O(\mu) = O(\mu)\ .$$

where the last equality follows from $\mu = 2^{-o(\lambda^{\varepsilon_2})}$.

By our assumption on PRG, the input-output dependency graph G of PRG has $n = \mathrm{poly}(\lambda)S^{1-\alpha}$ input nodes, $m = \mathrm{poly}(\lambda)S$ output nodes, and every output nodes has constant degree ℓ. Furthermore, the degrees of the input nodes are bounded by $\ell' < o(S^{1-\alpha})$. The set $\{(i,j,k)\}_{i\in[\lambda],k\in[\phi]}$ has size $t = \mathrm{poly}(\lambda)S^\alpha$ and $\mathcal{K}'$ has size $s = O(\lambda^{\varepsilon_2})$ with probability $1 - O(\mu)$. Importantly, both sets are independent of the random permutation π. Thus, we can set the constant c_0 such that $c_0 \cdot \lambda^{\varepsilon_2} - s \geq \lambda^{\varepsilon_2}$. Furthermore, we have $s\ell\ell't \leq m - s - t$ for sufficiently large S (relative to λ) and $\alpha < 1/2$ with probability $1 - O(\mu)$. Hence, the inequality follows immediately from the following claim.

Claim 1. *Let G be a bipartite graph with n input nodes and m output nodes, where the degrees of the input and output nodes are bounded by ℓ' and ℓ, respectively. Let $J, \mathcal{J} \subseteq [m]$ be sets of size t and s, respectively, such that $s \leq t$ and $s\ell\ell't \leq m - s - t$, and let $B \geq s + 2$. We then have over the choice of a random permutation π over the output nodes (i.e., $\pi\colon [m] \to [m]$),*

$$\Pr_\pi\left[\left|\pi(J)\ \cap\ \left(G^{-1}\circ G\left(\pi(\mathcal{J})\right)\right)\right| \geq B\ \right] \leq \exp\left(-\frac{B-s-1}{3}\right).$$

Proof. Since $\pi(\mathcal{J}) \subseteq G^{-1} \circ G\left(\pi(\mathcal{J})\right)$, we have $\pi(J \cap \mathcal{J}) \subseteq \pi(J) \cap \left(G^{-1}\circ G\left(\pi(\mathcal{J})\right)\right)$ for all π. We therefore need to bound the probability of

$$\left|\pi(J\setminus\mathcal{J})\ \cap\ \left(G^{-1}\circ G\left(\pi(\mathcal{J})\right)\right)\right| \geq B - |J \cap \mathcal{J}|.$$

The random experiment can equivalently be described as follows: First, assign to each $j \in \mathcal{J}$ a random, distinct $j' \in [m]$. Let $\mathcal{J}'$ be the set of all these j', and let $\hat{\mathcal{J}} := G^{-1} \circ G(\mathcal{J}')$. Then, assign to each $v \in J \setminus \mathcal{J}$ a random, distinct $v' \in [m] \setminus \mathcal{J}'$. Since $\left|\hat{\mathcal{J}}\right| \leq s\ell\ell'$, the probability that the first of these v' is in $\hat{\mathcal{J}}$ is at most $\frac{s\ell\ell'}{m-s}$. The second v' is then chosen from a set of size $m - s - 1$ since it must be different from the first v'. Hence, the probability that it is in $\hat{\mathcal{J}}$ is at most $\frac{s\ell\ell'}{m-s-1}$. Using $|J \setminus \mathcal{J}| \leq t$, this implies that the probability of any v' landing in $\hat{\mathcal{J}}$ is at most $\frac{s\ell\ell'}{m-s-t}$. While the assignments of the v' are not independent (because they must be distinct), whether they are in $\hat{\mathcal{J}}$ or not only depends on previous outcomes in the sense that the probability gets smaller if previous v' are in $\hat{\mathcal{J}}$ (since less elements in $\hat{\mathcal{J}}$ are available). One can therefore define independent random variables $X_1, \ldots, X_{|J \setminus \mathcal{J}|}$, where each X_i is 1 with probability $\frac{s\ell\ell'}{m-s-t}$, and 0 otherwise, such that the probability of $X_i = 1$ upper bounds the probability of the ith v' being in $\hat{\mathcal{J}}$. This shows that

$$\Pr_{\pi}\left[\left|\pi(J \setminus \mathcal{J}) \ \cap \ \left(G^{-1} \circ G\left(\pi(\mathcal{J})\right)\right)\right| \geq B - |J \cap \mathcal{J}|\right]$$
$$\leq \Pr\bigl[X_1 + \ldots + X_{|J \setminus \mathcal{J}|} \geq B - |J \cap \mathcal{J}|\bigr] \leq \Pr\bigl[X_1 + \ldots + X_{|J \setminus \mathcal{J}|} \geq B - s\bigr].$$

Note that the expected value of $X_1 + \ldots + X_{|J \setminus \mathcal{J}|}$ is $\nu := \frac{s\ell\ell'}{m-s-t} \cdot |J \setminus \mathcal{J}| \leq \frac{s\ell\ell' t}{m-s-t} \leq 1$. Let $\delta := \frac{B-s}{\nu} - 1$. Since $\nu \leq 1$ and $B \geq s+2$, we have $\delta \geq 1$. Hence, the Chernoff bound implies

$$\Pr\bigl[X_1 + \ldots + X_{|J \setminus \mathcal{J}|} \geq B - s\bigr] = \Pr\bigl[X_1 + \ldots + X_{|J \setminus \mathcal{J}|} \geq (1+\delta)\nu\bigr] \leq \exp(-\delta\nu/3).$$

Again using $\nu \leq 1$, we obtain $\delta\nu = B - s - \nu \geq B - s - 1$, which concludes the proof of the claim. □

This also concludes the proof of Lemma 5.5. □

We are now ready to argue that for any PPT adversary A, over the choice of $(\{\bar{\mathbf{x}}^j\}, \mathsf{view}_A)$ from H_2, the probability that A seeing view_A can predict which input v_b is shared in $\{\bar{\mathbf{x}}^j\}$ is at most $O(\mu)$.

Claim 2. *For every PPT adversary A,*

$$\Pr\left[(\{\bar{\mathbf{x}}^j\}, \mathsf{view}_A) \leftarrow H_2 \ : \ b \leftarrow A(\mathsf{view}) \ \wedge \ \bar{\mathbf{x}}^1 \textit{ shares } v_b\right] \leq 1/2 + O(\mu)$$

Proof. Fix any f and any (X_K, K, st) in the support of $\mathsf{D}\mathcal{D}_{\mathsf{Sim}}(f)$ such that $|K| = O(\lambda^{\varepsilon_2})$ and Lemma 5.5 holds. Such f and (X_K, K, st) account for an $1 - O(\mu)$ fraction.

Conditioned on them being sampled, H_2 further samples $\mathcal{R}|_{X_K, K}$, which samples M sharing $\{\bar{\mathbf{x}}^j\}$ of v_b for a randomly chosen bit b, conditioned on a few $O(\lambda^{\varepsilon_2})$ fixed bits contained in X_K. Observe that the inputs to DSim depend only on output shares $\{\bar{o}_i^j\}_{(i,j) \notin K_{\mathrm{re}}}$ and input shares $\{\bar{x}_i^j\}_{(i,j) \in K_{\mathrm{re}}}$ (all other information is independent of $\bar{\mathbf{x}}$). Since $|K| = O(\lambda^{\varepsilon_2})$ and Lemma 5.5 holds, for every sharing $\bar{\mathbf{x}}^j$, at most $O(\lambda^{\varepsilon_2})$ bits of it are fixed, and at most $O(\lambda^{\varepsilon_2})$ shares $\bar{x}_i^j$ are revealed to DSim. Therefore, it follows from the $(O(\lambda^{\varepsilon_2}), \lambda^{\omega(1)}\mu)$-bit-fixing security of BF (and the fact that there are at most a polynomial number of sharing of BF) that, the probability that A receiving the output of DSim can predict b is at most $1/2 + \mu$. This conclude the claim. □

Finally, it follows from the fact that H_0 and H_2 are $O(\mu)$-indistinguishable, we conclude that A receiving the real view as generated in $H_0 = \mathsf{Real}$ can only predict b with probability $1/2 + O(\mu)$.

$$\Pr\left[(\{\mathbf{x}^j\}, \mathsf{view}_A) \leftarrow \mathsf{Real} \ : \ b \leftarrow A(\mathsf{view}) \ \wedge \ \mathbf{x}^1 \text{ shares } v_b\right] \leq 1/2 + O(\mu)$$

This concludes the proof of the theorem. □

6 Construction of Candidate PFG

6.1 Properties of (Flawed-)Smudging Distributions

6.1.1 Preservation Under Addition of Independent Value

We show that if the distribution of X is (flawed-)smudging and Y is independent from X, then the distribution of $X+Y$ is (flawed-)smudging.

Lemma 6.1. *Let ℓ and B be positive integers, let $\varepsilon \in [0,1]$, and let X be a random variable over $\mathbb{Z}^\ell$ with (B,ε)-smudging distribution. Further let Y be a random variable over $\mathbb{Z}^\ell$ such that X and Y are independent. Then, the distribution of $X+Y$ is (B,ε)-smudging.*

Proof. Since adding an independent random variable cannot increase the statistical distance, we have for all $e \in [-B,B]^\ell \cap \mathbb{Z}^\ell$,

$$\delta(X+Y, X+Y+e) \leq \delta(X, X+e) \leq \varepsilon.$$

Hence, the distribution of $X+Y$ is (B,ε)-smudging. □

Lemma 6.2. *Let ℓ and m be positive integers and let $\mathcal{X}$ and $\mathcal{V}$ be distributions over $\mathbb{Z}^\ell$ and $\mathbb{Z}^m$, respectively. Further let for $i \in [\ell]$, $\Phi_i \subseteq [m]$, let $E_i\colon \mathbb{Z}^{|\phi_i|} \to \mathbb{Z}$ be functions, and let E be as in Definition 3.4. Further let X be a random variable with distribution $\mathcal{X}$ and let Y be an independent random variable. Assume that $\mathcal{X}$ is (K,μ)-flawed-smudging for $(E,\mathcal{V})$. Then, the distribution of $X+Y$ is (K,μ)-flawed-smudging for $(E,\mathcal{V})$.*

Proof. By assumption, there are randomized predicates $\left\{\mathrm{BAD}_i : \mathbb{Z}^{\ell+1} \to \{0,1\}\right\}_{i\in[\ell]}$ such that $\mathcal{D}_1$ and $\mathcal{D}_2$ as defined in Definition 3.4 are identical and $|\mathrm{bad}|_1 \leq K$ with probability at least $1-\mu$. Now define BAD'_i for the distribution of $X+Y$ as follows: Given $E_i(V_{\Phi_i})$ and $X+Y$, sample X' and Y' with marginal distributions equal to the distributions of X and Y, respectively, conditioned on $X'+Y' = X+Y$, and return $\mathrm{BAD}_i(E_i(V_{\Phi_i}), X')$. Let $\mathrm{bad}' = \left(\mathrm{bad}'_i \leftarrow \mathrm{BAD}'_i(E_i(V_{\Phi_i}), X+Y)\right)_{i\in[\ell]}$ and let $\overline{V}' \leftarrow \mathcal{V}|_{V_{\Phi_{\mathrm{bad}'}}, \Phi_{\mathrm{bad}'}}$. Since X and X' have the same distribution and $\mathcal{D}_1$ and $\mathcal{D}_2$ are equal, we also have that $\left(V, E(V)+X', \mathrm{bad}'\right)$ and $\left(\overline{V}', E(V)+X', \mathrm{bad}'\right)$ have the same distribution. Note that Y' is independent of $\left(V, E(V)+X', \mathrm{bad}'\right)$ and $\left(\overline{V}', E(V)+X', \mathrm{bad}'\right)$, since it only depends on X' given $X+Y$, and nothing depends on $X+Y$ beyond depending on X'. Hence, $\left(V, E(V)+X'+Y', \mathrm{bad}'\right)$ and $\left(\overline{V}', E(V)+X'+Y', \mathrm{bad}'\right)$ also have the same distribution. Using $X'+Y' = X+Y$, we obtain that $\mathcal{D}'_1$ and $\mathcal{D}'_2$ defined as in Definition 3.4 for $X+Y$ and BAD' are equal. Furthermore, bad and bad' are equally distributed, which implies that $|\mathrm{bad}'|_1 \leq K$ with probability at least $1-\mu$. Thus, the distribution of $X+Y$ is (K,μ)-flawed-smudging for $(E,\mathcal{V})$. □

6.1.2 Mixtures of (Flawed-)Smudging Distributions

We next show that the probabilistic mixture of several (flawed-)smudging distributions is also (flawed-)smudging.

Lemma 6.3. *Let ℓ, B, and k be positive integers, and let for $i \in [k]$, $\varepsilon_i \in [0,1]$, and let X_i be a random variable over $\mathbb{Z}^\ell$ with (B,ε_i)-smudging distribution. Further let $\alpha_1,\ldots,\alpha_k \in [0,1]$ such that $\sum_{i=1}^k \alpha_i = 1$, and define the random variable X with $\Pr[X=x] = \sum_{i=1}^k \alpha_i \Pr[X_i = x]$. Then, the distribution of X is $\left(B, \sum_{i=1}^k \alpha_i\varepsilon_i\right)$-smudging.*

Proof. Let $e \in [-B, B]^\ell \cap \mathbb{Z}^\ell$. We then have

$$\begin{aligned}\delta(X, X+e) &= \frac{1}{2} \sum_{x \in \mathbb{Z}^\ell} \bigl|\Pr[X = x] - \Pr[X + e = x]\bigr| \\ &\le \frac{1}{2} \sum_{x \in \mathbb{Z}^\ell} \sum_{i=1}^{k} \alpha_i \cdot \bigl|\Pr[X_i = x] - \Pr[X_i + e = x]\bigr| \\ &= \sum_{i=1}^{k} \alpha_i \cdot \delta(X_i, X_i + e).\end{aligned}$$

Since $\delta(X_i, X_i + e) \le \varepsilon_i$, the claim follows. □

Lemma 6.4. *Let ℓ, m, and k be positive integers, let $\mathcal{V}$ be a distribution over $\mathbb{Z}^m$, and let E be as in Definition 3.4. Let for $i \in [k]$, $\mu_i \in [0,1]$, and let X_i be a random variable over $\mathbb{Z}^\ell$ with (K, μ_i)-flawed-smudging distribution for $(E, \mathcal{V})$. Further let $\alpha_1, \ldots, \alpha_k \in [0,1]$ such that $\sum_{i=1}^k \alpha_i = 1$, and define the random variable X with $\Pr[X = x] = \sum_{i=1}^k \alpha_i \Pr[X_i = x]$. Then, the distribution of X is $\bigl(K, \sum_{i=1}^k \alpha_i \mu_i\bigr)$-flawed-smudging for $(E, \mathcal{V})$.*

Proof. By assumption, there are randomized predicates $\bigl\{\mathrm{BAD}_j^{(i)} : \mathbb{Z}^{\ell+1} \to \{0,1\}\bigr\}_{j \in [\ell]}$ for $i \in [k]$ such that $\mathcal{D}_1^{(i)}$ and $\mathcal{D}_2^{(i)}$ as defined in Definition 3.4 for X_i are identical and $|\mathrm{bad}^{(i)}|_1 \le K$ with probability at least $1 - \mu_i$. Now define BAD'_j for the distribution of X as follows: Given $E_j(V_{\Phi_j})$ and X, sample $(X'_1, \ldots, X'_k, A)$ conditioned on $X'_A = X$, where X'_i is distributed as X_i for $i \in [k]$ and $\Pr[A = i] = \alpha_i$. Then output $\mathrm{BAD}_j^{(A)}(E_j(V_{\Phi_j}), X'_A)$. Let $\mathrm{bad}' = \bigl(\mathrm{bad}'_j \leftarrow \mathrm{BAD}'_j(E_j(V_{\Phi_j}), X)\bigr)_{j \in [\ell]}$ and let $\overline{V}' \leftarrow \mathcal{V}|_{V_{\Phi_{\mathrm{bad}'}}, \Phi_{\mathrm{bad}'}}$ and $\overline{V}^{(i)} \leftarrow \mathcal{V}|_{V_{\Phi_{\mathrm{bad}^{(i)}}}, \Phi_{\mathrm{bad}^{(i)}}}$.

We have for all t

$$\begin{aligned}\Pr\bigl[\bigl(\overline{V}', E(V) + X'_A, \mathrm{bad}'\bigr) = t\bigr] &= \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr\bigl[\bigl(\overline{V}', E(V) + X'_A, \mathrm{bad}'\bigr) = t \bigm| A = i\bigr] \\ &= \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr\bigl[\bigl(\overline{V}^{(i)}, E(V) + X_i, \mathrm{bad}^{(i)}\bigr) = t\bigr] \\ &= \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr\bigl[\bigl(V, E(V) + X_i, \mathrm{bad}^{(i)}\bigr) = t\bigr] \\ &= \Pr\bigl[\bigl(V, E(V) + X, \mathrm{bad}'\bigr) = t\bigr].\end{aligned}$$

This established the desired equality of distributions.

Using that the distribution of bad' conditioned on $A = i$ equals the distribution of $\mathrm{bad}^{(i)}$, we obtain

$$\Pr\bigl[|\mathrm{bad}'|_1 \le K\bigr] = \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr\bigl[|\mathrm{bad}^{(i)}|_1 \le K\bigr] \ge \sum_{i=1}^{k} \alpha_i \cdot (1 - \mu_i) = 1 - \sum_{i=1}^{k} \alpha_i \cdot \mu_i.$$ □

6.1.3 Smudging and Independence Implies Flawed-Smudging

We want to show that if several mutually independent random variables each have a smudging distribution, then their joint distribution is flawed-smudging. To this end, we first prove a technical lemma. Our starting point is the known fact that for two random variables X and X', one can define events BAD and BAD$'$ such that the probability of these events equals $\delta(X, X')$ and the distribution of X conditioned on $\neg$BAD is equal to the distribution of X' conditioned on $\neg$BAD$'$ [MPR07]. We are interested in the following related statement, for which the bad event can be defined similarly: Assume X and E are random variables over $\mathbb{Z}$ such that E is B-bounded and $\delta(X, X + e)$ is "small" for all e in the support of E. Then, there exists an event BAD with "small" probability such that if BAD does not occur, then E does not depend on $X + E$ and the fact that BAD does not occur.

The high-level idea is to define the event BAD such that for all $y \in \mathbb{Z}$, $\Pr[X + E = y \wedge \neg\mathrm{BAD}] = \min_e\{\Pr[X + e = y]\}$. This means that given $X + E = y$, the probability of BAD gets larger the greater the gap between the probabilities of $X + e = y$ for different values of e is.

The following lemma generalizes this in two ways: First, we consider several X_i and E_i. Secondly, the E_i are functions of several V_j, which can be correlated. We then define events BAD_i for each $X_i + E_i$ and get the statement that intuitively says: If V_j is not used in the computation of any E_i for which BAD_i occurs, then V_j does not depend on any $X_i + E_i$.

Lemma 6.5. *Let $m, \ell \in \mathbb{N}$ and let $X_1, \dots, X_\ell$, $V_1, \dots, V_m$ be random variables over $\mathbb{Z}$ such that the $\ell+1$ random variables $X_1, \dots, X_\ell$, and $(V_1, \dots, V_m)$ are mutually independent.*[6] *Further let for $i \in [\ell]$, $\Phi_i \subseteq [m]$, let $E_i\colon \mathbb{Z}^{|\Phi_i|} \to \mathbb{Z}$ be functions, let $F_i := E_i\big((V_j)_{j\in\Phi_i}\big)$, and let $\mathcal{F}_i$ be the support of F_i with $|\mathcal{F}_i| < \infty$. For $I \subseteq [\ell]$, let $\Phi_I := \bigcup_{i\in I} \Phi_i$. Then, there exist events $\mathrm{BAD}_1, \dots, \mathrm{BAD}_\ell$ with $\mathrm{BAD}_i = \mathrm{BAD}_i(F_i, X_i)$ such that for all $y_1, \dots, y_\ell \in \mathbb{Z}$, $v_1, \dots, v_m \in \mathbb{Z}$, and for all $I \subseteq [\ell]$,*

(i) $\Pr\big[\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \mid \bigwedge_{i\in[\ell]} X_i + F_i = y_i \wedge \bigwedge_{i\in\Phi_I} V_i = v_i \wedge \bigwedge_{i\in I} \mathrm{BAD}_i \wedge \bigwedge_{i\in[\ell]\setminus I} \neg\mathrm{BAD}_i\big] = \Pr\big[\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \mid \bigwedge_{i\in\Phi_I} V_i = v_i\big]$*,*

(ii) $\Pr\big[\bigwedge_{i\in I} \mathrm{BAD}_i\big] \le \prod_{i\in I} 4 \cdot \sum_{f_i\in\mathcal{F}_i} \delta(X_i, X_i + f_i)$*.*

Proof. We define BAD_i for $F_i = f_i$ and $X_i = x_i$ as

$$\Pr[\mathrm{BAD}_i(f_i, x_i) = 1] := 1 - \min_{f'_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f'_i = x_i + f_i)}{\Pr(X_i = x_i)}\right\}.$$

Note that this is a valid probability since $\min_{f'_i\in\mathcal{F}_i}\{\Pr(X_i + f'_i = x_i + f_i)\} \le \Pr(X_i = x_i)$. We then have for $y_1, \dots, y_\ell \in \mathbb{Z}$ and $v_1, \dots, v_m \in \mathbb{Z}$ with $\Pr\big[\bigwedge_{j\in[\ell]} X_j + F_j = y_j \wedge \bigwedge_{j\in[m]} V_j = v_j\big] > 0$,

$$\Pr\left[\mathrm{BAD}_i \,\middle|\, \bigwedge_{j\in[\ell]} X_j + F_j = y_j \wedge \bigwedge_{j\in[m]} V_j = v_j\right] = 1 - \min_{f_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i\big((v_j)_{j\in\Phi_i}\big) = y_i)}\right\},$$

where the BAD_i are sampled independently given $\bigwedge_{j\in[\ell]} X_j + F_j = y_j \wedge \bigwedge_{j\in[m]} V_j = v_j$.

Now fix $y_1, \dots, y_\ell \in \mathbb{Z}$, $v_1, \dots v_m \in \mathbb{Z}$, and $I \subseteq [\ell]$. We then have using the independence of

[6] But, $V_1, \dots, V_m$ need not be independent.

the X_i and the conditional independence of the BAD_i,

$$\begin{aligned}
&\Pr\bigl[\textstyle\bigwedge_{i\in[\ell]} X_i + F_i = y_i \wedge \bigwedge_{i\in[m]} V_i = v_i \wedge \bigwedge_{i\in I} \mathrm{BAD}_i \wedge \bigwedge_{i\in[\ell]\setminus I} \neg\mathrm{BAD}_i\bigr] \\
&\quad = \Pr\Biggl[\bigwedge_{i\in[m]} V_i = v_i\Biggr] \cdot \prod_{i\in[\ell]} \Pr[X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i] \cdot \prod_{i\in[\ell]\setminus I} \min_{f_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i)}\right\} \\
&\qquad\qquad \cdot \prod_{i\in I}\left(1 - \min_{f_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i)}\right\}\right) \\
&\quad = \Pr\Biggl[\bigwedge_{i\in[m]} V_i = v_i\Biggr] \cdot \prod_{i\in I} \Pr[X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i] \cdot \prod_{i\in[\ell]\setminus I} \min_{f_i\in\mathcal{F}_i}\{\Pr(X_i + f_i = y_i)\} \\
&\qquad\qquad \cdot \prod_{i\in I}\left(1 - \min_{f_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i)}\right\}\right).
\end{aligned} \tag{5}$$

Note that by definition of Φ_I, $E_i\bigl((v_j)_{j\in\Phi_i}\bigr)$ for $i \in I$ does not depend on v_j for $j \in [m] \setminus \Phi_I$. Hence,

$$\begin{aligned}
&\Pr\Biggl[\bigwedge_{i\in[\ell]} X_i + F_i = y_i \wedge \bigwedge_{i\in\Phi_I} V_i = v_i \wedge \bigwedge_{i\in I} \mathrm{BAD}_i \wedge \bigwedge_{i\in[\ell]\setminus I} \neg\mathrm{BAD}_i\Biggr] \\
&\quad = \sum_{\substack{v_i\in\mathbb{Z} \\ i\in[m]\setminus\Phi_I}} \Pr\Biggl[\bigwedge_{i\in[\ell]} X_i + F_i = y_i \wedge \bigwedge_{i\in[m]} V_i = v_i \wedge \bigwedge_{i\in I} \mathrm{BAD}_i \wedge \bigwedge_{i\in[\ell]\setminus I} \neg\mathrm{BAD}_i\Biggr] \\
&\quad = \Pr\Biggl[\bigwedge_{i\in\Phi_I} V_i = v_i\Biggr] \cdot \prod_{i\in I} \Pr[X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i] \cdot \prod_{i\in[\ell]\setminus I} \min_{f_i\in\mathcal{F}_i}\{\Pr(X_i + f_i = y_i)\} \\
&\qquad\qquad \cdot \prod_{i\in I}\left(1 - \min_{f_i\in\mathcal{F}_i}\left\{\frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i\bigl((v_j)_{j\in\Phi_i}\bigr) = y_i)}\right\}\right).
\end{aligned} \tag{6}$$

Dividing equation (5) by equation (6) yields

$$\begin{aligned}
&\Pr\bigl[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \bigm| \bigwedge_{i\in[\ell]} X_i + F_i = y_i \wedge \bigwedge_{i\in\Phi_I} V_i = v_i \wedge \bigwedge_{i\in I} \mathrm{BAD}_i \wedge \bigwedge_{i\in[\ell]\setminus I} \neg\mathrm{BAD}_i\bigr] \\
&\qquad = \Pr\bigl[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \bigm| \bigwedge_{i\in\Phi_I} V_i = v_i\bigr],
\end{aligned}$$

and concludes the proof of the first claim of the lemma.

To prove (ii), note that $\bigwedge_{i\in I} \mathrm{BAD}_i$ only depends on X_i for $i \in I$ and V_j for $j \in \Phi_I$. Together with the independence of the X_i, this yields that $\Pr\big[\bigwedge_{i\in I}\mathrm{BAD}_i\big]$ equals

$$\begin{aligned}
&\sum_{\substack{y_i\in\mathbb{Z}\\ i\in I}}\sum_{\substack{v_i\in\mathbb{Z}\\ i\in\Phi_I}}\Pr\Bigg[\bigwedge_{i\in I}X_i+F_i=y_i\wedge\bigwedge_{i\in\Phi_I}V_i=v_i\Bigg]\cdot\Pr\Bigg[\bigwedge_{i\in I}\mathrm{BAD}_i\,\Bigg|\,\bigwedge_{i\in I}X_i+F_i=y_i\wedge\bigwedge_{i\in\Phi_I}V_i=v_i\Bigg]\\
&=\sum_{\substack{y_i\in\mathbb{Z}\\ i\in I}}\sum_{\substack{v_i\in\mathbb{Z}\\ i\in\Phi_I}}\Pr\Bigg[\bigwedge_{i\in I}X_i+F_i=y_i\wedge\bigwedge_{i\in\Phi_I}V_i=v_i\Bigg]\cdot\prod_{i\in I}\Bigg(1-\min_{f_i\in\mathcal{F}_i}\Bigg\{\frac{\Pr(X_i+f_i=y_i)}{\Pr(X_i+E_i\big((v_j)_{j\in\Phi_i}\big)=y_i)}\Bigg\}\Bigg)\\
&=\sum_{\substack{y_i\in\mathbb{Z}\\ i\in I}}\sum_{\substack{v_i\in\mathbb{Z}\\ i\in\Phi_I}}\Pr\Bigg[\bigwedge_{i\in\Phi_I}V_i=v_i\Bigg]\cdot\prod_{i\in I}\Bigg(\Pr(X_i+E_i\big((v_j)_{j\in\Phi_i}\big)=y_i)-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\Bigg)\\
&\le\sum_{\substack{y_i\in\mathbb{Z}\\ i\in I}}\sum_{\substack{v_i\in\mathbb{Z}\\ i\in\Phi_I}}\Pr\Bigg[\bigwedge_{i\in\Phi_I}V_i=v_i\Bigg]\cdot\prod_{i\in I}\Bigg(\max_{f_i\in\mathcal{F}_i}\Pr(X_i+f_i=y_i)-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\Bigg).
\end{aligned}$$

Since the term in the product over $i\in I$ does not depend on v_i, we obtain

$$\begin{aligned}
\Pr\Bigg[\bigwedge_{i\in I}\mathrm{BAD}_i\Bigg]&\le\sum_{\substack{y_i\in\mathbb{Z}\\ i\in I}}\prod_{i\in I}\Bigg(\max_{f_i\in\mathcal{F}_i}\Pr(X_i+f_i=y_i)-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\Bigg)\\
&=\prod_{i\in I}\sum_{y_i\in\mathbb{Z}}\Bigg(\max_{f_i\in\mathcal{F}_i}\Pr(X_i+f_i=y_i)-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\Bigg)\\
&\le\prod_{i\in I}\sum_{y_i\in\mathbb{Z}}\Bigg(\bigg|\max_{f_i\in\mathcal{F}_i}\Pr(X_i+f_i=y_i)-\Pr[X_i=y_i]\bigg|\\
&\qquad\qquad+\bigg|\Pr[X_i=y_i]-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\bigg|\Bigg).
\end{aligned}$$

Since $\big|\max_{f_i\in\mathcal{F}_i}\Pr(X_i+f_i=y_i)-\Pr[X_i=y_i]\big|$ and $\big|\Pr[X_i=y_i]-\min_{f_i\in\mathcal{F}_i}\big\{\Pr(X_i+f_i=y_i)\big\}\big|$ are both upper bounded by $\sum_{f_i\in\mathcal{F}_i}|\Pr[X_i=y_i]-\Pr(X_i+f_i=y_i)|$, we have

$$\begin{aligned}
\Pr\Bigg[\bigwedge_{i\in I}\mathrm{BAD}_i\Bigg]&\le\prod_{i\in I}2\cdot\sum_{f_i\in\mathcal{F}_i,y_i\in\mathbb{Z}}|\Pr[X_i=y_i]-\Pr(X_i+f_i=y_i)|\\
&=\prod_{i\in I}4\cdot\sum_{f_i\in\mathcal{F}_i}\delta(X_i,X_i+f_i).
\end{aligned}$$
□

Using this lemma, we can now show the following result.

Proposition 6.6. *Let ℓ and K be positive integers and let $\mathcal{X}$ be a distribution over $\mathbb{Z}^\ell$ such that for $(X_1,\dots,X_\ell)\xleftarrow{\$}\mathcal{X}$, $X_1,\dots,X_\ell$ are mutually independent and the distribution of each X_i is (B,ε)-smudging for $\varepsilon\le\frac{K+1}{22\ell\cdot(2B+1)}$. Further let $\mu=2\big(\frac{11\ell\cdot(2B+1)\varepsilon}{K+1}\big)^{K+1}$. Then, $\mathcal{X}$ is (K,μ)-flawed-smudging for B-bounded distributions.*

Proof. Let m be a positive integer and let for $i\in[\ell]$, $\Phi_i\subseteq[m]$, and let $E_i\colon\mathbb{Z}^{|\Phi_i|}\to\mathbb{Z}$ be functions. For $I\subseteq[\ell]$, let $\Phi_I:=\bigcup_{i\in I}\Phi_i$ and let $E\colon\mathbb{Z}^m\to\mathbb{Z}^\ell$ be as in Definition 3.4. Let $\mathcal{V}$ be a distribution over $\mathbb{Z}^m$ such that $E(\mathcal{V})$ is B-bounded. Further let BAD be the events guaranteed by Lemma 6.5, let V, X, and $\overline{V}$ be random variables sampled as in Definition 3.4, and let $\mathcal{D}_1$, $\mathcal{D}_2$ be the distributions from Definition 3.4. Finally let $I \subseteq [\ell]$, $y \in \mathbb{Z}^\ell$, and $v \in \mathbb{Z}^m$. Then,

$$\begin{aligned}
&\Pr[V = v \wedge E(V) + X = y \wedge \mathrm{BAD} = I] \\
&= \Pr\big[\textstyle\bigwedge_{i\in\Phi_I} V_i = v_i \wedge E(V) + X = y \wedge \mathrm{BAD} = I\big] \\
&\qquad\qquad \cdot \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i \wedge E(V) + X = y \wedge \mathrm{BAD} = I\big] \\
&= \Pr\big[\textstyle\bigwedge_{i\in\Phi_I} V_i = v_i \wedge E(V) + X = y \wedge \mathrm{BAD} = I\big] \cdot \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i\big],
\end{aligned} \tag{7}$$

where the last equality follows from Lemma 6.5. Now let $Z \coloneqq \bigwedge_{i\in\Phi_I} V_i = v_i \wedge \mathrm{BAD} = I$. Then,

$$\begin{aligned}
&\Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \wedge E(V) + X = y \wedge Z\big] \\
&\qquad = \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \;\big|\; E(V) + X = y \wedge Z\big] \cdot \Pr\big[E(V) + X = y \wedge Z\big].
\end{aligned}$$

Note that we have by definition of $\overline{V}$ that

$$\Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \;\big|\; E(V) + X = y \wedge Z\big] = \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i\big].$$

Hence,

$$\begin{aligned}
&\Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \wedge E(V) + X = y \wedge Z\big] \\
&\qquad = \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i\big] \cdot \Pr\big[E(V) + X = y \wedge Z\big].
\end{aligned}$$

Dividing this equation by $\Pr[Z]$ yields

$$\begin{aligned}
&\Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \wedge E(V) + X = y \;\big|\; Z\big] \\
&\quad = \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i\big] \cdot \Pr\big[E(V) + X = y \;\big|\; Z\big].
\end{aligned} \tag{8}$$

We finally obtain

$$\begin{aligned}
&\Pr[V = v \wedge E(V) + X = y \wedge \mathrm{BAD} = I] \\
&\overset{(7)}{=} \Pr[Z] \cdot \Pr\big[E(V) + X = y \;\big|\; Z\big] \cdot \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} V_i = v_i \;\big|\; \bigwedge_{i\in\Phi_I} V_i = v_i\big] \\
&\overset{(8)}{=} \Pr[Z] \cdot \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \wedge E(V) + X = y \;\big|\; Z\big] \\
&= \Pr\big[\textstyle\bigwedge_{i\in[m]\setminus\Phi_I} \overline{V}_i = v_i \wedge \bigwedge_{i\in\Phi_I} V_i = v_i \wedge E(V) + X = y \wedge \mathrm{BAD} = I\big].
\end{aligned}$$

Since $\overline{V}_i = V_i$ for $i \in \Phi_I$ if $\mathrm{BAD} = I$, this implies that the distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are equal.

By Lemma 6.5 (ii), we have for all $I \subseteq [\ell]$, $\Pr\big[\bigwedge_{i\in I} \mathrm{BAD}_i\big] \leq \prod_{i\in I} 4 \cdot \sum_{f_i\in\mathcal{F}_i} \delta(X_i, X_i + f_i)$. By our assumptions on $\mathcal{X}$ and $E(\mathcal{V})$, we have $\delta(X_i, X_i + f_i) \leq \varepsilon$ for all $f_i \in \mathcal{F}_i$ and $|\mathcal{F}_i| \leq 2B+1$. Thus, $\Pr\big[\bigwedge_{i\in I} \mathrm{BAD}_i\big] \leq (4(2B+1)\varepsilon)^{|I|}$. We therefore have

$$\begin{aligned}
\Pr[|\mathrm{BAD}|_1 > K] &= \sum_{k=K+1}^{\ell} \Pr[|\mathrm{BAD}|_1 = k] \\
&\leq \sum_{k=K+1}^{\ell} \sum_{\substack{I\subseteq[\ell]\\|I|=k}} \Pr\left[\bigwedge_{i\in I} \mathrm{BAD}_i\right] \leq \sum_{k=K+1}^{\ell} \sum_{\substack{I\subseteq[\ell]\\|I|=k}} (4(2B+1)\varepsilon)^k.
\end{aligned}$$

There are $\binom{\ell}{k}$ subsets $I \subseteq [\ell]$ of size k. Using the bound $\binom{\ell}{k} \leq \left(\frac{\ell e}{k}\right)^k$, where e is Euler's constant, we obtain

$$\Pr[|\mathrm{BAD}|_1 > K] \leq \sum_{k=K+1}^{\ell} \left(\frac{\ell e \cdot 4(2B+1)\varepsilon}{k}\right)^k < \sum_{k=K+1}^{\ell} \left(\frac{11\ell \cdot (2B+1)\varepsilon}{K+1}\right)^k.$$

Let $\alpha \coloneqq \frac{11\ell \cdot (2B+1)\varepsilon}{K+1}$. Since $\varepsilon \leq \frac{K+1}{22\ell \cdot (2B+1)}$ by assumption, we have $\alpha \leq 1/2$. Using the formula $\sum_{k=0}^{n} \alpha^k = \frac{1-\alpha^n}{1-\alpha}$ for the first n terms of the geometric series, we then have

$$\Pr[|\mathrm{BAD}|_1 > K] < \sum_{k=0}^{\ell} \alpha^k - \sum_{k=0}^{K} \alpha^k = \frac{\alpha^{K+1} - \alpha^{\ell+1}}{1-\alpha} \leq 2\alpha^{K+1}.$$

Hence, the probability that $|\mathrm{BAD}|_1 \leq K$ is at least $1 - 2\alpha^{K+1} = 1 - \mu$. Altogether, we conclude that $\mathcal{X}$ is (K, μ)-flawed-smudging for B-bounded distributions. □

6.2 Candidate Construction

6.2.1 Definitions

We next provide a candidate PFG. First, we define the following three distributions:

Definition 6.7. For even $n \in \mathbb{N}$ and for all $C \in \mathbb{N}$, let $\mathcal{S}_{n,C}$ be the distribution of S defined as

$$S\colon \mathbb{Z}^n \to \mathbb{Z}, (x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n/2} \alpha_i x_{\sigma(2i-1)} x_{\sigma(2i)} + \sum_{i=1}^{n} \beta_i x_i + \gamma,$$

where $\alpha_1, \ldots, \alpha_{n/2}, \beta_1, \ldots, \beta_n$, and γ are independent and distributed uniformly over $\{-C, \ldots, C\}$, and σ is chosen independently and uniformly from the set of permutations of $\{1, \ldots, n\}$.

Definition 6.8. For $n \in \mathbb{N}$ and for all $C \in \mathbb{N}$, let $\mathcal{MQ}_{n,C}$ be the distribution of MQ defined as

$$\mathrm{MQ}\colon \mathbb{Z}^n \to \mathbb{Z}, (x_1, \ldots, x_n) \mapsto \sum_{1 \leq i \leq j \leq n} \alpha_{i,j} x_i x_j + \sum_{i=1}^{n} \beta_i x_i + \gamma,$$

where for $1 \leq i \leq j \leq n$, $\alpha_{i,j}$, β_i, and γ are independent and distributed uniformly over $\{-C, \ldots, C\}$.

Definition 6.9. For $B_1, B_2 \in \mathbb{N}$ with $B_1 \leq B_2$, let $\mathcal{X}_{B_1,B_2}$ be the distribution that with probability 1/2 samples uniformly from $\{-B_1, \ldots, B_1\}$ and with probability 1/2 samples uniformly form $\{-B_2, \ldots, B_2\}$. That is, if X is sampled from $\mathcal{X}_{B_1,B_2}$, then

$$\forall x \in \mathbb{Z} \quad \Pr(X = x) = \begin{cases} \frac{1}{2} \cdot \frac{1}{2B_1+1} + \frac{1}{2} \cdot \frac{1}{2B_2+1}, & |x| \leq B_1, \\ \frac{1}{2} \cdot \frac{1}{2B_2+1}, & B_1 < |x| \leq B_2, \\ 0, & |x| > B_2. \end{cases}$$

Using the above distributions, we next define our candidate construction. It basically samples from these distributions and sets the generator to be the sum of S and MQ. The seed for S is sampled from $\mathcal{X}_{B_1,B_2}$, and the seed for MQ is uniform.

Definition 6.10. Let n, n', and m be polynomials, where $n(\lambda)$ is even for all $\lambda \in \mathbb{N}$. Further let $C \in \mathbb{N}$ be a constant, let B_1, B_2 be nonconstant polynomials with $B_1(\lambda) \leq B_2(\lambda)$, and let B' and C' be polynomials for all $\lambda \in \mathbb{N}$. We then define our candidate $(n+n', m)$-PFG $\mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}$ for $\lambda \in \mathbb{N}$ as follows: $\mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}$ samples a function PFG: $\mathbb{Z}^{n(\lambda)+n'(\lambda)} \to \mathbb{Z}^{m(\lambda)}$ by sampling $S_i \overset{\$}{\leftarrow} \mathcal{S}_{n(\lambda),C}$, $\mathrm{MQ}_i \overset{\$}{\leftarrow} \mathcal{MQ}_{n'(\lambda),C'(\lambda)}$ for $i \in \{1,\ldots,m(\lambda)\}$, and setting

$$\mathrm{PFG}\big((x_i)_{i\in[n(\lambda)]},(x'_i)_{i\in[n'(\lambda)]}\big) = \Big(S_j\big((x_i)_{i\in[n(\lambda)]}\big) + \mathrm{MQ}_j\big((x'_i)_{i\in[n'(\lambda)]}\big)\Big)_{j\in[m(\lambda)]}.$$

Moreover, let $\mathcal{D}_{\text{seed}}$ be the distribution that samples $x_1,\ldots,x_{n(\lambda)}$ from $\mathcal{X}_{B_1(\lambda),B_2(\lambda)}$, samples $x'_1,\ldots,x'_{n'(\lambda)}$ uniformly from $\{-B'(\lambda),\ldots,B'(\lambda)\}$, and outputs $\big(x_1,\ldots,x_{n(\lambda)},x'_1,\ldots,x'_{n'(\lambda)}\big)$.

Efficiency of our candidate is clear. We prove in the next section that it is also (K,μ)-pseudo-flawed-smudging for B-bounded distributions, for an appropriate choice of parameters, if the following assumption holds:

Assumption 6.11. Let $n, n', m, B_1, B_2, B', C$, and C' be as in Definition 6.10. The assumption is that for $\lambda \in \mathbb{N}$, there exist distributions $\mathcal{DS}(\lambda)$, $\mathcal{D}'(\lambda)$ such that $\mathcal{DS}(\lambda)$ is the joint distribution of mutually independent $Y_1,\ldots,Y_{m(\lambda)}$, where the distribution of each Y_i equals $\mathcal{S}_{n(\lambda),C}\big((\mathcal{X}_{B_1(\lambda),B_2(\lambda)})^{n(\lambda)}\big)$, and the following ensembles are μ-indistinguishable:

$$\Big\{(\mathrm{PFG},\mathcal{D}_{\text{seed}}) \overset{\$}{\leftarrow} \mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}; \text{seed} \overset{\$}{\leftarrow} \mathcal{D}_{\text{seed}} : (\mathrm{PFG},\mathrm{PFG}(\text{seed}))\Big\}_{\lambda\in\mathbb{N}},$$
$$\Big\{(\mathrm{PFG},\mathcal{D}_{\text{seed}}) \overset{\$}{\leftarrow} \mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}; Y \overset{\$}{\leftarrow} \mathcal{DS}(\lambda); Z \overset{\$}{\leftarrow} \mathcal{D}'(\lambda) : (\mathrm{PFG}, Y+Z)\Big\}_{\lambda\in\mathbb{N}}.$$

The intuition is as follows: We show that $\mathcal{S}_{n(\lambda),C}\big((\mathcal{X}_{B_1(\lambda),B_2(\lambda)})^{n(\lambda)}\big)$ is smudging (see Corollary 6.17). By Proposition 6.6, we get that if all components are sampled independently from that distribution, the resulting distribution is flawed-smudging. This is preserved when adding independent values by Lemma 6.2. The addition is supposed to make uncovering dependencies between the components harder, which is formalized by Assumption 6.11.

6.2.2 Flawed-Smudging Property of Candidate

We first need to prove some lemmata.

Lemma 6.12. *Let $B \in \mathbb{N}$, $c_1, c_2 \in \mathbb{Z}\setminus\{0\}$ such that $\gcd(c_1,c_2) = 1$, and let X_1, X_2 be independent and uniformly distributed over $\{-B,\ldots,B\}$. Then,*

$$\forall z,e \in \mathbb{Z} \quad |\Pr(c_1X_1 + c_2X_2 = z) - \Pr(c_1X_1 + c_2X_2 = z - e)| \leq \frac{1}{(2B+1)^2}\cdot\left(\frac{|e|}{|c_1c_2|}+2\right).$$

Proof. Let $z \in \mathbb{Z}$. Since X_1 and X_2 are independent, we have

$$\Pr(c_1X_1 + c_2X_2 = z) = \sum_{y\in\mathbb{Z}} \Pr(c_1X_1 = y)\cdot\Pr(c_2X_2 = z-y).$$

Note that we have for $i \in \{0,1\}$ and $t \in \mathbb{Z}$, $\Pr(c_iX_i = t) = \frac{1}{2B+1}$ if $t \in c_i\cdot\{-B,\ldots,B\}$ and $\Pr(c_iX_i = t) = 0$ otherwise. Therefore,

$$\Pr(c_1X_1 + c_2X_2 = z) = \frac{1}{(2B+1)^2}\cdot\underbrace{\big|\{y \in c_1\cdot\{-B,\ldots,B\} \mid z-y \in c_2\cdot\{-B,\ldots,B\}\}\big|}_{=:D}.$$

Let $d_i \coloneqq |c_i| \cdot B$ for $i \in \{1,2\}$. We then have $D = c_1\mathbb{Z} \cap [-d_1, d_1] \cap (c_2\mathbb{Z} + z) \cap [-d_2 + z, d_2 + z]$. Because c_1 and c_2 are coprime, the Chinese remainder theorem implies that there exists some $y_0 \in \mathbb{N}$ such that $y_0 < c_1c_2$ and $c_1\mathbb{Z} \cap (c_2\mathbb{Z} + z) = y_0 + c_1c_2\mathbb{Z}$. Hence, $D = (y_0 + c_2c_2\mathbb{Z}) \cap [-d_1, d_1] \cap [-d_2 + z, d_2 + z]$. As an intersection of two intervals, $[-d_1, d_1] \cap [-d_2 + z, d_2 + z]$ is an interval and therefore contains at least $\left\lfloor \frac{\ell}{c_1c_2} \right\rfloor$ integers congruent to y_0 modulo c_1c_2, and at most $\left\lceil \frac{\ell}{c_1c_2} \right\rceil$ of these, for $\ell \coloneqq |[-d_1, d_1] \cap [-d_2 + z, d_2 + z] \cap \mathbb{Z}|$. We can therefore conclude that for all $z \in \mathbb{Z}$,

$$\Pr(c_1X_1 + c_2X_2 = z) \in \left\{ \frac{1}{(2B+1)^2} \cdot \left(\frac{|[-d_1, d_1] \cap [-d_2 + z, d_2 + z] \cap \mathbb{Z}|}{c_1c_2} + r \right) \,\middle|\, r \in [-1, 1] \right\}.$$

Now let $e \in \mathbb{Z}$. Note that the number of integers in $[-d_1, d_1] \cap [-d_2 + z, d_2 + z]$ and in $[-d_1, d_1] \cap [-d_2 + z - e, d_2 + z - e]$ differ by at most $|e|$. This implies that $\Pr(c_1X_1 + c_2X_2 = z)$ and $\Pr(c_1X_1 + c_2X_2 = z - e)$ differ by at most $\frac{1}{(2B+1)^2} \cdot \left(\frac{|e|}{|c_1c_2|} + 2 \right)$. □

Lemma 6.13. *Let $B \in \mathbb{N}$, $c_1, c_2 \in \mathbb{Z}$ such that $\gcd(c_1, c_2) = 1$, and let X_1, X_2 be independent and uniformly distributed over $\{-B, \dots, B\}$. Further let $e \in \mathbb{Z}$, $|e| \le 2B + 1$. Then,*

$$\delta(c_1X_1 + c_2X_2, c_1X_1 + c_2X_2 + e) \le \frac{2|e| + |c_1| + |c_2| + 2}{2B + 1}.$$

Proof. By definition, we have

$$\delta(c_1X_1 + c_2X_2, c_1X_1 + c_2X_2 + e) = \frac{1}{2} \sum_{z \in \mathbb{Z}} |\Pr(c_1X_1 + c_2X_2 = z) - \Pr(c_1X_1 + c_2X_2 + e = z)|.$$

If $c_1 = 0$, then $\gcd(c_1, c_2) = 1$ implies $c_2 \in \{-1, 1\}$. In this case,

$$\begin{aligned} \delta(c_2X_2, c_2X_2 + e) &= \frac{1}{2} \cdot \frac{1}{2B+1} \cdot |\{z \in Z \mid (z \in \{-B, \dots, B\} \land z \notin \{-B + e, \dots, B + e\}) \\ &\qquad\qquad\qquad\qquad \lor (z \notin \{-B, \dots, B\} \land z \in \{-B + e, \dots, B + e\})| \\ &= \frac{1}{2} \cdot \frac{1}{2B+1} \cdot 2|e| \le \frac{2|e| + |c_1| + |c_2| + 2}{2B + 1}. \end{aligned}$$

Hence, the claim of the lemma holds for $c_1 = 0$. Analogously, it holds for $c_2 = 0$.

Now assume that $c_1 \neq 0 \neq c_2$. Note that for all z with $|z| > (|c_1| + |c_2|)B + |e|$, we have $\Pr(c_1X_1 + c_2X_2 = z) = 0$ and $\Pr(c_1X_1 + c_2X_2 + e = z) = 0$. That is, there are at most $2(|c_1| + |c_2|)B + 2|e| + 1$ values of z for which these probabilities can differ. Moreover, by Lemma 6.12, they differ by at most $\frac{1}{(2B+1)^2} \cdot \left(\frac{|e|}{|c_1c_2|} + 2 \right)$. The statistical distance is thus at most

$$\begin{aligned} &\frac{1}{2} \cdot \big(2(|c_1| + |c_2|)B + 2|e| + 1\big) \cdot \frac{1}{(2B+1)^2} \cdot \left(\frac{|e|}{|c_1c_2|} + 2 \right) \\ &\qquad = \frac{1}{2} \cdot \frac{1}{(2B+1)^2} \cdot \left(\frac{2|e|(|c_1| + |c_2|)B + 2|e|^2 + |e|}{|c_1c_2|} + 4(|c_1| + |c_2|)B + 4|e| + 2 \right). \end{aligned}$$

Since $|c_1| + |c_2| \le 2 \cdot \max\{|c_1|, |c_2|\}$ and $|c_1|, |c_2| \ge 1$, we have $\frac{|c_1|+|c_2|}{|c_1c_2|} \le 2$. This implies

$$\begin{aligned} \delta(c_1X_1 + c_2X_2, c_1X_1 + c_2X_2 + e) &\le \frac{1}{(2B+1)^2} \cdot \frac{4|e|B + 2|e|^2 + |e| + 4(|c_1| + |c_2|)B + 4|e| + 2}{2} \\ &= \frac{1}{(2B+1)^2} \big(|e|(2B + 1) + |e|^2 + \tfrac{3}{2}|e| + 2(|c_1| + |c_2|)B + 1 \big). \end{aligned}$$

Because for all $0 < b \leq a$, $\frac{a}{b} - \frac{a+1}{b+1} = \frac{ab+a-ab-b}{b(b+1)} \geq 0$, we have $\frac{2(|c_1|+|c_2|)B+1}{2B+1} \leq \frac{2(|c_1|+|c_2|)B}{2B} = |c_1| + |c_2|$. Using $|e| \leq 2B + 1$, we further have $\frac{|e|}{2B+1} \leq 1$. This yields

$$\delta(c_1X_1 + c_2X_2, c_1X_1 + c_2X_2 + e) \leq \frac{|e| + |e| + \frac{3}{2} + |c_1| + |c_2|}{2B + 1} \leq \frac{2|e| + |c_1| + |c_2| + 2}{2B + 1}. \qquad \square$$

Lemma 6.14. *Let $C \in \mathbb{N} \setminus \{0\}$ and let X, Y be independent and distributed uniformly over $\{-C, \dots, C\}$. Then,*

$$\Pr(\gcd(X, Y) = 1) \geq \frac{13}{22}.$$

Proof. Let X', Y' be independent and distributed uniformly over $\{0, \dots, C\}$. By a result of Fontein and Wocjan [FW14, Proposition 3.1], we have

$$\Pr(\gcd(X', Y') = 1) \geq \frac{13}{22}. \tag{9}$$

Note that X and Y are coprime if and only if $|X|$ and $|Y|$ are coprime. The distributions of $|X|$ and $|Y|$ are not uniform because 0 has smaller probability than all other numbers. Since 0 is only coprime to 1, and all numbers are coprime to 1, decreasing the probability of 0 can only increase the probability of the numbers being coprime. Hence,

$$\Pr(\gcd(X, Y) = 1) = \Pr(\gcd(|X|, |Y|) = 1) \geq \Pr(\gcd(X', Y') = 1) \geq \frac{13}{22}. \qquad \square$$

Lemma 6.15. *Let $\alpha_1, \beta_1, \alpha_2, \beta_2 \in \mathbb{Z}$ such that $\gcd(\alpha_1, \beta_1) = \gcd(\alpha_2, \beta_2) = 1$. We then have for all sufficiently large $B \in \mathbb{N}$ and independent X_1, X_2 that are distributed uniformly over $\{-B, \dots, B\}$,*

$$\Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1) \geq \frac{1}{2}.$$

Proof. If $\alpha_i = 0$ for some $i \in \{0, 1\}$, then $\gcd(\alpha_i, \beta_i) = 1$ implies $\beta_i = 1$ and therefore $\Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1) = 1$ in this case. Assume now that $\alpha_1 \neq 0 \neq \alpha_2$. Then, $\alpha_1X_1 + \beta_1$ and $\alpha_2X_2 + \beta_2$ are nonconstant polynomials and we have by [CS87, equation 5.1, see also Theorem 4.1][7]

$$\lim_{N\to\infty} \frac{1}{N^2} \cdot \left|\left\{(x_1, x_2) \in \{1, \dots, N\}^2 \mid \gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1\right\}\right| = \frac{1}{\zeta(2)} \prod_{\substack{p|\alpha_1\alpha_2 \\ p \text{ prime}}} \frac{1}{1 - \frac{1}{p^2}},$$

where ζ is the Riemann zeta function. Since $\frac{1}{1-\frac{1}{p^2}} \geq 1$ for all primes p, the above limit is at least $\frac{1}{\zeta(2)} = \frac{6}{\pi^2} > 0.6$. This implies that for all sufficiently large B,

$$\Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1 \mid X_1 > 0 \wedge X_2 > 0) \geq 0.6.$$

For $X_i < 0$, we have $\alpha_iX_i + \beta_i = -\alpha_i(-X_i) + \beta_i$ and similarly for $X_2 < 0$. Therefore, we can also lower bound the probability by 0.6 for negative values of X_1 and X_2. Moreover, we have for $B \geq 6$ that $\Pr(X_1 \neq 0 \wedge X_2 \neq 0) = \left(\frac{2B}{2B+1}\right)^2 \geq \frac{5}{6}$. Hence, we obtain for all sufficiently large B

$$\begin{aligned}
&\Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1) \\
&\quad \geq \Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1 \mid X_1 \neq 0 \wedge X_2 \neq 0) \cdot \Pr(X_1 \neq 0 \wedge X_2 \neq 0) \\
&\quad \geq 0.6 \cdot \frac{5}{6} = \frac{1}{2}. \qquad \square
\end{aligned}$$

[7] Note that equation 5.1 in [CS87] contains $\sum_{p|\alpha_1\alpha_2}$ instead of $\prod_{p|\alpha_1\alpha_2}$, which is a typo.

Using the above lemmata, we can now prove that if S is chosen from $\mathcal{S}_{n,C}$, it will with high probability be "good" and for all (fixed) "good" S and small values of $e \in \mathbb{Z}$, $S(X_1, \ldots, X_n)$ and $S(X_1, \ldots, X_n) + e$ are statistically close, where $X_1, \ldots, X_n$ are sampled from $\mathcal{X}_{B_1,B_2}$.

Proposition 6.16. *Let $C \in \mathbb{N} \setminus \{0\}$. We then have for all even $n \in \mathbb{N}$, for all sufficiently large $B_1 \in \mathbb{N}$ and all $B_2 \geq B_1$, for S chosen from $\mathcal{S}_{n,C}$, that with probability (over the choice of S) of at least $1 - 2^{-n/141}$, it is the case that for $X_1, \ldots, X_n$ chosen independently from $\mathcal{X}_{B_1,B_2}$ (and now fixed S) and for all $e \in \mathbb{Z}$ with $|e| \leq 2B_2 + 1$,*

$$\delta(S(\mathbf{X}), S(\mathbf{X}) + e) \leq \frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1} + \left(\frac{31}{32}\right)^{n/16},$$

where $\mathbf{X} := (X_1, \ldots, X_n)$.

Proof. Note that

$$S(\mathbf{X}) = \sum_{i=1}^{n/2} \Big(\big(\alpha_i X_{\sigma(2i-1)} + \beta_{\sigma(2i)}\big) \cdot X_{\sigma(2i)} + \beta_{\sigma(2i-1)} X_{\sigma(2i-1)} \Big) + \gamma.$$

For $i \in \{1, \ldots, n/2\}$, let E_i be the event that $\gcd\big(\alpha_i, \beta_{\sigma(2i)}\big) = 1$. By Lemma 6.14, we have $\Pr(E_i) \geq \frac{13}{22}$. Since these events are independent, we further have $\Pr(E_{2i-1} \cap E_{2i}) \geq \frac{13^2}{22^2}$. Now let for $i \in \{1, \ldots, n/4\}$, G_i be the random variable that is 1 if $E_{2i-1} \cap E_{2i}$ occurs, and 0 otherwise. Let $p_G := \Pr(G_1 = 1) = \ldots = \Pr(G_{n/4} = 1) \geq \frac{13^2}{22^2}$. We then have by the Chernoff-Hoeffding bound for all $\epsilon > 0$

$$\Pr\left(\sum_{i=1}^{n/4} G_i \leq \left(\frac{13^2}{22^2} - \epsilon \right) \frac{n}{4} \right) \leq \Pr\left(\sum_{i=1}^{n/4} G_i \leq (p_G - \epsilon) \frac{n}{4} \right) \leq \exp\left(-2\epsilon^2 \cdot \frac{n}{4} \right).$$

In particular, we obtain for $\epsilon = \frac{12}{121}$

$$\Pr\left(\sum_{i=1}^{n/4} G_i \leq \frac{n}{16} \right) \leq \exp\left(-2 \Big(\frac{12}{121} \Big)^2 \cdot \frac{n}{4} \right) = \exp\left(-\frac{72}{14641} n \right) < 2^{-n/141}.$$

Let

$$I_S := \big\{ i \in 2 \cdot \{1, \ldots n/4\} \bigm| \gcd\big(\alpha_{i-1}, \beta_{\sigma(2(i-1))}\big) = \gcd\big(\alpha_i, \beta_{\sigma(2i)}\big) = 1 \big\}.$$

The result above then implies that $\Pr(|I_S| \geq n/16) \geq 1 - 2^{-n/141}$.

Now assume S is chosen (and fixed) such that $|I_S| \geq n/16$. For $i \in I_S$, let G'_i be the event that

(i) $|X_{\sigma(2(i-1)-1)}| \leq B_1$ and $|X_{\sigma(2i-1)}| \leq B_1$,

(ii) $X_{\sigma(2(i-1))}$ and $X_{\sigma(2i)}$ are distributed uniformly over $\{-B_2, \ldots, B_2\}$, and

(iii) $\alpha_{i-1} X_{\sigma(2(i-1)-1)} + \beta_{\sigma(2(i-1))}$ and $\alpha_i X_{\sigma(2i-1)} + \beta_{\sigma(2i)}$ are coprime.

Note that conditions (i) and (ii) are satisfied with probability at least 1/4 each by definition of the distribution $\mathcal{X}_{B_1,B_2}$, and condition (iii) is satisfied with probability at least 1/2 for sufficiently large B_1 by Lemma 6.15. We thus have for sufficiently large B_1 and for all $i \in I_S$

$$\Pr(G'_i) \geq \frac{1}{32}. \tag{10}$$

If we fix $X_{\sigma(2(i-1)-1)} = x_{\sigma(2(i-1)-1)}$ and $X_{\sigma(2i-1)} = x_{\sigma(2i-1)}$ and assume G'_i, then $S(\mathbf{X})$ is of the form

$$c_1 X_{\sigma(2(i-1))} + c_2 X_{\sigma(2i)} + Y,$$

for some constants c_1, c_2 with $\gcd(c_1, c_2) = 1$ and $|c_1|, |c_2| \le C \cdot (B_1 + 1)$, and some Y that is independent from $X_{\sigma(2(i-1))}$ and $X_{\sigma(2i)}$. Since Y and $c_1 X_{\sigma(2(i-1))} + c_2 X_{\sigma(2i)}$ are independent, the statistical distance of $S(\mathbf{X})$ and $S(\mathbf{X}) + e$ for $e \in \mathbb{Z}$ can in this case be upper bounded by the statistical distance of $c_1 X_{\sigma(2(i-1))} + c_2 X_{\sigma(2i)}$ and $c_1 X_{\sigma(2(i-1))} + c_2 X_{\sigma(2i)} + e$. By Lemma 6.13, this can in turn be upper bounded by

$$\frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1}.$$

For $i \in I_S$, let $F_{i,x,x'}$ be the event that $X_{\sigma(2i-3)} = x \wedge X_{\sigma(2i-1)} = x'$ for $x, x' \in \mathbb{Z}$. Then, the statistical distance of $S(\mathbf{X})$ conditioned on G'_i and $S(\mathbf{X}) + e$ conditioned on G'_i is

$$\begin{aligned}
&\frac{1}{2} \sum_{z \in \mathbb{Z}} \left| \Pr(S(\mathbf{X}) = z \mid G'_i) - \Pr(S(\mathbf{X}) + e = z \mid G'_i) \right| \\
&= \frac{1}{2} \sum_{z \in \mathbb{Z}} \left| \sum_{\substack{x,x' \in \mathbb{Z} \\ \Pr(G'_i \cap F_{i,x,x'}) \neq 0}} \Pr(F_{i,x,x'}) \cdot \big(\Pr(S(\mathbf{X}) = z \mid G'_i \cap F_{i,x,x'}) - \Pr(S(\mathbf{X}) + e = z \mid G'_i \cap F_{i,x,x'}) \big) \right| \\
&\le \sum_{\substack{x,x' \in \mathbb{Z} \\ \Pr(G'_i \cap F_{i,x,x'}) \neq 0}} \Pr(F_{i,x,x'}) \cdot \underbrace{\frac{1}{2} \sum_{z \in \mathbb{Z}} \left| \Pr(S(\mathbf{X}) = z \mid G'_i \cap F_{i,x,x'}) - \Pr(S(\mathbf{X}) + e = z \mid G'_i \cap F_{i,x,x'}) \right|}_{\le \frac{2|e| + 2C \cdot (B_1+1) + 2}{2B_2 + 1}} \\
&\le \frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1}.
\end{aligned}$$

Overall, we get using equation (10) and $|I_S| \ge n/16$,

$$\begin{aligned}
\delta(S(\mathbf{X}), S(\mathbf{X}) + e) &\le \frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1} + \Pr(\forall i \in I_S\ \neg G'_i) \\
&\le \frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1} + \left(\frac{31}{32}\right)^{n/16}.
\end{aligned}$$

□

Corollary 6.17. *Let $C \in \mathbb{N} \setminus \{0\}$. We then have for all even $n \in \mathbb{N}$, for all sufficiently large $B_1 \in \mathbb{N}$ and all $B_2 \ge B_1$, for S chosen from $\mathcal{S}_{n,C}$, and $X_1, \ldots, X_n$ chosen independently from $\mathcal{X}_{B_1,B_2}$, that the distribution of $S(X_1, \ldots, X_n)$ is (B, ε)-smudging for all $B \le 2B_2 + 1$ and*

$$\varepsilon = \frac{2B + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1} + 2 \cdot \left(\frac{31}{32}\right)^{n/16}.$$

Proof. Let $e \in \mathbb{Z}$, $|e| \le B$, and let $\varepsilon' := \frac{2|e| + 2C \cdot (B_1+1) + 2}{2B_2 + 1} + \left(\frac{31}{32}\right)^{n/16}$. Further let G be the set of all s_G with $\delta(s_G(\mathbf{X}), s_G(\mathbf{X}) + e) \le \varepsilon'$ and let $\overline{G}$ be the set of all other s. By Proposition 6.16, we have $\Pr\big[S \in \overline{G}\big] \le 2^{-n/141}$. Therefore,

$$\begin{aligned}
\delta(S(\mathbf{X}), S(\mathbf{X}) + e) &= \sum_{s \in G} \Pr[S = s] \cdot \delta(s(\mathbf{X}), s(\mathbf{X}) + e) + \sum_{s \in \overline{G}} \Pr[S = s] \cdot \delta(s(\mathbf{X}), s(\mathbf{X}) + e) \\
&\le \Pr[S \in G] \cdot \varepsilon' + \Pr[S \in \overline{G}] \\
&\le \varepsilon' + 2^{-n/141}.
\end{aligned}$$

Since $|e| \leq B$ and $\left(\frac{1}{2}\right)^{n/141} \leq \left(\frac{31}{32}\right)^{n/16}$, we have $\varepsilon' + 2^{-n/141} \leq \varepsilon$, which concludes the proof. □

Putting our results together, we obtain the main result of this section:

Theorem 6.18. *Let $n, n', m, B_1, B_2, B', C$, and C' be as in Definition 6.10. Then, Assumption 6.11 implies that $\mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}$ is (K, μ')-pseudo-flawed-smudging for B-bounded distributions, where $\mu' = 2\left(\frac{11n\cdot(2B+1)\varepsilon}{K+1}\right)^{K+1}$ and $\varepsilon = \frac{2B+2C\cdot(B_1+1)+2}{2B_2+1} + 2 \cdot \left(\frac{31}{32}\right)^{n/16}$, if $\varepsilon \leq \frac{K+1}{22n\cdot(2B+1)}$ and $B \leq 2B_2 + 1$.*

Proof. Let $\mathcal{DS}(\lambda)$, $\mathcal{D}'(\lambda)$ be the distributions that exist by Assumption 6.11. Then, $\mathcal{DS}(\lambda)$ is the joint distribution of mutually independent $Y_1, \ldots, Y_{m(\lambda)}$, where the distribution of each Y_i equals $\mathcal{S}_{n(\lambda),C}\big((\mathcal{X}_{B_1(\lambda),B_2(\lambda)})^{n(\lambda)}\big)$. By Corollary 6.17, we have for all sufficiently large λ that the distributions of each Y_i are (B, ε)-smudging. Then by Proposition 6.6, $\mathcal{DS}(\lambda)$ is (K, μ')-flawed-smudging for B-bounded distributions. By Lemma 6.2, the distribution of the sum of samples from $\mathcal{DS}(\lambda)$ and $\mathcal{D}'(\lambda)$ are (K, μ')-flawed-smudging for B-bounded distributions. The μ-indistinguishability guaranteed by Assumption 6.11 finally implies the claim. □

Remark 6.19. Note that the distribution $\mathcal{MQ}_{n,C}$ and the distribution of its part of the seed is only relevant for Assumption 6.11, but not for the proof of Theorem 6.18. One can therefore replace these distributions by all other distributions for which Assumption 6.11 is satisfied.

6.2.3 Choice of Parameters

For the construction of our FE scheme for constant degree polynomials in Section 4, we need an $(S^{1-\alpha}, S)$-PFG that is (K, μ)-flawed-smudging for B-bounded distributions with $K = \lambda^{\varepsilon_2}$ and subexponentially small μ. To satisfy this, we can take our candidate $\mathcal{CPFG}_\lambda^{n,n',m,B_1,B_2,B',C,C'}$ and set the parameters as follows: Set $n = \lceil S^{1-\alpha}/2 \rceil$, $n' = S^{1-\alpha} - n$, and $m = S$. Further set B_1, B', C, and C' arbitrarily, and set

$$B_2 \geq \Omega(nB^2 + nBB_1).$$

Theorem 6.18 then implies that under Assumption 6.11, our candidate is (K, μ)-flawed-smudging for B-bounded distributions, for

$$\mu = O\left(\frac{1}{K+1}\right)^{K+1} = O\Big(\big(\lambda^{-\varepsilon_2}\big)^{(\lambda^{\varepsilon_2})}\Big).$$

Acknowledgments. Many thanks to Shweta Agrawal for sharing an early version of [Agr18a].

The authors are supported by NSF grants CNS-1528178, CNS-1514526, CNS-1652849 (CAREER), a Hellman Fellowship, the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236, and a subcontract No. 2017-002 through Galois. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography*, pages 528–556, Berlin, Heidelberg, 2015. Springer.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 657–677, Berlin, Heidelberg, 2015. Springer.

[ADGM17] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[AGIS14] Prabhanjan Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 646–658, New York, NY, USA, 2014. ACM.

[Agr18a] Shweta Agrawal. New methods for indistinguishability obfuscation: Bootstrapping and instantiation. Cryptology ePrint Archive, Report 2018/633, 2018. https://eprint.iacr.org/2018/633.

[Agr18b] Shweta Agrawal. Personal communication and a previous version of eprint report 2018/633, 2018.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC0. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 166–175, 10 2004.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *computational complexity*, 15(2):115–162, 6 2006.

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

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

[AS17] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 152–181, Cham, 2017. Springer International Publishing.

[BBKK18] Boaz Barak, Zvika Brakerski, Ilan Komargodski, and Pravesh K. Kothari. Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 649–679, Cham, 2018. Springer International Publishing.

[BCFG17] Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 67–98, Cham, 2017. Springer International Publishing.

[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 Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 1–18, Berlin, Heidelberg, 2001. Springer.

[BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 337–367, Berlin, Heidelberg, 2015. Springer.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under ddh. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 509–539, Berlin, Heidelberg, 2016. Springer.

[BGK$^+$14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 221–238, Berlin, Heidelberg, 2014. Springer.

[BGP06] Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A practical stream cipher with provable security. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 109–128, Berlin, Heidelberg, 2006. Springer.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[BNPW16] Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From cryptomania to obfustopia through secret-key functional encryption. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 391–418, Berlin, Heidelberg, 2016. Springer.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *Theory of Cryptography*, pages 1–25, Berlin, Heidelberg, 2014. Springer.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *Theory of Cryptography (TCC)*, pages 253–273, Berlin, Heidelberg, 2011. Springer.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, 10 2011.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 171–190, 10 2015.

[CGH$^+$15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 247–266, Berlin, Heidelberg, 2015. Springer.

[CGH17] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 278–307, Cham, 2017. Springer International Publishing.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 3–12, Berlin, Heidelberg, 2015. Springer.

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

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 267–286, Berlin, Heidelberg, 2015. Springer.

[CM01] Mary Cryan and Peter Bro Miltersen. On pseudorandom generators in NC0. In Jiří Sgall, Aleš Pultr, and Petr Kolman, editors, *Mathematical Foundations of Computer Science 2001*, pages 272–284, Berlin, Heidelberg, 2001. Springer.

[CS87] J. Chidambaraswamy and R. Sitaramachandrarao. On the probability that the values of m polynomials have a given g.c.d. *Journal of Number Theory*, 26(3):237 – 245, 1987.

[DGG$^+$16] Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Pratyay Mukherjee. Obfuscation from low noise multilinear maps. Cryptology ePrint Archive, Report 2016/599, 2016. https://eprint.iacr.org/2016/599.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[FW14] Felix Fontein and Pawel Wocjan. On the probability of generating a lattice. *Journal of Symbolic Computation*, 64:3 – 15, 2014. Mathematical and computer algebra techniques in cryptology.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme.* PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[GGH$^+$13] 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, 10 2013.

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

[GKP$^+$13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 555–564, New York, NY, USA, 2013. ACM.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240, 2010.

[GLSW15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 151–170, 10 2015.

[GMM$^+$16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 241–268, Berlin, Heidelberg, 2016. Springer.

[HLY12] Yun-Ju Huang, Feng-Hao Liu, and Bo-Yin Yang. Public-key cryptography from new multivariate quadratic assumptions. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, pages 190–205, Berlin, Heidelberg, 2012. Springer.

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Stephan Eidenbenz, Francisco Triguero, Rafael Morales, Ricardo Conejo, and Matthew Hennessy, editors, *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 244–256, Berlin, Heidelberg, 2002. Springer.

[KNT18] Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Obfustopia built on secret-key functional encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 603–648, Cham, 2018. Springer International Publishing.

[LATV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1219–1234, New York, NY, USA, 2012. ACM.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 28–57, Berlin, Heidelberg, 2016. Springer.

[Lin17] Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 599–629, Cham, 2017. Springer International Publishing.

[LPST16a] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, pages 447–462, Berlin, Heidelberg, 2016. Springer.

[LPST16b] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography*, pages 96–124, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 239–256, Berlin, Heidelberg, 2014. Springer.

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

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 11–20, 10 2016.

[LV17] Alex Lombardi and Vinod Vaikuntanathan. Limits on the locality of pseudorandom generators and applications to indistinguishability obfuscation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography*, pages 119–137, Cham, 2017. Springer International Publishing.

[MPR07] Ueli Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 130–149, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[MST03] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On e-biased generators in NC0. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 136–145, 10 2003.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 629–658, Berlin, Heidelberg, 2016. Springer.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 735–763, Berlin, Heidelberg, 2016. Springer.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. https://eprint.iacr.org/2010/556.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 500–517, Berlin, Heidelberg, 2014. Springer.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 439–467, Berlin, Heidelberg, 2015. Springer.