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 $\chi$, 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 $\chi$ 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 $\mu$-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 $P/\text{poly}$, $\text{NC}^1$, 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: $2\text{FE}^N_S$ for Quadratic Polynomials 17  
4.2 Recursive Construction: $d+1\text{FE}^N_S$ for degree $d + 1$ Polynomials 19  
4.2.1 Proof of Special Purpose Simulation Security of $d+1\text{FE}$ 24  
4.2.2 Property of $D\text{FE}$ 29

5 **Functional Encryption for $\text{NC}_1$** 31  
5.1 Bit-fixing Homomorphic Sharing 31  
5.1.1 Construction from Threshold Multi-Key FHE 33  
5.2 Construction of FE for $\text{NC}_1$ 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 [GMM+13, BR14, BGGK+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 3 [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 $\lambda$), 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 locations. More precisely, we require the output of a PFG to be indistinguishable to a, so-called, flawed-smudging distribution $Y \leftarrow \mathcal{Y}$. It ensures that for some polynomial bound $B$, and every $B$-bounded noise vector distribution $e \leftarrow \chi$, $Y + e$ “hides” the noise vector $e$ at all but a few locations in the following sense. There is a random variable $I$ correlated with $e, Y$, representing a small ($|I| = o(\lambda)$) subset of locations to be fixed, so that, the joint distribution of $(I, e, Y + e)$ is close to that of $(I, e', Y + e)$,

$$\{ I, e, Y + e \} \approx \{ I, e', Y + e \}, \quad \text{where } e' \leftarrow \chi_{|e|, I},$$

with $$
where \( e' \) is a fresh new sample from \( \chi \) conditioned on agreeing with \( e \) at locations in \( I \) (i.e., \( e'_i = e_i \) for all \( i \in I \)). In short, given \( Y + e \), 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 \( Y \) from such distributions can hide a small noise vector \( e \) entirely at all locations (with overwhelming probability), in the sense that \( (e, Y + e) \approx (e', Y + e) \) for a completely independent \( e' \leftarrow \chi \). In fact, to hide the noise vectors \( e \) entirely, it is necessary that \( Y \) 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 \( e \) 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 \( \text{NC}_1 \) 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 \( \text{NC}_1 \), 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 \( D_1 \times \cdots \times D_m \), satisfying that every distribution \( D_i \) has sufficiently (but still polynomially) small statistical distance between \( D_i \) and \( D_i + e \) 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 \( Y \) and a sample from an arbitrary independent distribution \( T \) 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 \( D_i \) 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 \( SD(D_i, D_i + 1) \) 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 \( SD(D_i, D_i + e) \) 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 \( \text{NC}_1 \), 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 \( \text{NC}_1 \) 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 \textit{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 \( \text{NC}_1 \) must be \textit{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 \( x \), and then extract the output \( y = f(x) \) 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 \( x \) using a HE scheme and a secret vector \( s \); one can then homomorphically evaluate any function \( f \) on \( x \) and obtain a ciphertext \( \text{ct}_f \) encrypting \( y \). The difficult part is how to decrypt a ciphertext \( \text{ct}_f \) using a “legitimate” function \( f \) (ones for which secret keys are generated) to recover \( f(x) \) 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 \( \text{ct}_f \) 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 \( \langle \text{ct}_f, s \rangle \), already yields an approximate output, such as \( y + 2e \), perturbed by a small noise vector \( e \). However, the noise \( e \) is \textit{sensitive}, revealing information about the input \( x \), the HE secret \( s \), and the noises used for generating the original ciphertext encrypting \( x \). Therefore, we need a way to hide \( e \) using just degree 2 computation. This is where noise smudging comes in: We instead compute the approximate output \( y + 2e + 2Y \) further shifted by a large noise \( Y \) in order to hide \( e \). 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 \( Y \). 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 \( Y \) cannot hide \( e \) 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 \( e + Y \) hides \( e \) at \textit{all but
a few locations. However, revealing 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 \((\text{mpk}, \text{sk}_f, \text{ct}_x)\) consisting of an honestly generated master public key \text{mpk}, a secret key \text{sk}_f for a distributional function \(f \leftarrow \mathcal{F}\mathcal{N}\), and a ciphertext \(\text{ct}_x\) for a distributional input \(x \leftarrow \mathcal{X}\), can be simulated by a simulator \text{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 \text{Fix} over the fixed locations \(K\) and values \(x^*\), such that, \(|K| = o(\lambda)|, and

\[
\{ x, \text{mpk}, \text{sk}_f, \text{ct}_x \} \approx \{ x, \text{Sim} ((x^*, K), f, y = f(x)) \},
\]

where \((x^*, K) \leftarrow \text{Fix}, \text{and } x \leftarrow \mathcal{X}|x^*, K\).

In other words, given \text{mpk}, \text{sk}_f, \text{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 \(s\) comes from a small domain (e.g., \(s\) is binary), the hardness of LWE holds as long as \(s\) 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 \(e\) is a function over other distributional secret \(w \leftarrow \mathcal{R}; \text{there is again a correlated random variable } I, \text{such that,}

\[
\{ I, w, Y + e(w) \} \approx \{ I, w', Y + e(w) \}, \text{where } w' \leftarrow \chi|w, I.
\]

This means given \(Y + E(w)\), only a few locations of the secret \(w\) get fixed and leaked. In our construction of constant-degree FE, we use this guarantee to bound what information of the HE input \(x\) and secret \(s\) is fixed and leaked through leakage of the noise \(e\) 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 \(\text{NC}_1\) 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 \(\text{NC}_1\) computation \(g(v)\) into a simple constant-degree polynomial \(\hat{g}(v; r)\), and uses a constant locality PRG to supply pseudorandom coins \(r = \text{PRG(seed)}\) 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) [BG115] in syntax, but differs in security and efficiency requirements. It enables compiling a single computation \(g(v)\) into a collection of computations \(o_1 = h_1(x_1), \ldots, o_{\lambda} = h_{\lambda}(x_{\lambda})\) that operates on a secret sharing \(x_1, \ldots, x_{q}\) of the original input \(v\), and from the collection of outputs \(o_1, \ldots, o_q\), 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 \(\{\hat{h}_i(x_i, r_i)\}\). Through careful analysis, we show that the weak security of constant-degree FE only leads to a small subset of the computation \(a_i = h_i(x_i)\) 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 \(s\) 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 \(Y\) to flood the noise \(e\) in \(ct_f\), so that it is safe to reveal \(y + e + Y\). In this work, we follow her approach and explore what happens when \(Y\) is polynomially bounded and \(e\) 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 \(\text{NC}_1\) 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

\[
H_{\infty}(X) := -\log\left(\max_x \Pr[X = x]\right).
\]
We next state the decisional learning with errors (LWE) assumption, which was introduced by Dodis et al. [DORS08]

\[ H_\infty(X \mid Z) := -\log\left( \mathbb{E}_{z \sim Z} \left[ \max_x \Pr[X = x \mid Z = z] \right] \right) \]

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 \( \nu \) 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 \( \mu \)-Indistinguishability

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

\[
\left| \Pr\left[ x \overset{\$}{\leftarrow} X_\lambda : D(1^\lambda, x, z) = 1 \right] - \Pr\left[ y \overset{\$}{\leftarrow} Y_\lambda : D(1^\lambda, y, z) = 1 \right] \right| \leq O(\mu(\lambda)).
\]

### 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 \( \text{LWE}_{n,m,q,\chi} \) assumption with \( \mu \)-indistinguishability is that the following distributions are \( \mu \)-indistinguishable:

\[
\left\{ \mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}, s \overset{\$}{\leftarrow} \mathbb{Z}_q^n, e \overset{\$}{\leftarrow} \chi^m : (\mathbf{A}, \mathbf{A} \cdot s + e) \right\}_{\lambda \in \mathbb{N}}
\]

\[
\left\{ \mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}, u \overset{\$}{\leftarrow} \chi^m : (\mathbf{A}, u) \right\}_{\lambda \in \mathbb{N}}
\]

It has been shown that this assumptions holds when \( \chi \) 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 \( \text{LWE}_{n,m,q,\chi}^{\text{WL}}(\gamma,k) \) assumption with \( \mu \)-indistinguishability states that for all efficiently samplable correlated random variables \( (s, \text{aux}) \), where the support of \( s \) is \([-\gamma, \gamma]^n \cap \mathbb{Z}^n \) and \( H_\infty(s \mid \text{aux}) \geq k \), the following distributions are \( \mu \)-indistinguishable

\[
(\text{aux}, \mathbf{A}, \mathbf{A} \cdot s + e), \quad (\text{aux}, \mathbf{A}, u),
\]

where \( \mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}, e \overset{\$}{\leftarrow} \chi^m, \) and \( u \overset{\$}{\leftarrow} \mathbb{Z}_q^m \) are sampled independently of \( (s, \text{aux}) \).
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 $\text{PRG} = \{\text{PRG}_\lambda\}_{\lambda \in \mathbb{N}}$ satisfying the following properties:

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

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

**$\mu$-Indistinguishability:** The following ensembles are $\mu$-indistinguishable:

$$\left\{ \text{PRG} \overset{\$}{\leftarrow} \text{PRG}_\lambda; s \overset{\$}{\leftarrow} \{0,1\}^{n(\lambda)} : (\text{PRG},\text{PRG}(s)) \right\}_{\lambda},$$

$$\left\{ \text{PRG} \overset{\$}{\leftarrow} \text{PRG}_\lambda; r \overset{\$}{\leftarrow} \{0,1\}^{m(\lambda)} : (\text{PRG},r) \right\}_{\lambda}.$$

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

We next define pseudorandom function (PRF) families.

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

$$\left| \Pr[K \overset{\$}{\leftarrow} K : A^{\text{PRF}(K,\cdot)}(1^\lambda) = 1] - \Pr[U \overset{\$}{\leftarrow} (X \to Y) : A^{U(\cdot)}(1^\lambda) = 1] \right| \leq O(\mu(\lambda)).$$

2.5 Indistinguishability Obfuscation

We recall the notion of indistinguishability obfuscation for a class of circuit defined by $\text{BGI}^\dagger 01$.

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

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

$$\Pr[C' \leftarrow iO(1^\lambda,C) : C'(x) = C(x)] = 1$$

where the probability is taken over the coin-tosses of the obfuscator $iO$. 

9
\(\mu\)-Indistinguishability: For every ensemble of pairs of circuits \(\{C_{0,\lambda}, C_{1,\lambda}\}_{\lambda \in \mathbb{N}}\) satisfying that 
\(C_{b,\lambda} \in C_{\lambda}, |C_{0,\lambda}| = |C_{1,\lambda}|,\) and \(C_{0,\lambda}(x) = C_{1,\lambda}(x)\) for every \(x\), the following ensembles of distributions are \(\mu\)-indistinguishable:
\[
\{C_{1,\lambda}, C_{2,\lambda}, iO(1^\lambda, C_{1,\lambda})\}_{\lambda \in \mathbb{N}} \\
\{C_{1,\lambda}, C_{2,\lambda}, iO(1^\lambda, C_{2,\lambda})\}_{\lambda \in \mathbb{N}}
\]

Definition 2.9 (IO for \(P/poly\)). A uniform PPT machine \(iO_{P/poly}(\star, \star)\) is an indistinguishability obfuscator for \(P/poly\) if it is an indistinguishability obfuscator for the class \(\{C_{\lambda}\}_{\lambda \in \mathbb{N}}\) of circuits of size at most \(\lambda\).

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,

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

Correctness: The two algorithms \(\text{REnc}\) and \(\text{REval}\) satisfy the following correctness condition:
For all security parameters \(\lambda \in \mathbb{N}\), circuit \(C\), input \(x\), it holds that,
\[
\Pr\left[\hat{C}_x \xleftarrow{} \text{REnc}(1^\lambda, C, x) : \text{Eval}(\hat{C}_x) = C(x)\right] = 1
\]

\(\mu\)-Simulation Security: There exists a PPT algorithm \(\text{RSim}\), such that, for every ensemble \(\{C_{\lambda}, x_{\lambda}\}_{\lambda}\) where \(|C_{\lambda}|, |x_{\lambda}| \leq \text{poly}(\lambda)\), the following ensembles are \(\mu\)-indistinguishable for all \(\lambda \in \mathbb{N}\).
\[
\{\hat{C}_x \xleftarrow{} \text{REnc}(1^\lambda, C, x) : \hat{C}_x\}_{\lambda \in \mathbb{N}} \\
\{\hat{C}_x \xleftarrow{} \text{RSim}(1^\lambda, C(x), 1^{|C|}, 1^{|x|}) : \hat{C}_x\}_{\lambda \in \mathbb{N}}
\]

where \(C = C_{\lambda}\) and \(x = x_{\lambda}\).

Furthermore, let \(C\) be a complexity class, we say that randomized encoding scheme \(RE\) is in \(C\), if the encoding algorithm \(\text{REnc}\) 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 \( \mathcal{X} = \{ \lambda \mathcal{X} \}_{\lambda \in \mathbb{N}} \) and \( \mathcal{Y} = \{ \lambda \mathcal{Y} \}_{\lambda \in \mathbb{N}} \) 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 \( \text{FE} \) for \( \{ \mathcal{F}_\lambda \}_{\lambda \in \mathbb{N}} \) consists of four PPT algorithms (\( \text{FE.Setup, FE.KeyGen, FE.Enc, FE.Dec} \)).

- **Setup:** \( \text{FE.Setup}(1^\lambda, \text{pp}) \) 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 \( (\text{mpk}, \text{msk}) \).
- **Key Generation:** \( \text{FE.KeyGen}(\text{msk}, f) \) on input the master secret key \( \text{msk} \) and the description of a function \( f \in \mathcal{F}_\lambda \), outputs a secret key \( \text{sk}_f \).
- **Encryption:** \( \text{FE.Enc}(\text{mpk}, x) \) on input the master public key \( \text{mpk} \) and a message \( x \in \mathcal{X}_\lambda \), outputs an encryption \( \text{ct} \) of \( x \).
- **Decryption:** \( \text{FE.Dec}(\text{sk}, \text{ct}) \) 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 \( \lambda, f \in \mathcal{F}_\lambda, x \in \mathcal{X}_\lambda \), it holds that,

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

**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 \( \text{sk}_f \), 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 Full-Sel-Ind-secure). A public-key functional encryption scheme \( \text{FE} = (\text{FE.Setup, FE.KeyGen, FE.Enc, FE.Dec}) \) for \( \{ \mathcal{F}_\lambda \}_{\lambda \in \mathbb{N}} \) is 1-key \( \mu \)-Full-Sel-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 \( \mu \)-indistinguishable.

\[
\begin{cases}
(\text{mpk}, \text{msk}) \leftarrow \text{FE.Setup}(1^\lambda, \text{pp}) \\
\text{sk} \leftarrow \text{FE.KeyGen}(\text{msk}, f_\lambda) \\
\text{ct} \leftarrow \text{FE.Enc}(\text{mpk}, x^0_\lambda) \\
\lambda \in \mathbb{N}
\end{cases}
\]

\[
\begin{cases}
(\text{mpk}, \text{msk}) \leftarrow \text{FE.Setup}(1^\lambda, \text{pp}) \\
\text{sk} \leftarrow \text{FE.KeyGen}(\text{msk}, f_\lambda) \\
\text{ct} \leftarrow \text{FE.Enc}(\text{mpk}, x^1_\lambda) \\
\lambda \in \mathbb{N}
\end{cases}
\]

Note that our notion of fully-selective security is weaker than the notion of selective security in some papers in the literature (e.g., \cite{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 $FE = (FE.Setup, FE.KeyGen, FE.Enc, FE.Dec)$ for $\{F_\lambda\}_{\lambda \in \mathbb{N}}$ is 1-key $\mu$-Full-Sel-Sim-secure, if for every sequence of efficiently samplable distributions $\{D_\lambda\}_{\lambda \in \mathbb{N}}$ over pairs $(f, x)$ where $f \in F_\lambda$ and $x$ is in the domain of $f$, there exists a simulator $\text{Sim}$, such that, the following distributions are $\mu$-indistinguishable.

$$
\{(f, x) \leftarrow D_\lambda : \text{Sim}(f, f(x))\}_{\lambda \in \mathbb{N}}
$$

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

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

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

**Compactness** In the above definition of families of FE schemes for $\{F^{N,S}_N\}$, 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 $\mathcal{FE} = \{FE^{N,S}\}$ be a family of FE schemes for $\{F^{N,S}\}$.

**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 $FE^{N,S}$ 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 $FE^{N,S}$ 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 FHE.KeyGen on input a security parameter $1^\lambda$, outputs a key $s$.

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

**Evaluation:** The algorithm FHE.Eval on input a circuit $C$ and a tuple of ciphertexts $(ct_1, \ldots, ct_t)$, outputs a ciphertext $ct'$.

**Decryption:** The algorithm 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 FHE.KeyGen($1^\lambda$), all circuits $C$, all messages $x_1, \ldots, x_t$, and all ciphertexts $ct_1, \ldots, ct_t$ output by FHE.Enc($s, x_1$), $\ldots$, FHE.Enc($s, x_t$), we have

$$\Pr[ct' \overset{\$}{\leftarrow} \text{FHE.Eval}(C, (ct_1, \ldots, ct_t)) : \text{FHE.Dec}(s, ct') = C(x_1, \ldots, x_t)] = 1 - \text{negl}(\lambda).$$

**Compactness:** There is a polynomial $p$ such that for all $s$ output by FHE.KeyGen($1^\lambda$), all circuits $C$, all messages $x_1, \ldots, x_t$, all ciphertexts $ct_1, \ldots, ct_t$ output by FHE.Enc($s, x_1$), $\ldots$, FHE.Enc($s, x_t$), and all $ct'$ output by FHE.Eval($C, (ct_1, \ldots, ct_t)$), we have that $|ct'| \leq p(\lambda)$ and FHE.Dec($s, ct'$) 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 $\{\text{HE}^{(d)}\}_{d \in \mathbb{N}}$, where for all $d \in \mathbb{N}$, $\text{HE}^{(d)}$ is homomorphic for all circuits of depth at most $d$, is called leveled fully homomorphic if all $\text{HE}^{(d)}$ use the same decryption circuit, and the computational complexity of all algorithms in $\text{HE}^{(d)}$ is polynomial in $\lambda$, $d$, 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 $\text{FHE} = (\text{FHE.KeyGen}, \text{FHE.Enc}, \text{FHE.Eval}, \text{FHE.Dec})$ be a FHE scheme. We say FHE is $\mu$-CPA-secure if for every PPT adversary $A$ and for every sufficiently large $\lambda$, the advantage of $A$ in the following game is bounded by $O(\mu(\lambda))$:

- The challenger runs $s \overset{\$}{\leftarrow} \text{FHE.KeyGen}(1^\lambda)$.
- The adversary $A$ with access to an encryption oracle $\text{FHE.Enc}(s, \cdot)$ 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 $ct \overset{\$}{\leftarrow} \text{FHE.Enc}(s, x_b)$, and sends $ct$ to $A$. 

13
The adversary $A$ again has access to an encryption oracle $\text{FHE.Enc}(s, \cdot)$, and finally outputs a bit $b'$.

The advantage of $A$ is defined as

$$\text{Adv}^\text{FHE}_A := \vert 2 \cdot \Pr[b' = b] - 1 \vert.$$

### 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:

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

We require the following properties:

#### $\mu$-Semantic Security:

For any polynomial $d$ and any two messages $x_0, x_1$, the following two distributions are $\mu$-indistinguishable:

$$\begin{align*}
\{ \text{params} &\leftarrow \text{MFHE.Setup}(1^\lambda, 1^d(\lambda)) \} \\
\{ \text{pk, sk} &\leftarrow \text{MFHE.KeyGen}(\text{params}) \}
\end{align*}$$

$$\begin{align*}
\{ \text{params} &\leftarrow \text{MFHE.Setup}(1^\lambda, 1^d(\lambda)) \} \\
\{ \text{pk, sk} &\leftarrow \text{MFHE.KeyGen}(\text{params}) \}
\end{align*}$$

#### Correctness and Compactness:

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

#### Correctness of Expansion:

$\text{MFHE.Dec}(\text{params}, (\text{sk}_1, \ldots, \text{sk}_N), \hat{\text{ct}}_i) = x_i$ for all $i \in [\ell]$. 

14
Correctness of Evaluation: \( \text{MFHE.Dec}(\text{params}, (sk_1, \ldots, sk_N), \hat{ct}) = C(x_1, \ldots, x_\ell) \).

Compactness: \(|\hat{ct}|\) is polynomial in \( \lambda, d, \) and \( N \) (independently of \( C \) and \( \ell \)).

**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:

- **MFHE.PartDec** on input an expanded ciphertext \( \hat{ct} \), public keys \( pk_1, \ldots, pk_N \), an index \( i \in [N] \), and the ith secret key \( sk_i \), outputs a partial decryption \( p_i \).

- **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 \( \text{params} \xleftarrow{\$} \text{MFHE.Setup}(1^\lambda, 1^d) \), \( (pk_i, sk_i) \xleftarrow{\$} \text{MFHE.KeyGen}(\text{params}) \) for \( i \in [N] \), and let \( x_1, \ldots, x_\ell \) be messages. Further let \( I_1 \subseteq [N], \ldots, I_\ell \subseteq [N] \), and let \( ct_i \xleftarrow{\$} \text{MFHE.Enc}(pk_i, x_i) \) and \( \hat{ct}_i \xleftarrow{\$} \text{MFHE.Expand}((pk_1, \ldots, pk_N), I_i, ct_i) \) for \( i \in [\ell] \). Finally let \( C \) be a circuit of depth at most \( d \) and let \( \hat{ct} := \text{MFHE.Eval}(\text{params}, C, (\hat{ct}_1, \ldots, \hat{ct}_\ell)) \). We then require:

**Correctness of Decryption:** For \( p_i \xleftarrow{\$} \text{MFHE.PartDec}(\hat{ct}, (pk_1, \ldots, pk_N), i, sk_i), i \in [N] \), we have \( \text{MFHE.FinDec}(p_1, \ldots, p_N) = C(x_1, \ldots, x_\ell) \) with probability 1.

**\( \mu \)-Simulatability of Partial Decryptions:** There exists a PPT simulator \( \text{Sim} \) that on input \( i \in [N], (sk_j)_{j \in [N] \backslash \{i\}}, \hat{ct}, \) 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 \( p_i \xleftarrow{\$} \text{MFHE.PartDec}(\hat{ct}, (pk_1, \ldots, pk_N), i, sk_i) \). Here, the randomness for the statistical distance is only over the coins of \( \text{Sim} \) and \( \text{MFHE.PartDec} \), and all other values are fixed.

3 Definition of Pseudo Flawed-Smudging Generators

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

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

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

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

**Definition 3.4 (Flawed-Smudging Distributions).** Let \( \ell \) 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 \( E_i: \mathbb{Z}^{[\Phi_i]} \rightarrow \mathbb{Z} \) be functions. For \( \ell \subseteq [\ell], \) let \( \Phi_I := \bigcup_{i \in I} \Phi_i \). Define \( E: \mathbb{Z}^m \rightarrow \mathbb{Z}^\ell \) as

\[
E(v_1, \ldots, v_m) = (E_1((v_j)_{j \in \Phi_1}), \ldots, E_\ell((v_j)_{j \in \Phi_\ell})).
\]
We say that \( X \) is \((K, \mu)\)-flawed-smudging for \((E, \mathcal{V})\), if there exist randomized predicates \( \{ \text{BAD}_i : \mathbb{Z}^{t+1} \to \{0, 1\}\}_{i \in [t]} \) such that the following two distributions are identical,

\[
\mathcal{D}_1 = \left\{ \begin{array}{l}
V \leftarrow \mathcal{V} \\
X \leftarrow X \\
\text{bad} = (\text{bad}_i \leftarrow \text{BAD}_i(E_i(V_{\Phi_i}), X))_{i \in [t]}
\end{array} \right\},
\]

\[
\mathcal{D}_2 = \left\{ \begin{array}{l}
\overline{V} \leftarrow \mathcal{V} \\
X \leftarrow X \\
\text{bad} = (\text{bad}_i \leftarrow \text{BAD}_i(E_i(V_{\Phi_i}), X))_{i \in [t]}
\end{array} \right\},
\]

and in addition, with probability at least \( 1 - \mu \), the 1-norm of \( \text{bad} \) is bounded by \( |\text{bad}|_1 \leq K \).

We say that \( \mathcal{X} \) is \((K, \mu)\)-flawed-smudging for \( B \)-bounded distributions, if it is \((K, \mu)\)-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 \( \mathcal{PF}G = \{\mathcal{PF}G_{\lambda}\}_{\lambda \in \mathbb{N}} \) satisfying the following properties:

**Syntax:** For every \( \lambda \in \mathbb{N} \), every \((\mathcal{PF}G, D_{\text{seed}})\) in the support of \( \mathcal{PF}G_{\lambda} \) defines a function \( \mathcal{PF}G : \mathbb{Z}^{n(\lambda)} \to \mathbb{Z}^{m(\lambda)} \) and a polynomial \( (\lambda) \)-bounded distribution \( D_{\text{seed}} \) over seeds.

**Efficiency:** There is a uniform Turing machine \( M \) satisfying that for every \( \lambda \in \mathbb{N} \), every \((\mathcal{PF}G, D_{\text{seed}}) \in \text{Support}(\mathcal{PF}G_{\lambda}) \) and seed \( \in \text{Support}(D_{\text{seed}}) \), \( M(\mathcal{PF}G, \text{seed}) \) runs in time \( \text{poly}(\lambda) \) and we have \( M(\mathcal{PF}G, \text{seed}) = \mathcal{PF}G(\text{seed}) \). Furthermore, \( \mathcal{PF}G \) and all \( D_{\text{seed}} \) in the support of \( \mathcal{PF}G_{\lambda} \) are efficiently samplable.

**\((K, \mu)\)-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 \( \mu \)-indistinguishable:

\[
\left\{ (\mathcal{PF}G, D_{\text{seed}}) \xleftarrow{\$} \mathcal{PF}G_{\lambda}; \text{seed} \xleftarrow{\$} D_{\text{seed}} : (\mathcal{PF}G, \mathcal{PF}G(\text{seed})) \right\}_{\lambda \in \mathbb{N}},
\]

\[
\left\{ (\mathcal{PF}G, D_{\text{seed}}) \xleftarrow{\$} \mathcal{PF}G_{\lambda}; X \xleftarrow{\$} \mathcal{X}_{\lambda} : (\mathcal{PF}G, X) \right\}_{\lambda \in \mathbb{N}}.
\]

### 4 Functional Encryption for Constant Degree Polynomials

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

- **Family** \( d\mathcal{F}^{N,S} = \{d\mathcal{F}^{N,S}_{\lambda}\} \) 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: $2\text{FE}^{N,S}$ for Quadratic Polynomials

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

- A polynomial-stretch $(S^{1-\alpha}, S)$-PFG (PFG, $D_{\text{seed}}$), whose output is indistinguishable to a $(\lambda^{2}, \mu)$-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 $\overline{B} = \text{poly}(B, \lambda)$. In Section 6.2, we provide a candidate of such generators. (Here for convenience, we use a single PFG (PFG, $D_{\text{seed}}$). It can be replaced with a family of PFGs $\mathcal{PFG}$ by sampling (PFG, $D_{\text{seed}}$) $\leftarrow \mathcal{PFG}_{\lambda}$ in the FE setup algorithm.)

- FE schemes $\{Q\text{FE}^{N,S}\}$ 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 $\text{poly}(\lambda) N$, depending linearly on the input length and is independent of the size of computation.
  - Simulation Security: The scheme satisfies 1-key $2^{-\lambda^{1.1}}$-Full-Sel-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 $x$, $Q\text{FE}.\text{Dec}(qsk, qct) = [f(x)]_T$, where $[f(x)]_T = g^f(x)_T$ is encoding of the output in the target group of the underlying bilinear map.

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

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

- $2\text{FE}.\text{Enc}(mpk, x)$: On input $mpk = (\text{pp}, qmpk)$ and $x \in \mathbb{Z}_{p^2}^N$, do:
  - For every $3 \leq \beta \leq D$ and every $k \in [n_{\text{pad}}]$, sample $seed_{\beta,k} \leftarrow D_{\text{seed}}$.
  - Encrypt $qct \leftarrow Q\text{FE}.\text{Enc}(qmpk, (x, \{seed_{\beta,k}\}_{\beta,k}))$.

  Output $2ct = qct$.

- $2\text{FE}.\text{KeyGen}(msk, f)$: On input $msk$ and a quadratic polynomial $f$ over $\mathbb{Z}_{p^2}^N$, do:
  - Let $g(x, \{seed_{\beta,k}\}_{\beta,k})$ denote the following quadratic function: For every output element $i$ of $f$,
    \[ g_i(x, \{seed_{\beta,k}\}_{\beta,k}) := f_i(x) + \sum_{3 \leq \beta \leq D} p_{\beta} \times \left( \sum_{k \in [n_{\text{pad}}]} \text{PFG}_i(seed_{\beta,k}) \right). \]
For simplicity of notation, we suppress the subscript \( i \) and write

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

Note that since PFG is a quadratic polynomial over \( \mathbb{Z} \), \( g \) is a quadratic polynomial over \( \mathbb{Z}_p^2 \).

- Generate key \( qsk \leftarrow \text{QFE.KeyGen}(\text{qmsk}, g) \).

Output \( 2sk = qsk \).

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

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 \( \text{seed}_{\beta,k} \) has length \( S^{1-\alpha} \). Then, by the linear compactness of QFE, we have that the size of ciphertext is

\[
|2ct| = |qct| = \text{poly}(\lambda)(N + n_{\text{pad}} \times S^{1-\alpha}) = \text{poly}(\lambda)(N + S^{1-\alpha}),
\]

where the last equality follows from the fact that \( n_{\text{pad}} = \text{poly}(\lambda) \). Note that unlike standard sublinear compactness which allows ciphertext size to grow polynomially with the length of the input \( \text{poly}(\lambda, N)S^{1-\alpha} \), the ciphertext size of 2FE 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

\[
y = g(x, \{\text{seed}_{\beta,k}\})
\]

\[
= f(x) + \sum_{3 \leq \beta \leq D} p_\beta \times \left( \sum_{k \in [n_{\text{pad}}]} \text{PFG}(\text{seed}_{\beta,k}) \right)
\]

\[
= f(x) + \sum_{3 \leq \beta \leq D} p_\beta \times \Lambda_\beta,
\]

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

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

- **Special-Purpose Simulation Security:** It follows from the 1-key \( \mu \)-Full-Sel-Sim-security of QFE that if for every sequence of distributions \( \{F_{\mathcal{N}_\lambda}\} \) over \( f \in \mathcal{F}_\lambda^{N,S} \), 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 \(\{D_\lambda\}\) over \(\mathbb{Z}_{q_2}^N\), there exists a simulator \(\text{Sim}\), such that, the following distributions are \(\mu\)-indistinguishable.

\[
\begin{cases}
  f \leftarrow \mathcal{F}_N^\lambda, \ x \leftarrow D_\lambda \\
  (\text{mpk}, \text{msk}) \leftarrow \text{2FE.Setup}(1^\lambda, \text{pp}) \\
  2\text{sk} \leftarrow \text{2FE.KeyGen}(\text{msk}, f) \\
  2\text{ct} \leftarrow \text{2FE.Enc}(\text{mpk}, x)
\end{cases} \quad \lambda \in \mathbb{N}
\]

\[
\begin{cases}
  f \leftarrow \mathcal{F}_N^\lambda, \ x \leftarrow D_\lambda \\
  \{\text{seed}_{\beta,k} \leftarrow D_{\text{seed}}\}_3 \leq \beta \leq D, k \in [n_{\text{pad}}] \\
  \text{Sim}(f, y = f(x) + \sum p_{\beta} \text{PFG}(\text{seed}_{\beta,k}))
\end{cases} \quad \lambda \in \mathbb{N}
\]

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 \((\lambda^{\text{e}2}, \mu)\)-flawed-smudging distribution \(X\) for \(B\)-bounded distributions.

\[
\begin{cases}
  f \leftarrow \mathcal{F}_N^\lambda, \ x \leftarrow D_\lambda \\
  \{\Delta_{\beta,k} \leftarrow X_{\lambda}\}_3 \leq \beta \leq D, k \in [n_{\text{pad}}] \\
  \text{Sim}(f, y = f(x) + \sum p_{\beta} \Delta_{\beta,k})
\end{cases} \quad \lambda \in \mathbb{N}
\]

\section*{4.2 Recursive Construction: \(d+1\text{FE}^{N,S}\) for degree \(d+1\) Polynomials}

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

**Secret-key Homomorphic Encryption for Constant-Degree Polynomials** We describe a specific instantiation of \(\text{HE}\) 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 \(a, (a, s) + p_{d+1}e + x\) 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 \(s\) is binary from \([−1, 1]\), the hardness of LWE holds even if the secret \(s\) is only entropic instead of uniformly random. This means the semantic security of the scheme holds as long as \(s\) has sufficient entropy. Below we describe the scheme, and highlight the special properties that are useful for our construction of FE schemes.

- \(\text{HE.KeyGen}(1^n, \text{hpp} = (p_{d+1}, p_d))\) on input a security parameter \(1^n\), outputs a random binary vector \(s \leftarrow \{−1, 1\}^n\).

- \(\text{HE.Enc}(s, x)\), on input message \(x \in \mathbb{Z}_{p_{d+1}}^N\), samples a vector \(a \leftarrow \mathbb{Z}_{p_d}^n\), and noise \(e \leftarrow \chi\) where \(\chi\) is a \(\text{poly}(n)\)-bounded noise distribution, and outputs
  \[
  \text{hct} = (a, e = (a, s) + p_{d+1}e + x) \quad (\text{mod } p_d).
  \]

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

\[
\text{hct} = \text{HE.Enc}'(a, (a, s), x, e),
\]
In addition, note that \textsc{HE.Enc}' has constant locality in its input.

- \textsc{HE.Eval}(f, hct_1, \cdots, hct_t) on input a degree $d + 1$ polynomial $f : \mathbb{Z}_{p_{d+1}}^N \rightarrow \mathbb{Z}_{p_{d+1}}$, and a list of ciphertexts, expand $f$ as a sum of monomials

$$f(x) = \sum_j \text{mnl}_j(x), \text{ where } \text{mnl}_j(x) = \prod_{\gamma \in [d+1]} x_j^\gamma$$

output ciphertext

$$\text{hct}^f = \sum_j \text{hct}^{\text{mnl}_j}, \text{ where } \text{hct}^{\text{mnl}_j} = \otimes_{\gamma \in [d+1]} \text{hct}_j, \mod p_d,$$

where $\otimes$ stands for vector tensor product.

- \textsc{HE.Dec}(s, hct) learns the degree $d + 1$ of homomorphic evaluation from $|\text{hct}|$, and outputs

$$y = (s^\otimes d, \text{hct}) \mod p_d, \text{ where } s^\otimes d = \otimes_{\gamma \in [d+1]} s.$$

Note that the actual message can be extracted as $y \mod p_{d+1}$. For convenience of notation, we let \textsc{HE.Dec} output directly $y$.

Property 2 – Additive and Local Structure of Noise

By construction, for \text{hct}^f derived from ciphertexts \text{hct}_1 \cdots \text{hct}_t described above,

$$\text{HE.Dec}(s, \text{hct}^{\text{mnl}_j}) = \text{mnl}_j(x) + e^{\text{mnl}_j}(x_{j_1}, \cdots, x_{j_d}, e_{j_1}, \cdots, e_{j_d}),$$

$$\text{HE.Dec}(s, \text{hct}^f) = f(x) + e^f = f(x) + \sum_j e^{\text{mnl}_j}$$

where $x_i, e_i$ are respectively the input and noise underlying the $i$’th input ciphertext \text{hct}_i, and $e^{\text{mnl}_j}(x_{j_1}, \cdots, x_{j_d}, e_{j_1}, \cdots, e_{j_d})$ is the noise underlying \text{hct}^{\text{mnl}_j}, which depends only on the inputs and noises underlying the ciphertexts \text{hct}_j_1, \cdots, \text{hct}_j_d involved in computing this monomial.

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

Property 3 Decomposing homomorphic evaluation and decryption:

For every polynomial $f(x)$ of degree $d + 1$, we define the following polynomials

$$\text{HE.Dec}(s, \text{HE.Eval}(f, (\text{hct}_1, \cdots, \text{hct}_t))) := \hat{f}_1(s, (\text{hct}_1, \cdots, \text{hct}_t)) + \hat{f}_2(\text{hct}_1, \cdots, \text{hct}_t),$$

The total degree of $\hat{f}_1$ is $d + 1$ and its degree in $(\text{hct}_1, \cdots, \text{hct}_t)$ is $d$. Hence there is a degree $d$ function $f_1$ s.t.

$$\hat{f}_1((s^\otimes d + 1) \otimes (\text{hct}_1, \cdots, \text{hct}_t, 1)) := \hat{f}_1(s^\otimes d + 1, (\text{hct}_1, \cdots, \text{hct}_t)).$$

When having multiple functions $f = \{f_i\}$, one can perform homomorphic evaluation and decryption component-wise.

$$\text{HE.Dec}(s, \text{HE.Eval}(f_i, (\text{hct}_1, \cdots, \text{hct}_t))) = \hat{f}_{2,i}(\text{hct}_1, \cdots, \text{hct}_t) + \hat{f}_{1,i}(s^\otimes d + 1, (\text{hct}_1, \cdots, \text{hct}_t, 1)).$$

For convenience, below we often suppress the subscript $i$. By construction of \textsc{HE}, $\hat{f}_2$ is a degree $d$ polynomial over $\mathbb{Z}_{pd}$. For each component $\hat{f}_{2,i}$, its size and input-length are only a multiplicative \text{poly}(\lambda) factor larger than the size and input-length of $f_i$.
LWE distributions are {uniform distribution over variables}. Remark: semantic security holds as long as the binary LWE secret has sufficient entropy. More precisely, 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, hardness of LWE with binary secrets.

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

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

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

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 \((p_d \text{ here})\) 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 \( k, \ell, m, n, \beta, \gamma, \sigma, \) and \( p_d \) be integers, and let \( \Psi \) be a distribution (all parameterized by \( \lambda \)) such that \( \Pr_{x \leftarrow \Psi}[|x| \geq \beta] \leq negl(\lambda) \) and \( \sigma \geq \beta \gamma n m \). Further let \( \chi_\sigma \) be either the discrete Gaussian distribution with standard deviation \( \sigma \), or the uniform distribution over \([-\sigma, \sigma] \cap \mathbb{Z}\). Assuming that the \( LWE_{\ell,m,p_d,\Psi} \)-assumption holds, the weak and leaky \( LWE_{n,m,p_d,\chi_\sigma} \)-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 \( \chi_\sigma \), can be set to a polynomial, depending on \( n, \gamma, \) and \( \beta \) all bounded by fixed polynomials in \( \lambda \), 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 \( dFE = dFE_{N',S'} \) for degree \( d \) polynomials with appropriate \( N', S' \) set below, and the homomorphic encryptions scheme \( HE \) for constant degree computation above, our \( d+1FE = d+1FE_{N,S} \) scheme proceeds as follows:

- \( d+1FE.\text{Setup}(1^\lambda, pp) \): On input the security parameter \( 1^\lambda \) and public parameter \( pp \), generate a pair of \( dFE \) keys \((\text{dmpk, dmsk}) \leftarrow dFE.\text{Setup}(1^\lambda, pp)\), and output \( \text{mpk} = (pp, \text{dmpk}), \text{msk} = \text{dmsk} \).

- \( d+1FE.\text{Enc}(\text{mpk}, x) \): On input \( \text{mpk} = (pp, \text{dmpk}) \) and \( x \in \mathbb{Z}_p^N \), do:
  - Sample \( s \leftarrow HE.\text{KeyGen}(1^n, \text{hpp} = (p_{d+1}, p_d)) \), where \( n(\lambda) = \text{poly}(\lambda) \) is an appropriate polynomial in the global security \( \lambda \).
  - Encrypt \( \text{hct} \leftarrow \{\text{hct}_i = \text{HE.Enc}'(a_i, (a_i, s), x_i; e_i)\}_{i \in [N]}\).
  - Generate message \( X = (s^{d+1}, 1) \otimes (\text{hct}, 1) \), and encrypt \( \text{dct} \leftarrow dFE.\text{Enc}(\text{dmpk}, X) \).

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

*(Note that \( |\text{hct}| = \text{poly}(\lambda)N, X \in \mathbb{Z}_p^{N|X|}, \text{and } |X| = \text{poly}(\lambda)N.*)
• d+1FE.KeyGen(msk, f): On input msk = dmsk, and a degree \( d + 1 \) polynomial \( f \) over \( \mathbb{Z}_{p^{d+1}} \), output a key for the polynomial \( \tilde{f}_1(X) \) associated with \( f \) (Equation 3), \( \text{sk} \leftarrow \text{dFE.KeyGen}(\text{dmsk}, \tilde{f}_1) \).

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

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

\[
y = \hat{f}_2(\text{hct}) \oplus \text{dFE.Dec}(\text{sk}, \text{dct}).
\]

where \( \oplus \) 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 \( \text{dFE}^{N',S'} \) satisfies special-purpose \( (1 - \alpha) \)-sublinear compactness, that is,

\[
|\text{dct}| = \text{poly}(\lambda)(N' + S'^{1-\alpha}).
\]

Then, \( \text{d+1FE}^{N,S} \) constructed above also satisfies that

\[
|\text{ct}| = \text{poly}(\lambda)(N + S^{1-\alpha}).
\]

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

\[
|\text{hct}| = \text{poly}(\lambda)|x|,
\]

\[
|\text{dct}| = \text{poly}(\lambda)(N' + S^{1-\alpha}) = \text{poly}(\lambda)(N + S^{1-\alpha}),
\]

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

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

\[
\text{dFE.Dec}(\text{sk}, \text{dct}) = \left\lbrack \tilde{f}(X) + \sum_{d+1 \leq \beta \leq D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \leq \beta < d+1} p_\beta \tilde{Y}_\beta \right\rbrack_T,
\]

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

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

\[
\text{d+1FE.Dec}(\text{sk}, \text{ct}) = \left\lbrack f(x) + \sum_{d+2 \leq \beta \leq D} p_\beta \times \Lambda_\beta + \sum_{3 \leq \beta < d+2} p_\beta Y_\beta \right\rbrack_T,
\]

22
where again every $\Lambda_\beta$ is $(Bn_{\text{pad}})$-bounded, and every $Y_\beta$ is $(\overline{B} + B)n_{\text{pad}}$-bounded.

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

$$d+1\text{FE}.\text{Dec}(sk, ct) = \tilde{f}_2(hct) \oplus d\text{FE}.\text{Dec}(dsk, dct)$$

$$= \left[ \tilde{f}_2(hct) + \tilde{f}_1(X) + \sum_{d+1 \leq \beta \leq D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \leq \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T$$

$$= \left[ f(x) + e^f + \sum_{d+1 \leq \beta \leq D} p_\beta \times \tilde{\Lambda}_\beta + \sum_{3 \leq \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T$$

$$= \left[ f(x) + \sum_{d+2 \leq \beta \leq D} p_\beta \times \tilde{\Lambda}_\beta + p_{d+1}(Y_{d+1} = \tilde{\Lambda}_{d+1} + e^f) + \sum_{3 \leq \beta < d+1} p_\beta \tilde{Y}_\beta \right]_T.$$

We claim that $e^f$ is $(Bn_{\text{pad}})$-bounded, when $B$ and $n_{\text{pad}}$ are set to sufficiently large polynomial in $\lambda$. 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), $e^{f_i} = \sum_j e_{\text{mln},j}$, where $\text{mln}_{i,j}$ is the $j$'th monomial in function $f_i$, and $e_{\text{mln},j}$ is poly($\lambda$)-bounded, depending on the degree $d + 1$ and size of LWE noises from $\chi$.

Set $B = B(\lambda)$ to be larger than $e_{\text{mln},j}$ for maximum degree $D$ and noise distribution $\chi$.

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

Set $n_{\text{pad}}$ 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 $Bn_{\text{pad}}$-bounded.

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

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

- the following distributions $d\text{Real}$ (top) and $d\text{deal}$ (bottom) are $\mu$-indistinguishable,
and

\[
\begin{align*}
&\left\{ f \leftarrow \mathcal{F}_N, \ r \leftarrow \mathcal{R}, \ x \leftarrow \text{inp}(r) \ \\
&(\text{mpk}, \text{msk}) \leftarrow \text{dFE.Setup}(1^\lambda, \text{pp}) \ \\
&\text{sk} \leftarrow \text{dFE.KeyGen}(\text{msk}, \ f) \ \\
&\text{ct} \leftarrow \text{dFE.Enc}(\text{mpk}, \ x) \right\} : (r, (\text{mpk}, \text{sk}, \text{ct})) \\
&\left\{ (r_K, \text{st}) \leftarrow \text{dDSim}(f), \ r' \leftarrow \mathcal{R}|_{r_K; K}, \ x = \text{inp}(\bar{r}) \ \\
&\{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \leq \beta \leq D, k \in [n_{\text{pad}}]} \right\} \lambda \in \mathbb{N}
\end{align*}
\]

\[
\begin{align*}
f &\leftarrow \mathcal{F}_N, \ (r_K, \text{st}) \leftarrow \text{dDSim}(f), \ r' \leftarrow \mathcal{R}|_{r_K; K}, \ x = \text{inp}(\bar{r}) \ \\
\{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \leq \beta \leq D, k \in [n_{\text{pad}}]} \right\} \lambda \in \mathbb{N}
\end{align*}
\]

- with probability \(1 - \mu\), \(|K| \leq O(\lambda^{\frac{3}{2}})\).

More intuitively, the special-purpose simulation security states that honestly generated \((\text{mpk}, \text{sk}, \text{ct})\) of \(\text{dFE}\) can be simulated using i) the output \(f(\bar{x})\) “perturbed” by some “noise” \(\Delta_{\beta,k}\) (sampled from the \((\lambda^{\frac{3}{2}}, \mu)\)-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 \(\text{inp}(\mathcal{R})\), \(\bar{x}\) is sampled using \(\bar{r}\) that is partially fixed at locations \(K\) to \(r_K\), where \((r_K, K)\) are sampled from a distribution \(\text{dDSim}\) together with a state \(\text{st}\). Importantly, the set of fixed locations is small, bounded sublinear in \(\lambda\).

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

**Lemma 4.5 (Induction).** If for every polynomial \(N', S', \text{dFE}^{N', S'}\) satisfies \(O(\mu)\)-special-purpose simulation security, then for every polynomial \(N, S, \text{dFE}^{N, S}\) 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 \(\text{dDSim}\).

### 4.2.1 Proof of Special Purpose Simulation Security of \(\text{dFE}\)

Fix \(\text{dFE}^{N, S}, \mathcal{F}_N, \) and \(\text{inp}(\mathcal{R})\) as in Definition 4.4. We want to show that there exists correlated random variables \((r_K, K, \text{st})\) sampled by \(\text{dDSim}\), and an efficient simulator \(\text{dSim}\), such that, \(\text{dReal} (\text{top})\) and \(\text{dIdeal} (\text{bottom})\) are \(O(\mu)\)-indistinguishable, and

\[
\begin{align*}
f &\leftarrow \mathcal{F}_N, \ r \leftarrow \mathcal{R}, \ x = \text{inp}(r) \ \\
(\text{mpk}, \text{msk}) &\leftarrow \text{dFE.Setup}(1^\lambda, \text{pp}) \ \\
\text{sk} &\leftarrow \text{dFE.KeyGen}(\text{msk}, \ f) \ \\
\text{ct} &\leftarrow \text{dFE.Enc}(\text{mpk}, \ x) \right\} : (r, (\text{mpk}, \text{sk}, \text{ct})) \ \\
\left\{ (r_K, \text{st}) &\leftarrow \text{dDSim}(f), \ r' \leftarrow \mathcal{R}|_{r_K; K}, \ x = \text{inp}(\bar{r}) \ \\
\{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \leq \beta \leq D, k \in [n_{\text{pad}}]} \right\}
\end{align*}
\]

\[
\begin{align*}
\left\{ f &\leftarrow \mathcal{F}_N, \ (r_K, \text{st}) \leftarrow \text{dDSim}(f), \ r' \leftarrow \mathcal{R}|_{r_K; K}, \ x = \text{inp}(\bar{r}) \ \\
\{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+2 \leq \beta \leq D, k \in [n_{\text{pad}}]} \right\} \lambda \in \mathbb{N}
\end{align*}
\]
We show this via a sequence of hybrids $H_0, \cdots H_4$, where $H_0 = d+1\text{Real}$ and $H_4 = d+1\text{ideal}$ for some $d+1D_\text{Sim}$ and $d+1\Sim$ constructed along the way.

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

$$\left\{ \left( r, \ hct, \ mpk, \ dsk(f_1), \ dct(X) \right) \right\} \leftarrow \text{Samp}_0,$$

where $\text{Samp}_0$ does:

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

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

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

\[
\begin{align*}
\tilde{f}_1 &\leftarrow d\mathcal{F}_N, \quad \text{output } \tilde{f}_1 \text{ associated with } f \\
\text{dinp}(r, s, A) &\leftarrow (\text{dinp}(R))
\end{align*}
\]

\[
\begin{align*}
X &\leftarrow \text{dinp}(R) \\
(\text{mpk}, \text{dmsk}) &\leftarrow \text{dFE.Setup}(1^\lambda, \text{pp}) \\
dsk &\leftarrow \text{dFE.KeyGen}(\text{msk}, \tilde{f}_1) \\
dct &\leftarrow \text{dFE.Enc}(\text{mpk}, X)
\end{align*}
\]

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

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

\[
\begin{align*}
\tilde{f}_1 &\leftarrow d\mathcal{F}_N \\
\{ (R_{K'}, K', st'), \tilde{R}, \tilde{X} \} &\leftarrow \text{Samp}_1(\tilde{f}_1) \\
\{ \Delta_{\beta, k} \in \mathcal{X} \}_{d+1 \leq \beta \leq D, k \in [n_{\text{pad}}} &\leftarrow \tilde{R}, \ d\Sim \left( \begin{array}{c}
y' = \tilde{f}_1(\tilde{X}) + \sum_{d+1 \leq \beta \leq D} \sum_{k \in [n_{\text{pad}}} p_{\beta} \Delta_{\beta, k} \\
(R_{K'}, K', st'), \tilde{f}_1,
\end{array} \right)
\end{align*}
\]

\[2\text{We changed the order of components, without loss of generality.}\]
where $\text{Samp}_1(\tilde{f}_1)$ does:

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

Since $H_0$ can be reconstructed from $d\text{Real}$, it is $O(\mu)$-indistinguishable to the following hybrid $H_1$ reconstructed from $d\text{ideal}$.

**Hybrid $H_1$,** based on $d\text{ideal}$, samples as follows:

\[
\left\{ \begin{array}{l}
\tilde{f}_1 \leftarrow d\mathcal{F}_N \\
((R_{K'}, K', st'), \tilde{R}, \tilde{X}) \leftarrow \text{Samp}_1(\tilde{f}_1), \\
\{\Delta_{\beta,k} \leftarrow \mathcal{X}\}_{d+1 \leq \beta \leq D, k \in [n_{\text{pad}}]} \end{array} \right. 
\]

\[
\bar{r}, \text{hct, dSim} \left( \begin{array}{l}
y' = \tilde{f}_1(\tilde{X}) + \sum_{k \in [n_{\text{pad}}]} p_\beta \Delta_{\beta,k} \\
y = \tilde{f}_1(\bar{X}) + \sum_{d+1 \leq \beta \leq D} p_\beta \Delta_{\beta,k}
\end{array} \right)
\]

where $\bar{r}$ is contained in $\tilde{R}$ and $\text{hct}$ in $\tilde{X}$.

By property of $\text{HE}$, homomorphic evaluation of $f$ followed by decryption (Property 3) can be decomposed into

\[
\text{HE.Dec}(s, \text{HE.Eval}(f, \text{hct})) = f_2(\text{hct}) + \tilde{f}_1(\tilde{X}) = f(\bar{x}) + p_{d+1} e^f,
\]

where $e^f = \{e^f_i\}$ and $\forall i, e^f_i = \sum_j e^{mnl_{ij}}(\bar{x}, \bar{e})$

The second line follows from the additive and local structure of noise of $\text{HE}$ (Property 2).

By the same property, we also know that $e^{mnl_{ij}}(\bar{x}, \bar{e})$ depends only on variables $(\bar{x}_i, \bar{e}_i)$ s.t. $\bar{x}_i$ appears in $mnl_{ij}$. Thus, $e^{mnl_{ij}}$ has constant locality $2(d+1)$.

Therefore, $y'$ above can be written as:

\[
y' = f(\bar{x}) - f_2(\text{hct}) + p_{d+1} \left( e^f + \sum_{k \in [n_{\text{pad}}]} \Delta_{d+1,k} \right) + \sum_{d+2 \leq \beta \leq D} p_\beta \Delta_{\beta,k}
\]

where $\forall i, e^{f_i} + \sum_k \Delta_{d+1,k,i} = \sum_k (e^{mnl_{ik}} + \Delta_{d+1,k,i})$.

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

For every $k$, $\{e^{mnl_{ik}}\}$ is B-bounded. On the other hand, each $\Delta_{d+1,k}$ is a sample from the $(\mathcal{X}^2, \mu)$-flawed-smudging distribution $\mathcal{X}$ for exactly $B$-bounded distributions. This means, there are randomized predicates $\{\text{BAD}_{k,i}\}$, such that, the following two distributions are
Consider the by construction of except in the sums to distribution constant locality, we have that on. In addition, with probability \(1 - \mu\), \(|\text{bad}|_1 = O(\lambda^2)\). Since both \(e^{\text{ml},i,k}\) and \(\text{inp}\) have constant locality, we have that \(|I| = O(\lambda^2)\).

**Hybrid H2:** Apply the above equality of distribution to hybrid \(H_1\), we have that it is identical to distribution \(H_2\) below, where \(\tilde{r}, \tilde{e}\) are replaced with \(\tilde{r}, \tilde{e}\) as sampled above everywhere, except in the sums \(\{e^{\text{ml},i,k} + \Delta_{d+1,k,i}\}_{k,i}\).

\[
\tilde{f}_1 \leftarrow \mathcal{DFN},
\]

Same sampling as in \(\mathcal{D}_1(\tilde{f}_1)\)

\[
\begin{align*}
((R_{K''}, K'', \text{st}'', \tilde{R}, \tilde{X}) & \leftarrow \text{Samp}_1(\tilde{f}_1)_{|_{(\tilde{r}, \tilde{e}), I}} \quad : (\tilde{r}, \tilde{e}), \tilde{hct}, \text{dSim}\left((R_{K''}, K'', \text{st}'', \tilde{R}, \tilde{X}) \leftarrow \text{Samp}_1(\tilde{f}_1)_{|_{(\tilde{r}, \tilde{e}), I}} \quad : (\tilde{r}, \tilde{e}), \tilde{hct}, \text{dSim}\left((R_{K''}, K'', \text{st}'', \tilde{R}, \tilde{X}) \right)
\end{align*}
\]

\[
\begin{align*}
\Delta_{i,k} & \leftarrow \mathcal{X}_{d+2\leq\beta\leq D, k\in[m_{\text{pad}]}}
\end{align*}
\]

where \(\tilde{y} = f(\tilde{x}) - \tilde{f}_2(\tilde{hct}) + \sum_{d+2\leq\beta\leq D, k\in[m_{\text{pad}]}} \beta \Delta_{i,k}\)

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

1. Sample \((R_{K''}, K'', \text{st}'') \leftarrow d\text{Sim}(\tilde{f}_1)\)
2. Sample \(\tilde{R} \leftarrow (R|_{R_{K''}, K''})_{|_{(\tilde{r}, \tilde{e}), I}}\). Note that the bit fixing distribution simply composes, requiring sampling \(\tilde{R} = (\tilde{r}, \tilde{e}, \tilde{s}A, A, \tilde{s})\) at random conditioned on \(R_{K''} = R_{K''}\) and \((\tilde{r}, \tilde{e}) = (\tilde{r}, \tilde{e})\). Since \(\mathcal{R}\) is bit-fixing efficient, this sampling can be done efficiently.
3. Compute \(\tilde{x} = \text{inp}(\tilde{r})\), and \(\tilde{X} = \text{dinp}(\tilde{R}) = (\tilde{s}^{d+1}, 1) \odot (\tilde{hct}, 1)\), where \(\tilde{hct} = \{\text{HE.Enc}'(\tilde{a}_i, (\tilde{a}_i, \tilde{s}), \tilde{x}_i, \tilde{c}_i)\}\).

Consider the HE ciphertexts in two cases:

- **Case 1:** \(\tilde{hct}_i = \text{HE.Enc}'(\tilde{a}_i, (\tilde{a}_i, \tilde{s}), \tilde{x}_i, \tilde{c}_i)\) has that \(i\) \(\tilde{a}_i\) is fixed in \(R_{K''}\), or \(ii\) \(\tilde{a}_i, \tilde{s}\) is fixed in \(R_{K''}\), or \(iii\) \(\tilde{c}_i\) is fixed in \((\tilde{r}, \tilde{e})\).

Let \(S = S(K'', I)\) denote the indices of these ciphertexts, efficiently computable from \(K''\) and \(I\). We have that \(|S| \leq |K''| + |I|\). With probability \(1 - O(\mu)\), both \(|K''|\) and \(|I|\) are bounded by \(O(\lambda^2)\), and so is \(|S|\).
• Case 2: for every $i \notin S$, $\widetilde{hct}_i$ is generated using randomly sampled $\tilde{a}_i$ and $\tilde{e}_i$, and $\langle \tilde{a}_i, \tilde{s} \rangle$ is not leaked nor fixed.

Intuitively, ciphertext in $S$ are generated using fixed values, and are potentially not hiding. However, since there are only a few $O(\lambda^2)$ of them, even if all information used for generating them are revealed, the secret key $\tilde{s}$ still have high entropy. Therefore, by robustness of HE, ciphertexts in $\neg S$, generated using random $\tilde{a}_i$ and $\tilde{e}_i$ and entropy $\tilde{s}$, remain semantically secure. This yields the next hybrid $H_3$.

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

$$
\begin{cases}
\tilde{f}_1 \leftarrow d\mathcal{F}N, \\
\text{Same sampling as in $D_1(\tilde{f}_1)$} \nonumber \\
((R_{K''}, K'', st''), \tilde{R}, \tilde{X}) \leftarrow \text{Samp}_1(\tilde{f}_1)_{|_{(\bar{f}, \tilde{e})}, I} : \\
S = S(K'', I) \\
\{\Delta_{\beta, k} \leftarrow \mathcal{X}\}_{d+2\leq \beta \leq D, k \in [n_{pad}]} \\
\tilde{r}, \langle \widetilde{\text{hct}}_S(\tilde{x}_S)\widetilde{\text{hct}}_{\neg S}(0) \rangle, \text{dSim}((R_{K''}, K'', st''), \tilde{f}_1, \tilde{y}) \bigg),
\end{cases}
$$

where

$$
\widetilde{\text{hct}}_S(x_S) = \{\widetilde{\text{hct}}_i = \text{HE.Enc}'(\tilde{a}_i, \langle \tilde{a}_i, \tilde{s} \rangle, \tilde{x}_i, \tilde{e}_i)\}_{i \in S},$$

$$
\widetilde{\text{hct}}_S(0) = \{\text{hct}_i = \text{HE.Enc}'(\tilde{a}_i, \langle \tilde{a}_i, \tilde{s} \rangle, 0, \tilde{e}_i)\}_{i \notin S},$$

$$
\tilde{y} = f(\tilde{x}) - \hat{f}_2(\langle \widetilde{\text{hct}}_S(\tilde{x}_S)\widetilde{\text{hct}}_{\neg S}(0) \rangle) + pd+1 \left\{ \sum_k (e^{mnl}_{ik} + \Delta_{d+1,k,i}) \right\} + \sum_{d+2 \leq \beta \leq D} \sum_{k \in [n_{pad}]} p_{\beta} \Delta_{\beta, k}.
$$

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

Note that $H_3$ and $H_4$ make the same potentially *inefficient* sampling steps, namely, sampling from $D_1$, $dD_{\text{Sim}}$ (first step in $\text{Samp}_1|_{(\bar{f}, \tilde{e})}, I$), and $\mathcal{X}$. The other sampling and computing steps are all efficient, including sampling from $(R|_{R_{K''}, K''})_{|_{(\bar{f}, \tilde{e})}, I}$ and computing $\tilde{x}, \tilde{X}$ (the second and third step in $\text{Samp}_1|_{(\bar{f}, \tilde{e})}, I$), and computing $S$.

Let $T$ be the set of possible values sampled from $D_1$, $dD_{\text{Sim}}$ and $\mathcal{X}$. We know that for an $1 - O(\mu)$ fraction of $T$, it holds that $|K''|$ and $|I|$ are bounded by $O(\lambda^2)$. Take any $t \in T$ satisfying this. $H_3$ and $H_4$ conditioned on $t$ being sampled are efficient. Furthermore, $\tilde{s}$ has high entropy conditioned on $t$, $\tilde{r}$, and ciphertexts in $S$, $H_\infty \left( \tilde{s} \mid t, \tilde{r}, \text{hct}_S \right) \geq n - O(\lambda^2)$.

Observe that $t, \tilde{r}$ contains information for computing the entire output of $H_3$ and $H_4$ except for $\text{hct}_S(x_{\neg S})$ in $H_3$ and $\text{hct}_{\neg S}(0)$ in $H_4$. Conditioned on $t$, by the robustness of
HE, these ciphertexts are $\mu$-indistinguishable. Therefore, conditioned on $t$, $H_3$ and $H_4$ are $\mu$-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 $d+1\text{Ideal}$ for $d+1\text{FE}$ scheme.

**Hybrid $H_4$** is the $d+1\text{Ideal}$ distribution with $d+1\text{DSim}$ and $d+1\text{Sim}$ specified below.

$$
\left\{ \begin{array}{l}
1. \text{ Sample } \left( (\tilde{r}, \tilde{e}), \left\{ e^{\text{mnl},k} + \Delta_{d+1,k,i} \right\}_{k,i}, \text{bad} \right) \leftarrow \mathcal{D}_1(\tilde{f}_1), \text{ where bad determines } I.
2. \text{ Sample } (R_{K''}, K'', \text{st}'') \leftarrow \text{dDSim}(\tilde{f}_1).
3. \text{ Compute } S = S(K'', I) \text{ and let } S' \text{ be the set of indices of randomness that determine inputs at location } S, \text{ that is, } u_S = \text{inp}(v_S).
4. \text{ Sample } (\hat{R} = (\tilde{r}, \tilde{e}, \tilde{s}, \tilde{A}, \tilde{s})) \leftarrow (\mathcal{R}|_{R_{K''}, K''})|_{(\tilde{r}, \tilde{e}), I}.
5. \text{ Set } K \text{ to be the set of indices in } \tilde{r} \text{ that are in } (S', K'', I), \text{ set } r'_K = \tilde{r}_K, \text{ and set st to include } (\tilde{e}, \tilde{s}, \tilde{A}, \tilde{s}), \left\{ e^{\text{mnl},k} + \Delta_{d+1,k,i} \right\}_{k,i}, (R_{K''}, K'', \text{st}'').
\end{array} \right.
$$

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

1. It generates $\left( \text{hct}_S(\tilde{x}_S)\text{hct}_S(0) \right)$ by first computing $\tilde{x}_S = \text{inp}(r'_S)$, generating ciphertexts $\text{hct}_S(\tilde{x}_S)$ and $\text{hct}_S(0)$ using $(\tilde{e}, \tilde{s}, \tilde{A}, \tilde{s})$ contained in st.
2. It generates $d\text{Sim} \left( (R_{K''}, K'', \text{st}''), \tilde{f}_1, \tilde{y} \right)$, using $(R_{K''}, K'', \text{st}'')$ contained in st and 

$$
\tilde{y} = y - \tilde{f}_2 \left( \left( \text{hct}_S(\tilde{x}_S)\text{hct}_S(0) \right) \right) + p_{d+1} \left\{ \left. \sum_k \left( e^{\text{mnl},k} + \Delta_{d+1,k,i} \right) \right|_i \right\}
$$

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

By construction of $d+1\text{DSim}$ and $d+1\text{Sim}$, hybrids $H_3$ and $H_4$ are identically distributed.

By a hybrid argument, we have that $d+1\text{Real}$ and $d+1\text{Ideal}$ are $O(\mu)$-indistinguishable, which concludes that $d+1\text{FE}$ 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 $\mathcal{D}_{\text{DSim}}$ when using such distributions.

**Special-Purpose Function Distribution** We will use DFE for function distribution $\mathcal{F}N$ that samples $f$ of form:
• \( f[G] : \{0,1\}^n \rightarrow \{0,1\}^n \) is defined by \(i\) a fixed collection of predicates \( \{g_i\}_{i \in [n]} \), one for every output bit, and \(ii\) an input-output dependency graph \( G \leftarrow G \) sampled from a distribution \( G \).

\[ \forall (x, \text{seed}) \in \{0,1\}^n, \quad f_i(x, \text{seed}) = g_i(x, \text{seed}_{G(i)}). \]

We refer to the second input as the seed.

• \( f[G] \) has constant locality \( \ell \) 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 \( \text{inp}(\mathcal{R} \times \mathcal{U}) = \mathcal{R} \times \mathcal{U} \) (i.e., \( \text{inp} \) 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 \( \text{DFE} \) implies the existence of \( \text{DDSim} \) and \( \text{Sim} \), such that, the real \( \text{DReal} \) distribution is indistinguishable to the following ideal \( \text{Dideal} \) distribution.

\[
\text{Dideal} = \left\{ \begin{array}{l}
\text{f} \leftarrow \mathcal{F}N, \\
\text{(x, seed)}_K \leftarrow \mathcal{DDSim}(f), \\
\text{(x, seed)} \leftarrow \mathcal{R}_{(x, \text{seed})_K, K}, \\
\text{(x, seed), DSim} \left((\text{(x, seed)}_K, K, st), f, y = f(\text{(x, seed)}) \right) \end{array} \right\}_{\lambda \in \mathbb{N}}
\]

With probability \( 1 - O(\mu), |K| = O(\lambda^2) \). Let \( K_{\text{seed}} \) be the random variable representing the indexes of seed bits seed\( _{\text{seed}} \) that are fixed in \( (x, \text{seed})_K \).

The following lemma says that the positions \( K_{\text{seed}} \) 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 \( K_{\text{seed}} \). This property will be very instrumental later in our construction of \( \text{FE} \) for \( \text{NC}_1 \).

**Lemma 4.6.** For every \( \lambda \), every superpolynomially small \( \mu = \lambda^{-\omega(1)} \), every special-purpose distribution \( \mathcal{F}N \), and every \( \mathcal{R} \), there exists a random variable \( \mathcal{K} \), such that, \(i\) \( \mathcal{K} \) is correlated with \( (f[G] \leftarrow \mathcal{F}N, (x_K, K, st) \leftarrow \mathcal{DDSim}(f[G]')) \), and \(ii\) \( \mathcal{K} \) is independent of \( G \), \(iii\) \( K_{\text{seed}} \subseteq G(\mathcal{K}) \) and \( |\mathcal{K}| = O(\lambda^2) \) with probability \( 1 - O(\sqrt{n}) \).

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

By construction, the function \( f^d \) that \( \mathcal{DDSim} \) evaluates is determined by \( f^{d+1} \) that \( \mathcal{DSim}^{d+1} \) evaluates. In particular, \( f^d = f_1^{d+1} \), operating on \( (s^{\otimes d+1}, 1) \otimes (\text{hct}, 1) \).

If \( f_1^{d+1}(x^{d+1}, \text{seed}^{d+1}) = g_i^{d+1}(x^{d+1}, \text{seed}^{d+1}_{G_{d+1}^{i+1}}) \), for fixed \( g_i^{d+1} \)

and distributional \( G^{d+1} \).

Then \( f_1^{d}((s^{\otimes d+1}, 1) \otimes (\text{hct}, 1)) = h_i^{d+1}((s^{\otimes d+1}, 1) \otimes (\text{hct}(x^{d+1}), \text{hct(\text{seed}^{d+1}_{G_{d+1}^{i+1}(i)})), 1)) = h_i^{d}(x^{d+1}, \text{seed}^{d+1}_{G_{d+1}^{i+1}(i)}) \), for fixed \( h_i^{d} \).

30
The last equality simply rewrites the second last line as evaluating a randomized function $h^d_i$ on $x^{d+1}$, seed$_{G^{d+1}(i)}$ (that samples other variables such as $s$, $A$, and $e$ that $g^{d+1}_i$ uses internally). In other words, $f_i^d$ depends ("recursively") only on seed bits seed$_{G^{d+1}(i)}$. Thus, BAD$_{k,i}^d$ can also be written as a fixed randomized predicate depending only on seed bits seed$_{G^{d+1}(i)}$. Given that the starting-point function $f_i^0 = g_i(x, \text{seed}_{G(i)})$ indeed has this structure, we have that every BAD$_{k,i}^d$ can be written as a fixed randomized predicate $P_{k,i}^d(x, \text{seed}_{G(i)})$, 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 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$_{k,i}^d(x, \text{seed}_{G(i)} ; r)$ 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.

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

For any $x, r$, let $E_{K,i}^d = \mathbb{E}[P_{k,i}^d(x, \text{seed}_{G(i)} ; r)]$ be the expectation of P$_{k,i}^d$ over the choice of seed and $G$. Observe that since $G$ has locality $\ell$ and the marginal distribution of seed$_{G(i)}$ is uniform, the expectation $E_{K,i}^d$ is either zero if P$_{k,i}^d(x, * ; r)$ is a zero predicate, or at least $1/2^\ell$. Let $K$ be the set of $i$ s.t., for some $d$ and $i$, P$_{k,i}^d(x, * ; r)$ is non-zero and $E_{K,i}^d \geq 1/2^\ell$. 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 $K$ satisfy the conditions in the lemma statement: i) $K$ is correlated $f$, ((x, seed)$_K$, K, st) $\leftarrow$ D$_{\text{Sim}}$, but ii) $K$ is independent of $G$ (it depends on fixed randomized predicates P$_{k,i}^d$, their randomness $r$, and input $x$). iii) $K_{\text{seed}} \subseteq G(K)$ as only non-zero predicate P$_{k,i}^d$ can lead to fixed seed bits seed$_{G(i)}$; in addition, with probability $1 - O(\sqrt{\mu})$, $|K| = O(2^\ell \lambda^{\varepsilon_2})$. \hfill $\Box$

### 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 $x_1, \ldots, x_\lambda$ 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 \((t, \mu)\)-bit-fixing homomorphic sharing scheme \( BF = (\text{BFsetup}, \text{BFshare}, \text{BFeval}, \text{BFdec}) \) consists of four efficient algorithms satisfying the following.

**Syntax:** The four algorithms have the following syntax

- \( \text{BFsetup}(1^\lambda, 1^s, 1^d) \) is a randomized algorithm that, on input a security parameter \( \lambda \) and bounds \( s \) and \( d \) on the function size and depth, outputs a CRS \( crs \).
- \( \text{BFshare}(crs, v) \) is a randomized algorithm that on input \( crs \) and \( v \in \{0, 1\}^{\text{poly}(\lambda)} \), outputs input shares \( x_1, \ldots, x_\lambda \) of \( v \). Let \( n = n(\lambda) \) be the length of all input shares.
- \( \text{BFeval}(crs, x_i, i, f) \) 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, \( \text{BFeval} \) is invoked separately for each output bit. The collection of output shares form \( o_i \).
- \( \text{BFdec}(crs, \{o_i\}, f) \) 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 \( d \),

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

\((t, \mu)\)-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 \( \text{poly}(\lambda) \)-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 \( A \),

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

\footnote{Formally, we should only quantify over \( x^* \) and \( J \) for which \( \text{BFshare}(crs, v_b) |_{x^*, J} \) is a valid distribution, i.e., for which the support of \( \text{BFshare}(crs, v_b) \) 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 ($x_1, \ldots, x_\lambda$) 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:

- BFsetup($1^\lambda, 1^s, 1^d$) runs params $\xleftarrow{\$}$ MFHE.Setup($1^\lambda, 1^d$) and outputs crs := params.
- BFshare(crs, $v$) on input $v \in \{0, 1\}^{\text{poly}(\lambda)}$, computes the following:
  - $(pk_i, sk_i) \xleftarrow{\$} \text{MFHE.KeyGen}(\text{crs})$ for $i \in [\lambda],$
  - an additive sharing $ss_1, \ldots, ss_\lambda$ of $v$, i.e., sample uniform bit strings $ss_1, \ldots, ss_{\lambda-1}$ of length $\lvert v \rvert$ each and set $ss_\lambda = v \oplus \bigoplus_{i=1}^{\lambda-1} ss_i,$
  - $ct_i \xleftarrow{\$} \text{MFHE.Enc}(pk_i, ss_i)$ for $i \in [\lambda],$
  - $\hat{ct}_i \xleftarrow{\$} \text{MFHE.Expand}((pk_1, \ldots, pk_\lambda), i, ct_i)$ for $i \in [\lambda].$

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

$$x_i := \left((\xleftarrow{\$}t_j, pk_j)_{j \in \lambda}, sk_i, K_i\right).$$

- BFeval(crs, $x_i, i, f$) on input crs, $x_i = \left((\xleftarrow{\$}t_j, pk_j)_{j \in \lambda}, sk_i, K_i\right)$, $i$, and a function $f$ with $\ell$ output bits, where $f_j$ denotes the function computing the $j$th output bit, computes for all $j \in [\ell]$: 
  - $\hat{ct}^{(j)} = \text{MFHE.Eval}(\text{crs}, f'_j, \hat{ct}_1, \ldots, \hat{ct}_\lambda)$, with $f'_j(ss_1, \ldots, ss_\lambda) := f_j(\bigoplus_{i=1}^{\lambda} ss_i) = f_j(v),$
  - $r_{i,j} = \text{PRF}(K_i, j),$
  - $o_{i,j} = \text{MFHE.PartDec}(\hat{ct}^{(j)}, (pk_1, \ldots, pk_\lambda), i, sk_i; r_{i,j})$, which means $r_{i,j}$ is used as the randomness for the algorithm MFHE.PartDec.

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

- BFdec(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 
  $$y_j = \text{MFHE FIN Dec}(o_{i,j}, \ldots, o_{\lambda,j})$$ for $j \in [\ell]$, and outputs $y = (y_1, \ldots, y_\ell).$

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 $\mu$-semantically secure, has $\mu$-simulatability of partial decryptions, and PRF is $\mu$-pseudorandom, then BF is $(t, O(L \cdot \mu))$-bit-fixing secure for $t \leq (\lambda - 1)/2$, 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($\lambda$)-bit inputs, let $x^* \in \{0,1\}^d$, 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 $\ell \leq L$ output bits and $f(v_0) = f(v_1)$, and let $A$ be a PPT adversary. Now let $H_0$ be the $(t, O(L \cdot \mu))$-bit-fixing-security experiment, i.e., $b \leftarrow \{0,1\}, \text{crs} \leftarrow$
We then have that the advantage of \( \text{Adv} \) will make use of the following building blocks w.r.t. \( \text{sk} \) without \( \text{pk} \). Consider the attacker •

\[
\text{Sim}_{\text{PRF}}(\lambda) \leftarrow \text{BFsetup}(\text{crs}, v_b) \leftarrow \text{BFshare}(\text{crs}, v_b) \leftarrow \text{BFeval}(\text{crs}, x_i, i, f),
\]

\( b' \leftarrow A(\text{Binary}(x) \cdot J, \{x_i\}_{i \in K}, \{a_i\}_{i \in [\lambda]}), \) and let the output of \( \text{Sim} \) be 1 if \( b' = b \), and 0 otherwise. Note that since \( |K| \leq t \) and \( x^t \) 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 \( \text{BFeval}(\text{crs}, x_{i_0}, i_0, f) \) gets replaced by a truly random value. Let \( \text{A}_{\text{PRF}} \) be an adversary with access to an oracle, which is either \( \text{PRF} \) or a truly random function, and let \( \text{A}_{\text{PRF}} \) emulate the experiment toward \( A \), where instead of running \( r_{i_0,j} = \text{PRF}(K_{i_0}, j) \), \( \text{A}_{\text{PRF}} \) 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 \( \text{A}_{\text{PRF}} \) output 1 if \( A \) guesses \( b \) correctly, and 0 otherwise. If the oracle of \( \text{A}_{\text{PRF}} \) is \( \text{PRF} \), 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 \( \mu \)-indistinguishability of \( \text{PRF} \) 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 \( \text{Sim} \) from \( \mu \)-simulatability of partial decryptions, where \( \text{Sim} \) is given \( i_0 \), all secret keys except \( \text{sk}_{i_0} \), \( \text{ct}_{i_0} \), 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 \( \text{BFshare} \) encrypts \( \bot \) instead of \( \text{ss}_{i_0} \) to obtain \( \text{ct}_{i_0} \). Consider the attacker \( A_{\text{sem}} \) against the semantic security of \( \text{MFHE} \) that is given \( \text{params} \), a public key \( \text{pk} \), and an encryption \( \text{ct} \) of either \( \text{ss}_{i_0} \) or \( \bot \), and then emulates \( H_{2,\ell} \) or \( H_3 \) by using \( \text{pk} \) as \( \text{pk}_{i_0} \) and instead of encrypting \( \text{ss}_{i_0} \) or \( \bot \), it uses its input \( \text{ct} \). Note that this can be done without \( \text{sk}_{i_0} \) since it is only used in the scheme to compute \( \{o_{i_0,j}\}_j \) and these are simulated in both hybrids without \( \text{sk}_{i_0} \). Finally let \( A_{\text{sem}} \) output 1 if \( A \) guesses \( b \) correctly, and 0 otherwise. We then have that the advantage of \( A_{\text{sem}} \) 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

For any fixed logarithmic function \( \text{Dep}(\lambda) = O(\log \lambda) \), we construct a family of FE schemes \( \{\text{FE}^{N,S,\text{Dep}}\} \) for computing the class of NC circuits \( C \) with depth \( \text{Dep} = \text{Dep}(\lambda) \), 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. \( \alpha \in (0,1/2), \varepsilon \in (0,1), \) and \( \mu = 2^{-o(\lambda^2)} \).

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

\[
\forall i, \text{PRG}_i(\text{seed}) = P(\text{seed}_{G(i)}).
\]

Moreover, all input nodes in the graph have degree bounded by \( o(S^{1-\alpha}) \).
• A \((O(\lambda^{2}), \lambda^{o(1)})\)-bit-fixing homomorphic sharing BF = (BFsetup, BFshare, BFeval, BFdec).

• The AIK randomized encoding [AIK04] RE = (REnc, REval) whose encoding algorithm for any function \(f\), \(REnc(f, x ; r)\) 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 \(\text{DFE}^{N', S'}\) = (DFE.Setup, DFE.Enc, DFE.KeyGen, 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 \(\text{FE}^{N, S}\) for \(\text{NC}_1\) computation. We in-line the analysis of its correctness and \((1 - \alpha)\)-sublinear compactness in italic font.

• \text{FE.Setup}(1^{\lambda})\): Generate a pair of DFE keys \((\text{Dmpk}, \text{Dmsk}) \leftarrow \text{DFE.Setup}(1^{\lambda}, pp)\), using public parameter \(pp = (p_2, p_3, \ldots, p_D, s_2, \ldots, s_D, n_{\text{pad}})\), 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, \(p_4\) is the HE ciphertext space modulus when the message space has modulus \(p_{d+1}\), and \(p_0 < \cdots < p_3 < \text{poly}(\lambda) \ll p_2\), \(s_d = \text{poly}(\lambda)\), and \(n_{\text{pad}} = \text{poly}(s_2)\).) Generate a CRS for the bit-fixing homomorphic sharing scheme \(\text{crs} \leftarrow \text{BFsetup}(1^{\lambda})\).

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

• \text{FE.KeyGen}(\text{msk}, C)\): Input circuit \(C\) has input-length \(N\), size \(S\), and depth \(\text{Dep}\). 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 \(I_1, \ldots, I_M\), where chunk \(I_j\) includes outputs bits \(((j - 1)S/M + 1, \ldots, jS/M)\). 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 \(x_{i,j}^j\) of \(v\), homomorphically evaluates \(C_{I_j}\)
      \[D_{i,j}^j(x_{i,j}^j) = \text{BFeval}(\text{crs}, x_{i,j}^j, i, C_{I_j}) = a_{i,j}^j.\]

Since \(\text{BFeval}\) 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 \(x_{i,j}^j\) is also bounded by \(\text{poly}(\lambda, s)\).

  * Choose a random permutation \(\pi: [\lambda] \times [M] \times [\phi] \rightarrow [\lambda M \phi]\). For every \(j \in [M]\) and \(i \in [\lambda]\), let \(f_{i,j}^j\) be the following function
    \[f_{i,j}^j(x_{i,j}^j, \text{seed}) = \text{REnc}(D_{i,j}^j, x_{i,j}^j ; \text{PRG}_{\Pi_i^j}(\text{seed}))\].

---

\(^{4}\)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 $\text{poly}(\lambda, s)S/M = \text{poly}(\lambda)S/M$ and hence uses at most $\phi = \text{poly}(\lambda)S/M$ random coins. $\text{PRG}_{\Pi_i^j}(\text{seed})$ contains $\phi$ PRG output bits at locations $\{\pi(i, j, k)\}_{k \in [0]}$ determined by the random permutation $\pi$. Overall, $|\text{PRG}(\text{seed})| = \text{poly}(\lambda, s)S = \text{poly}(\lambda)S$ and $|\text{seed}| = \text{poly}(\lambda)S^{1-\alpha}$.

Finally, set

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

The input length and size of $f$ is $N = |\{x_i^j\}| + |\text{seed}| = \text{poly}(\lambda, s)\lambda M + \text{poly}(\lambda)S^{1-\alpha} = \text{poly}(\lambda)S^{1-\alpha}$ and $S' = |f_i^j|\lambda M = \text{poly}(\lambda)S$. Since $\text{REnc}$ has constant locality $1 + 3\ell$, where $\ell$ is the locality of $\text{PRG}$, $f$ 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 $G_x$ sampled from some distribution $G$. To see this, recall that every output bit $i$ of $\text{PRG}$ is computed using the same predicate $P$ on a subset of seeds $G(i)$. Thus, $f_i^j$ can be written as

$$f_i^j(x_i^j, \text{seed}) = \text{REnc}(D_i^j, x_i^j; \{P(\text{seed}_{G(\pi(i,j,k))})\}_{k \in [0]})$$

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

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

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

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

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

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

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

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

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

where every $Y_\beta$ is $(B + B)n_{\text{pad}}$-bounded, and $z$ is bounded by $B = 1 + (\sum_{3 \leq \beta \leq D} p_\beta)(B + B)n_{\text{pad}} = \text{poly}(\lambda)$.)
For every $z' \in [-B, B] \cap \mathbb{Z}$, 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 ⊥.

(Correctness analysis, continued: we have that)

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

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

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

\[
\begin{align*}
\sigma_i^j &= \text{REval}(y_i^j) = D_i^j(x_i^j) = \text{BFeval}(\text{crs}, x_i^j, i, C_{I_i}) = \sigma_i^j .
\end{align*}
\]

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

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

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

Next, we show that $\text{FE}$ satisfies 1-key fully-selective indistinguishability-security.

**Theorem 5.4.** For any $\mu$ and $\varepsilon_2$ such that $\mu = 2^{-o(\lambda^{\varepsilon_2})}$. Assume that $\text{PRG}$ satisfies $\mu$-indistinguishability, $\text{BF}$ satisfies $(O(\lambda^{\varepsilon_2}), \lambda^{o(1)} \mu)$-bit-fixing security, $\text{DFE}^{N_1, S_1}$ satisfies $O(\mu)$-special-purpose simulation security. Then, $\text{FEN}_{S_1}^{N, \text{Dep}}$ above satisfies $O(\mu)$-Full Sel Ind security.

**Proof.** Fix any $\text{NC}_1$ circuit $\{C\}_\lambda$ with input-length $N$, size $S$, depth $\text{Dep}$, 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 $\lambda$ and $\mu = \mu(\lambda)$.

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

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

**Hybrid $H_0$:** This hybrid generates $\text{view}_A$ honestly as above, and additionally outputs $X$ encrypted by $\text{DFE}$.

\[
\text{Real} = \left\{ \begin{array}{l}
b \leftarrow \{0, 1\} \\
(\text{mpk}, \text{msk}) \leftarrow \text{FE.Setup}(1^\lambda, \text{pp}) \\
\text{sk} \leftarrow \text{FE.KeyGen}(\text{msk}, f) \\
\text{ct} \leftarrow \text{FE.Enc}(\text{mpk}, \text{ct})
\end{array} \right\}_\lambda
\]

Output $\text{view}_A = (\text{mpk}, \text{sk}, \text{ct})$.
By construction of $\text{DFE}$,

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

where every $x^j$ is a random and independent sharing of $v_b$.

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

**Hybrid $H_1$:** This hybrid samples $\text{view}_A$ using the simulator of $\text{DFE}$ as follows:

$$\text{Ideal} = \left\{ f \leftarrow \mathcal{F}_N, \ (X_K, K, \text{st}) \leftarrow \mathcal{D}_{\text{Sim}}, \ X \leftarrow \mathcal{R}|_{X_K,K}, \ {\{x^j\}}, \ \text{view}_A = \mathcal{D}_{\text{Sim}}((X_K, K, \text{st}), f, y = f(\bar{X})) \right\}_X,$$

where

$$\bar{X} = (\{x^j\}, \text{seed}) , \ y^j = \text{REnc}(D^j, \bar{x}^j; \text{PRG}_{s(j,\mu)}(\text{seed})) ,$$

where with probability $1 - O(\mu)$, $|K| \leq O(\lambda^2)$.\(^5\)

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

$$\mathcal{R}|_{X_K,K} = (\{x\} \leftarrow \mathcal{D}_x|x_{K_x,K_x}) \times (\text{seed} \leftarrow U|_{\text{seed}_{K_{\text{seed}},K_{\text{seed}}}}) ,$$

$$\mathcal{D}_x : b \leftarrow \{0, 1\}, \ (x = \{x^j\} \leftarrow \text{BFshare}(\text{crs}, v_b, 1^s, 1^{\text{Dep}}), \ \text{output } x ,$$

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

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

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

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

$$\text{PRG}_{\neg K_{\text{prg}}}(\text{seed}) \approx_\mu U .$$

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

---

\(^5\)Applying the special-purpose simulation security of $\text{DFE}$ directly would imply the indistinguishability of $\text{Real}$ and $\text{Ideal}$ where the distributions output the entire $X$ and $\bar{X}$. Since they contain $\{x^j\}$ and $\{\bar{x}^j\}$ 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 \(\Pi^j_i = \{\pi(i, j, k)\}_{k \in \phi}\). Let \(K_{\text{re}}\) denote the set of \((i, j)\) s.t. \(y_i^j\) depends on \(\text{PRG}_{\text{prg}}(\text{seed})\), that is,

\[
K_{\text{re}} := \{(i, j) : \Pi^j_i \cap K_{\text{prg}} \neq \emptyset\}
\]

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

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

\[
\begin{align*}
\{ y_i^j = \text{REnc}(D_i^j, \bar{x}_i^j; \text{PRG}_{\pi(j,i)}(\text{seed})) \}_{(i,j) \in K_{\text{re}}} \\
\{ \hat{y}_i^j \leftarrow \text{RSim}(\bar{y}_i^j = D_i^j(\bar{x}_i^j)) \}_{(i,j) \in \neg K_{\text{re}}}.
\end{align*}
\]

where the output \(\bar{y}_i^j\) is the output share obtained by homomorphically evaluating circuit \(C_{f_i}\) 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 \(\text{view}_A\) identically to \(H_1\), except that randomized encodings at location \(K_{\text{re}}\) are now simulated.

\[
\begin{align*}
&\{ f \leftarrow \mathcal{F}\mathcal{N}, \\
&\quad (X_K, K, \text{st}) \leftarrow \mathcal{D}_{\text{Sim}}(f) \\
&\quad \bar{X} \leftarrow \mathcal{R}_{\bar{X}|X_K, K} \\
&\quad \{\bar{x}_i^j, \text{view}_A = \mathcal{D}_{\text{Sim}}(X_K, K, \text{st}), f, y' = \left\{ y_i^j (i,j) \in K_{\text{re}} \\
&\quad \{ \hat{y}_i^j (i,j) \in \neg K_{\text{re}} \} \right\}_\lambda, \\
&\quad x_i = (\{\bar{x}_i^j, \text{seed}\}, \{ y_i^j = \text{REnc}(D_i^j, \bar{x}_i^j; \text{PRG}_{\pi(j,i)}(\text{seed})) \\
&\quad \hat{y}_i^j \leftarrow \text{RSim}(\bar{y}_i^j = D_i^j(\bar{x}_i^j)) \}.
\end{align*}
\]

Next, we argue that bad encodings in \(K_{\text{re}}\) are well spread around among different chunks of randomized encodings \(y_i^j = \{y_i^j\}_i\) that depend on the same sharing \(\bar{x}_i^j = \{ x_i^j \}\).

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

\[
\Pr \left[ \exists j, \left\{ \{(i, j)\}_{i \in [\lambda]} \cap K_{\text{re}}(K_{\text{seed}}) \right\} \geq c_0 \cdot \lambda^2 \right] \leq O(\mu).
\]

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

It follows from the property of \(\mathcal{D}_{\text{Sim}}\) 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[Gπ], (r_K, K, st), ii) it is independent of Gπ and hence π, and iii) the set of fixed seed bits satisfies Kseed ⊆ Gπ(K), and |K| = O(λ^2) with probability 1 − O(µ). Therefore, it suffices to show that

\[ \Pr \left[ \exists j, \left| \{(i, j)\}_{i \in K} \cap K_{re}(Gπ(K)) \right| \geq c_0 \cdot λ^2 \right] \leq O(µ). \]

J = Gπ(K) is the collection of seed bits that output bits K in f depends on; by construction of f, J = Gπ(K′) for some subset K′ (determined by the dependency graph of REnc), with |K′| = O(|K|). Recall that K_{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(Π_i^j = {π(i, j, k)}_{k \in [s]}) \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:

\[ \Pr \left[ \exists j, \left| \pi \left( \{(i, j, k)\}_{i \in [λ], k \in [s]} \right) \cap (G^{-1} \circ G (π(K′))) \right| \geq c_0 \cdot λ^2 \right] \leq 2^{-Ω(λ^2)} + O(µ) = O(µ). \]

where the last equality follows from µ = 2^{-ω(λ^2)}.

By our assumption on PRG, the input-output dependency graph G of PRG has n = poly(λ)S^{1−α} input nodes, m = poly(λ)S output nodes, and every output node has constant degree ℓ. Furthermore, the degrees of the input nodes are bounded by ℓ′ < o(S^{1−α}). The set \{(i, j, k)\}_{i \in [λ], k \in [s]} has size t = poly(λ)S^α and K′ has size s = O(λ^2) with probability 1 − O(µ). Importantly, both sets are independent of the random permutation π. Thus, we can set the constant c_0 such that c_0 \cdot λ^2 − s ≥ λ^2. Furthermore, we have sℓℓ′t ≤ m − s − t for sufficiently large S (relative to λ) and α < 1/2 with probability 1 − O(µ). 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, J ⊆ [m] be sets of size t and s, respectively, such that s ≤ t and sℓℓ′t ≤ m − s − t, and let B ≥ s + 2. We then have over the choice of a random permutation π over the output nodes (i.e., π: [m] → [m]),

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

**Proof.** Since π(J) ⊆ G^{-1} \circ G (π(J)), we have π(J ∩ J) ⊆ π(J) ∩ (G^{-1} \circ G (π(J))) for all π. We therefore need to bound the probability of

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

The random experiment can equivalently be described as follows: First, assign to each j ∈ J a random, distinct j′ ∈ [m]. Let J′ be the set of all these j′, and let J := G^{-1} \circ G(J′). Then, assign to each v ∈ J \ J a random, distinct v′ ∈ [m] \ J′. Since |J| ≤ sℓℓ′, the probability that the first of these v′ is in J is at most \frac{sℓℓ′}{m−s−1}. 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 J is at most \frac{sℓℓ′}{m−s−2}. Using |J \ J| ≤ t, this implies that the probability of any v′ landing in J is at most \frac{sℓℓ′}{m−s−2}. While the assignments of the v′ are not
This concludes the proof of the theorem. 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 $H_0 = \text{Real}$ can only predict $b$ with probability $1/2 + O(\mu)$.

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

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 $\ell$ and $B$ be positive integers, let $\varepsilon \in [0, 1]$, and let $X$ be a random variable over $\mathbb{Z}_\ell$ with $(B, \varepsilon)$-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 $(B, \varepsilon)$-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 $(B, \varepsilon)$-smudging. \hfill \square

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

**Proof.** By assumption, there are randomized predicates $\{BAD_i : \mathbb{Z}^{\ell+1} \rightarrow \{0, 1\}\}_{i \in [\ell]}$ such that $D_1$ and $D_2$ as defined in Definition 3.4 are identical and $|BAD|_1 \leq K$ 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 $BAD_i(E_i(V_{\Phi_i}), X')$. Let $bad' = (bad'_1 \leftarrow BAD'_i(E_i(V_{\Phi_i}), X + Y))_{i \in [\ell]}$ and let $\mathbb{V}' = \mathbb{V}|_{\Phi_{bad'}, \Phi_{bad'}}$. Since $X$ and $X'$ have the same distribution and $D_1$ and $D_2$ are equal, we also have that $(V, E(V) + X', bad')$ and $(\mathbb{V}', E(V) + X', bad')$ have the same distribution. Note that $Y'$ is independent of $(V, E(V) + X', bad')$ and $(\mathbb{V}', E(V) + X', bad')$, since it only depends on $X'$ given $X + Y$, and nothing depends on $X + Y$ beyond depending on $X'$. Hence, $(V, E(V) + X' + Y', bad')$ and $(\mathbb{V}', E(V) + X' + Y', bad')$ also have the same distribution. Using $X'' + Y'' = X + Y$, we obtain that $D'_1$ and $D'_2$ defined as in Definition 3.4 for $X + Y$ and $BAD'$ are equal. Furthermore, $bad$ and $bad'$ are equally distributed, which implies that $|BAD'|_1 \leq K$ with probability at least $1 - \mu$. Thus, the distribution of $X + Y$ is $(K, \mu)$-flawed-smudging for $(E, Y)$. \hfill \square

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 $\ell$, $B$, 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 $(B, \varepsilon_i)$-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 $(B, \sum_{i=1}^k \alpha_i \varepsilon_i)$-smudging.

42
Proof. Let \( e \in [-B, B] \cap \mathbb{Z}^\ell \). We then have

\[
\delta(X, X + e) = \frac{1}{2} \sum_{x \in \mathbb{Z}^\ell} |\Pr[X = x] - \Pr[X + e = x]| \\
\leq \frac{1}{2} \sum_{x \in \mathbb{Z}^\ell} \sum_{i=1}^{k} \alpha_i \cdot |\Pr[X_i = x] - \Pr[X_i + e = x]| \\
= \sum_{i=1}^{k} \alpha_i \cdot \delta(X_i, X_i + e).
\]

Since \( \delta(X_i, X_i + e) \leq \varepsilon_i \), the claim follows. \(\square\)

**Lemma 6.4.** Let \( \ell, m \), and \( k \) be positive integers, let \( V \) 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 \((K, \mu_i)\)-flawed-smudging distribution for \((E, V)\). 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 \((K, \sum_{i=1}^{k} \alpha_i \mu_i)\)-flawed-smudging for \((E, V)\).

**Proof.** By assumption, there are randomized predicates \( \{ \text{BAD}^{(i)}_j : \mathbb{Z}^{\ell+1} \rightarrow \{0, 1\} \}_{j \in [\ell]} \) for \( i \in [k] \) such that \( D^{(i)}_1 \) and \( D^{(i)}_2 \) as defined in Definition 3.4 for \( X_i \) are identical and \( |\text{bad}^{(i)}|_1 \leq K \) with probability at least \( 1 - \mu_i \). Now define \( \text{BAD}'_j \) 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 \( \text{BAD}'_j(E_j(V_{\Phi_j}), X'_A) \). Let \( \text{bad}' = (\text{bad}' \leftarrow \text{BAD}'_j(E_j(V_{\Phi_j}), X))_{j \in [\ell]} \) and let \( V' \leftarrow V|_{V_{\Phi_{\text{bad}'}}}, \Phi_{\text{bad}'} \) and \( V^{(i)} \leftarrow V|_{V_{\Phi_{\text{bad}^{(i)},}}, \Phi_{\text{bad}^{(i)}}} \).

We have for all \( t \)

\[
\Pr[(V', E(V) + X'_A, \text{bad}') = t] = \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr[(V', E(V) + X'_A, \text{bad}') = t \mid A = i] \\
= \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr[(V^{(i)}), E(V) + X_i, \text{bad}^{(i)}) = t] \\
= \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr[(V, E(V) + X_i, \text{bad}^{(i)}) = t] \\
= \Pr[(V, E(V) + X, \text{bad}') = t].
\]

This established the desired equality of distributions.

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

\[
\Pr[|\text{bad}'|_1 \leq K] = \sum_{i=1}^{k} \Pr[A = i] \cdot \Pr[|\text{bad}^{(i)}|_1 \leq K] \geq \sum_{i=1}^{k} \alpha_i (1 - \mu_i) = 1 - \sum_{i=1}^{k} \alpha_i \mu_i. \quad \square
\]

### 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 $X$ and the fact that BAD does not occur.

The high-level idea is to define the event BAD such that for all $y \in \mathbb{Z}$, $\Pr[X + E = y \land \neg$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 $X_1, \ldots, X_\ell, V_1, \ldots, V_m$ be random variables over $\mathbb{Z}$ such that the $\ell + 1$ random variables $X_1, \ldots, X_\ell$, and $(V_1, \ldots, V_m)$ are mutually independent.\footnote{But, $V_1, \ldots, V_m$ need not be independent.} Further let for $i \in [\ell], \Phi_i \subseteq [m]$, let $E_i : \mathbb{Z}^{[|\Phi_i|]} \to \mathbb{Z}$ be functions, let $F_i := E_i((V_j)_{j \in \Phi_i})$, and let $F_i$ be the support of $F_i$ with $|F_i| < \infty$. For $I \subseteq [\ell]$, let $\Phi_I := \bigcup_{i \in I} \Phi_i$. Then, there exist events BAD$_1, \ldots, BAD_\ell$ with BAD$_\ell = \text{BAD}_\ell(F_\ell, X_\ell)$ such that for all $y_1, \ldots, y_\ell \in \mathbb{Z}, v_1, \ldots, v_m \in \mathbb{Z}$, and for all $I \subseteq [\ell]$,

$$
\begin{align*}
(i) \quad & \Pr[\bigwedge_{i \in [m]} \Phi_i, V_i = v_i \mid \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_i} V_i = v_i \land \bigwedge_{i \in I} \neg$BAD$_i = \Pr[\bigwedge_{i \in [m]} \Phi_i, V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i], \\
(ii) \quad & \Pr[\bigwedge_{i \in I} \neg$BAD$_i \land \bigwedge_{i \in \Phi_i} V_i = v_i] \leq \prod_{i \in I} 4^4 \cdot \sum_{f_i \in F_i} \delta(X_i, X_i + f_i).
\end{align*}
$$

**Proof.** We define BAD$_i$ for $F_i = f_i$ and $X_i = x_i$ as

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

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

$$
\Pr[\text{BAD}_i \mid \bigwedge_{j \in [\ell]} X_j + F_j = y_j \land \bigwedge_{j \in [m]} V_j = v_j] = 1 - \min_{f_i \in F_i} \left\{ \frac{\Pr(X_i + f_i = y_i)}{\Pr(X_i + E_i((v_j)_{j \in \Phi_i}) = y_i)} \right\},
$$

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

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

\[
\Pr[\bigwedge_{i \in I} X_i + F_i = y_i \land \bigwedge_{i \in [m]} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i]
\]

\[
= \Pr \left[ \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i \right]
\]

\[
= \sum_{v_i \in \mathbb{Z} \setminus \Phi_I} \Pr \left[ \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in [m]} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i \right]
\]

\[
= \Pr \left[ \bigwedge_{i \in \Phi_I} V_i = v_i \right] \cdot \prod_{i \in I} \Pr[X_i + E_i((v_j)_{j \in \Phi_i}) = y_i] \cdot \prod_{i \in [\ell]\setminus I} \min_{v_i \in \Phi_I} \{\Pr(X_i + f_i = y_i)\}
\]

\[
= \prod_{i \in I} \left(1 - \min_{v_i \in \Phi_I} \{\Pr(X_i + f_i = y_i)\} \right)
\]

(5)

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

\[
\Pr \left[ \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i \right]
\]

\[
= \sum_{v_i \in \mathbb{Z} \setminus \Phi_I} \Pr \left[ \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in [m]} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i \right]
\]

(6)

Dividing equation (5) by equation (6) yields

\[
\Pr[\bigwedge_{i \in [m] \setminus \Phi_I} V_i = v_i \mid \bigwedge_{i \in [\ell]} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \land \bigwedge_{i \in I} \text{BAD}_i \land \bigwedge_{i \in [\ell]\setminus I} \neg \text{BAD}_i]
\]

\[
= \Pr[\bigwedge_{i \in [m] \setminus \Phi_I} V_i = v_i \mid \bigwedge_{i \in I} \text{BAD}_i]
\]

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

To prove (ii), note that $\bigwedge_{i \in I} \text{BAD}_i$ 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 $\Pr[\bigwedge_{i \in I} \text{BAD}_i]$ equals

$$\sum_{y_i \in Z} \sum_{v_i \in Z} \Pr \left[ \bigwedge_{i \in I} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \right] \cdot \Pr \left[ \bigwedge_{i \in I} \text{BAD}_i \lor \bigwedge_{i \in I} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \right]$$

$$= \sum_{y_i \in Z} \sum_{v_i \in Z} \Pr \left[ \bigwedge_{i \in I} X_i + F_i = y_i \land \bigwedge_{i \in \Phi_I} V_i = v_i \right] \cdot \prod_{i \in I} \left( 1 - \min_{f_i \in F_i} \left\{ \Pr(X_i + f_i = y_i) \right\} \right)$$

$$\leq \sum_{y_i \in Z} \sum_{v_i \in Z} \Pr \left[ \bigwedge_{i \in I} V_i = v_i \right] \cdot \prod_{i \in I} \left( \max_{f_i \in F_i} \Pr(X_i + f_i = y_i) - \min_{f_i \in F_i} \left\{ \Pr(X_i + f_i = y_i) \right\} \right).$$

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

$$\Pr \left[ \bigwedge_{i \in I} \text{BAD}_i \right] \leq \sum_{y_i \in Z} \prod_{i \in I} \left( \max_{f_i \in F_i} \Pr(X_i + f_i = y_i) - \min_{f_i \in F_i} \left\{ \Pr(X_i + f_i = y_i) \right\} \right)$$

$$= \prod_{i \in I} \sum_{y_i \in Z} \left( \max_{f_i \in F_i} \Pr(X_i + f_i = y_i) - \min_{f_i \in F_i} \left\{ \Pr(X_i + f_i = y_i) \right\} \right)$$

$$\leq \prod_{i \in I} \sum_{y_i \in Z} \left( \max_{f_i \in F_i} \Pr(X_i + f_i = y_i) - \Pr[X_i = y_i] \right)$$

$$+ \Pr[X_i = y_i] - \min_{f_i \in F_i} \left\{ \Pr(X_i + f_i = y_i) \right\} \right).$$

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

$$\Pr \left[ \bigwedge_{i \in I} \text{BAD}_i \right] \leq \prod_{i \in I} 2 \cdot \sum_{f_i \in F_i, y_i \in Z} |\Pr[X_i = y_i] - \Pr(X_i + f_i = y_i)|$$

$$= \prod_{i \in I} 4 \cdot \sum_{f_i \in F_i} \delta(X_i, X_i + f_i).$$

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

**Proposition 6.6.** Let $\ell$ and $K$ be positive integers and let $\mathcal{X}$ be a distribution over $\mathbb{Z}^\ell$ such that for $(X_1, \ldots, X_\ell) \overset{\$}{\leftarrow} \mathcal{X}$, $X_1, \ldots, X_\ell$ are mutually independent and the distribution of each $X_i$ is $(B, \varepsilon)$-smudging for $\varepsilon \leq \frac{K^\ell}{2^{K\ell}(2B+1)}$. Further let $\mu = 2^{(11\ell(2B+1)^\varepsilon)K^\ell + 1}$. Then, $\mathcal{X}$ is $(K, \mu)$-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 $E_i : \mathbb{Z}^{\left| \Phi_i \right|} \to \mathbb{Z}$ be functions. For $I \subseteq [\ell]$, let $\Phi_I := \bigcup_{i \in I} \Phi_i$ and let $E : \mathbb{Z}^m \to \mathbb{Z}^\ell$ be as in Definition 3.4. Let $\mathcal{V}$ be a
distribution over $Z^m$ such that $E(V)$ 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 $D_1$, $D_2$ be the distributions from Definition 3.4. Finally let $I \subseteq [\ell]$, $y \in Z^l$, and $v \in Z^m$. Then,

$$\Pr[V = v \wedge E(V) + X = y \wedge BAD = I]$$

= \Pr[\bigwedge_{i \in \Phi} V_i = v_i \wedge E(V) + X = y \wedge BAD = I]

\cdot \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i \wedge E(V) + X = y \wedge BAD = I] \tag{7}

= \Pr[\bigwedge_{i \in \Phi_i} V_i = v_i \wedge E(V) + X = y \wedge BAD = I] \cdot \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i | \bigwedge_{i \in \Phi_i} V_i = v_i],

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

$$\Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \mid E(V) + X = y \wedge Z]$$

= \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i \wedge E(V) + X = y \wedge Z].

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

$$\Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \mid E(V) + X = y \wedge Z] = \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i].$$

Hence,

$$\Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \wedge E(V) + X = y \wedge Z]$$

= \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i] \cdot \Pr[E(V) + X = y \wedge Z].

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

$$\Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \mid E(V) + X = y \mid Z]$$

= \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i] \cdot \Pr[E(V) + X = y \mid Z]. \tag{8}

We finally obtain

$$\Pr[V = v \wedge E(V) + X = y \wedge BAD = I]$$

\overset{(7)}{=} \Pr[Z] \cdot \Pr[E(V) + X = y \mid Z] \cdot \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} V_i = v_i \mid \bigwedge_{i \in \Phi_i} V_i = v_i]

\overset{(8)}{=} \Pr[Z] \cdot \Pr[\bigwedge_{i \in [m] \setminus \Phi_i} \overline{V}_i = v_i \wedge E(V) + X = y \mid Z]

= \Pr[\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 BAD = I].

Since $\overline{V}_i = V_i$ if BAD = I, this implies that the distributions $D_1$ and $D_2$ are equal.

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

$$\Pr[|BAD|_1 > K] = \sum_{k=K+1}^{\ell} \Pr[|BAD|_1 = k] \leq \sum_{k=K+1}^{\ell} \sum_{|I| = k}^{\ell} \Pr[\bigwedge_{i \in I} BAD_i] \leq \sum_{k=K+1}^{\ell} \sum_{|I| = k}^{\ell} (4B + 1)\varepsilon)^{|I|}.$$
There are \( \binom{k}{\ell} \) subsets \( I \subseteq [\ell] \) of size \( k \). Using the bound \( \binom{k}{\ell} \leq \left( \frac{e}{\ell} \right)^k \), where \( e \) is Euler’s constant, we obtain

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

Let \( \alpha := \frac{11\ell \cdot (2B + 1)\varepsilon}{K + 1} \). Since \( \varepsilon \leq \frac{K+1}{2\ell(2B+1)} \) 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

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

Hence, the probability that \( |\text{BAD}|_1 \leq K \) is at least \( 1 - 2\alpha^{K+1} = 1 - \mu \). Altogether, we conclude that \( X \) is \((K, \mu)\)-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 \( S_{n,C} \) be the distribution of \( S \) defined as

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

where \( \alpha_1, \ldots, \alpha_{n/2}, \beta_1, \ldots, \beta_n \), and \( \gamma \) are independent and distributed uniformly over \( \{-C, \ldots, C\} \), and \( \sigma \) 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

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

where for \( 1 \leq i \leq j \leq n \), \( \alpha_{i,j}, \beta_i \), and \( \gamma \) 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 \( X_{B_1,B_2} \) 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 \( X_{B_1,B_2} \), then

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

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 \( X_{B_1,B_2} \), and the seed for \( MQ \) is uniform.
Definition 6.10. Let \( n, n', m \) 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 \( B' \) be polynomials for all \( \lambda \in \mathbb{N} \). We then define our candidate \((n + n',m)\)-PFG \( \mathcal{CFG}_{\lambda}^{n,n',m,B_1,B_2,B',C,C'} \) for \( \lambda \in \mathbb{N} \) as follows: \( \mathcal{CFG}_{\lambda}^{n,n',m,B_1,B_2,B',C,C'} \) samples a function \( \text{PFG}: \mathbb{Z}^{n(\lambda)+n'(\lambda)} \rightarrow \mathbb{Z}^{m(\lambda)} \) by sampling \( S_i \stackrel{\$}{\leftarrow} \mathcal{S}_{n(\lambda),C}, \text{MQ}_i \stackrel{\$}{\leftarrow} \mathcal{MQ}_{n'(\lambda),C'}(\lambda) \) for \( i \in \{1,\ldots,m(\lambda)\} \), and setting

\[
\text{PFG}\left((x_i)_{i \in [n(\lambda)]},(x'_i)_{i \in [n'(\lambda)]}\right) = \left(S_j\left((x_i)_{i \in [n(\lambda)]}\right) + \text{MQ}_j\left((x'_i)_{i \in [n'(\lambda)]}\right)\right)_{j \in [m(\lambda)]};
\]

Moreover, let \( D_{\text{seed}} \) be the distribution that samples \( x_1,\ldots,x_n(\lambda) \) from \( X_{B_1(\lambda),B_2(\lambda)} \), samples \( x'_1,\ldots,x'_{n'}(\lambda) \) uniformly from \( \{-B'(\lambda),\ldots,B'(\lambda)\} \), and outputs \( (x_1,\ldots,x_n(\lambda),x'_1,\ldots,x'_{n'}(\lambda)) \).

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

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

\[
\left\{(\text{PFG}, D_{\text{seed}}) \stackrel{\$}{\leftarrow} \mathcal{CFG}_{\lambda}^{n,n',m,B_1,B_2,B',C,C'}; \text{seed} \stackrel{\$}{\leftarrow} D_{\text{seed}} : (\text{PFG}, \text{PFG}(\text{seed}))\right\}_{\lambda \in \mathbb{N}}; \quad \left\{(\text{PFG}, D_{\text{seed}}) \stackrel{\$}{\leftarrow} \mathcal{CFG}_{\lambda}^{n,n',m,B_1,B_2,B',C,C'}; Y \stackrel{\$}{\leftarrow} D'(\lambda) : (\text{PFG}, Y + Z)\right\}_{\lambda \in \mathbb{N}}.
\]

The intuition is as follows: We show that \( \mathcal{S}_{n(\lambda),C}(\mathcal{X}_{B_1(\lambda),B_2(\lambda)}^{n(\lambda)}) \) 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 lemmas.

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,

\[
\Pr(c_1X_1 + c_2X_2 = z) = \frac{1}{(2B + 1)^2} \cdot \left|\left\{y \in c_1 \cdot \{-B,\ldots,B\} \mid z - y \in c_2 \cdot \{-B,\ldots,B\}\right\}\right|.
\]
Let \( d_i := |c_i| \cdot B \) 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 \( y_0 < c_1 c_2 \) and \( c_1 \mathbb{Z} \cap (c_2 \mathbb{Z} + z) = y_0 + c_1 c_2 \mathbb{Z} \). Hence, \( D = (y_0 + c_2 c_1 \mathbb{Z}) \cap [-d_1, d_1] \cap [-d_2 + z, d_2 + z] \). 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 \( \left\lfloor \frac{\ell}{c_1 c_2} \right\rfloor \) integers congruent to \( y_0 \) modulo \( c_1 c_2 \), and at most \( \left\lfloor \frac{\ell}{c_1 c_2} \right\rfloor \) of these, for \( \ell := |[-d_1, d_1] \cap [-d_2 + z, d_2 + z] \cap \mathbb{Z}| \). We can therefore conclude that for all \( z \in \mathbb{Z} \),

\[
\Pr(c_1 X_1 + c_2 X_2 = z) \leq \frac{1}{(2B + 1)^2} \left( \left\lfloor \frac{|[-d_1, d_1] \cap [-d_2 + z, d_2 + z] \cap \mathbb{Z}|}{c_1 c_2} \right\rfloor + 1 \right). 
\]

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 \( \Pr(c_1 X_1 + c_2 X_2 = z) \) and \( \Pr(c_1 X_1 + c_2 X_2 = z - e) \) differ by at most \( \frac{1}{(2B + 1)^2} \cdot \left( \left\lfloor \frac{|e|}{c_1 c_2} \right\rfloor + 2 \right) \).

**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 \( \{-B, \ldots, B\} \). Further let \( e \in \mathbb{Z} \), \( |e| \leq 2B + 1 \). Then,

\[
\delta(c_1 X_1 + c_2 X_2, c_1 X_1 + c_2 X_2 + e) \leq \frac{2|e| + |c_1| + |c_2| + 2}{2B + 1}.
\]

**Proof.** By definition, we have

\[
\delta(c_1 X_1 + c_2 X_2, c_1 X_1 + c_2 X_2 + e) = \frac{1}{2} \sum_{z \in \mathbb{Z}} |\Pr(c_1 X_1 + c_2 X_2 = z) - \Pr(c_1 X_1 + c_2 X_2 + e = z)|.
\]

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

\[
\delta(c_2 X_2, c_2 X_2 + e) = \frac{1}{2} \cdot \frac{1}{2B + 1} \cdot 2|e| \leq \frac{2|e| + |c_1| + |c_2| + 2}{2B + 1}.
\]

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 \( |z| > \left( |c_1| + |c_2|\right) B + |e| \), we have \( \Pr(c_1 X_1 + c_2 X_2 = z) = 0 \) and \( \Pr(c_1 X_1 + c_2 X_2 + e = z) = 0 \). 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 \( \frac{1}{(2B + 1)^2} \cdot \left( \left\lfloor \frac{|e|}{|c_1 c_2|} \right\rfloor + 2 \right) \). The statistical distance is thus at most

\[
\frac{1}{2} \cdot (2(|c_1| + |c_2|) B + 2|e| + 1) \cdot \frac{1}{(2B + 1)^2} \cdot \left( \left\lfloor \frac{|e|}{|c_1 c_2|} \right\rfloor + 2 \right) = \frac{1}{2} \cdot \frac{1}{(2B + 1)^2} \cdot \left( 2|e||c_1 + c_2| B + 2|e|^2 + |e| + 4(|c_1| + |c_2|) B + 4|e| + 2 \right).
\]

Since \( |c_1| + |c_2| \leq 2 \cdot \max\{|c_1|, |c_2|\} \) and \( |c_1|, |c_2| \geq 1 \), we have \( \frac{|c_1| + |c_2|}{|c_1 c_2|} \leq 2 \). This implies

\[
\delta(c_1 X_1 + c_2 X_2, c_1 X_1 + c_2 X_2 + e) \leq \frac{1}{(2B + 1)^2} \cdot \left( 4|e| B + 2|e|^2 + |e| + 4(|c_1| + |c_2|) B + 4|e| + 2 \right) = \frac{1}{(2B + 1)^2} \cdot \left( |e|(2B + 1) + |e|^2 + \frac{3}{2}|e| + 2(|c_1| + |c_2|) B + 1 \right).
\]
Because for all $0 < b \leq a$, \( a - b \geq \frac{ab + a - ab - b}{b(b + 1)} = 0 \), we have \( \frac{2 \left(|c_1| + |c_2|\right)B + 1}{2B + 1} \leq \frac{2 \left(|c_1| + |c_2|\right)B}{2B} = |c_1| + |c_2| \). Using $|e| \leq 2B + 1$, we further have \( \frac{|e|}{2B + 1} \leq 1 \). This yields

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

\[\square\]

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

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

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

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

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,

\[
\Pr(\gcd(X, Y) = 1) = \Pr(\gcd(|X|, |Y|) = 1) \geq \Pr(\gcd(X', Y') = 1) \geq \frac{13}{22}.
\]

\[\square\]

**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 \{-$B$, \ldots, $B$\},

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

**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]\footnote{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.}

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

where $\zeta$ 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 $B$,

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

For $X_1 < 0$, 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 $\Pr(X_1 \neq 0 \land X_2 \neq 0) = \left( \frac{2B}{2B+1} \right)^2 \geq \frac{5}{6}$. Hence, we obtain for all sufficiently large $B$

\[
\Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1)
\geq \Pr(\gcd(\alpha_1X_1 + \beta_1, \alpha_2X_2 + \beta_2) = 1 \mid X_1 \neq 0 \land X_2 \neq 0) \cdot \Pr(X_1 \neq 0 \land X_2 \neq 0)
\geq 0.6 \cdot \frac{5}{6} = \frac{1}{2}.
\]

\[\square\]
Using the above lemmata, we can now prove that if $S$ is chosen from $S_{n,C}$, 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 $X_{B_1, B_2}$.

**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 $S_{n,C}$, 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 $X_{B_1, B_2}$ (and now fixed $S$) and for all $e \in \mathbb{Z}$ with $|e| \leq 2B_2 + 1$, 

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

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

**Proof.** Note that 

$$
S(X) = \sum_{i=1}^{n/2} \left( (\alpha_i X_{\sigma_{2i-1}} + \beta_{\sigma_{2i}}) \cdot X_{\sigma_{2i}} + \beta_{\sigma_{2i-1}} X_{\sigma_{2i-1}} \right) + \gamma.
$$

For $i \in \{1, \ldots, n/4\}$, let $E_i$ be the event that $\gcd(\alpha_i, \beta_{\sigma_{2i}}) = 1$. 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 $\epsilon > 0$

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

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

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

Let

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

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 $X_{B_1, B_2}$, 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 $i \in I_S$

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

52
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 $S(X)$ 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 $|c_1|, |c_2| \leq C \cdot (B_1 + 1)$, 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 $S(X)$ and $S(X) + e$ for $e \in Z$ 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 $X_{\sigma(2i-3)} = x \land X_{\sigma(2i-1)} = x'$ for $x, x' \in Z$. Then, the statistical distance of $S(X)$ conditioned on $G'_i$ and $S(X) + e$ conditioned on $G'_i$ is

$$\frac{1}{2} \sum_{z \in Z} \left| \Pr(S(X) = z \mid G'_i) - \Pr(S(X) + e = z \mid G'_i) \right|$$

$$\leq \frac{1}{2} \sum_{z \in Z} \left[ \sum_{x,x' \in Z} \frac{\Pr(F_{i,x,x'} \cdot (\Pr(S(X) = z \mid G'_i \cap F_{i,x,x'}) - \Pr(S(X) + e = z \mid G'_i \cap F_{i,x,x'}))}{\Pr(G'_i \cap F_{i,x,x'}) \neq 0} \right]$$

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

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

$$\delta(S(X), S(X) + e) \leq \frac{2|e| + 2C \cdot (B_1 + 1) + 2}{2B_2 + 1} + \Pr(\forall i \in I_S \neg G'_i)$$

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

**Corollary 6.17.** Let $C \in N \setminus \{0\}$. We then have for all even $n \in N$, for all sufficiently large $B_1 \in N$ and all $B_2 \geq B_1$, for $S$ chosen from $S_{n,C}$, and $X_1, \ldots, X_n$ chosen independently from $X_{B_1,B_2}$, that the distribution of $S(X_1, \ldots, X_n)$ is $(B, \varepsilon)$-smudging for all $B \leq 2B_2 + 1$ and

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

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

$$\delta(S(X), S(X) + e) = \sum_{s \in G} \Pr[S = s] \cdot \delta(s(X), s(X) + e) + \sum_{s \in \overline{G}} \Pr[S = s] \cdot \delta(s(X), s(X) + e)$$

$$\leq \Pr[S \in G] \cdot \varepsilon' + \Pr[S \in \overline{G}]$$

$$\leq \varepsilon' + 2^{-n/141}.$$
Since $|e| \leq B$ and $\left(\frac{1}{37}\right)^{n/141} \leq \left(\frac{31}{32}\right)^{n/16}$, we have $\varepsilon + 2^{n/141} \leq \varepsilon$, which concludes the proof. \(\square\)

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

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

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

**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 $(K,\mu)$-flawed-smudging for $B$-bounded distributions with $K = \lambda^{e_2}$ and subexponentially small $\mu$. To satisfy this, we can take our candidate $\mathcal{CPF}_\lambda^{n,n',m,B_1,B_2,B',C,C'}$ and set the parameters as follows: Set $n = [S^{1-\alpha}/2]$, $n' = S^{1-\alpha} - n$, 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 $(K,\mu)$-flawed-smudging for $B$-bounded distributions, for

$$\mu = O\left(\frac{1}{K+1}\right)^{K+1} = O\left(\lambda^{-e_2}(\lambda^{e_2})\right).$$

**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 (CARER), 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


[KNT18] Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Obfustopia built on secret-


