GGH15 Beyond Permutation Branching Programs: Proofs, Attacks, and Candidates

Yilei Chen[1], Vinod Vaikuntanathan[2], and Hoeteck Wee[3]

[1] Boston University, Boston, USA
chenyl@bu.edu
[2] MIT, Cambridge, USA
vinodv@csail.mit.edu
[3] CNRS and ENS, PSL, Paris, France
wee@di.ens.fr

Abstract. We carry out a systematic study of the GGH15 graded encoding scheme used with *general* branching programs. This is motivated by the fact that general branching programs are more efficient than permutation branching programs and also substantially more expressive in the read-once setting. Our main results are as follows:

- **Proofs.** We present new constructions of private constrained PRFs and lockable obfuscation, for constraints (resp. functions to be obfuscated) that are computable by general branching programs. Our constructions are secure under LWE with subexponential approximation factors. Previous constructions of this kind crucially rely on the permutation structure of the underlying branching programs. Using general branching programs allows us to obtain more efficient constructions for certain classes of constraints (resp. functions), while posing new challenges in the proof, which we overcome using new proof techniques.
- **Attacks.** We extend the previous attacks on indistinguishability obfuscation (iO) candidates that use GGH15 encodings. The new attack simply uses the rank of a matrix as the distinguisher, so we call it a "rank attack". The rank attack breaks, among others, the iO candidate for general read-once branching programs by Halevi, Halevi, Shoup and Stephens-Davidowitz (CCS 2017).
- **Candidate Witness Encryption and iO.** Drawing upon insights from our proofs and attacks, we present simple candidates for witness encryption and iO that resist the existing attacks, using GGH15 encodings. Our candidate for witness encryption crucially exploits the fact that formulas in conjunctive normal form (CNFs) can be represented by general, *read-once* branching programs.

1 Introduction

Graph-induced graded encodings – henceforth called GGH15 encodings – were put forth by Gentry, Gorbunov and Halevi [23] as a candidate instantiation of (approximate) cryptographic multilinear maps [8, 20], with the hope that these

encodings could in turn be used to build advanced cryptographic primitives whose security is related to the hardness of the learning with errors (LWE) problem [36]. In addition, following [20,21], the same work presented candidate constructions of multi-party key exchange and indistinguishability obfuscation (iO) starting from these graded encoding schemes.

In the last few years, a very fruitful line of works has shed a great deal of insight into the use of GGH15 encodings in two complementary settings: constructing security reductions from LWE (partially validating the intuition in GGH15), and demonstrating efficient attacks. The former include constructions of private constrained pseudorandom functions (PRFs) [13], lockable obfuscation (aka obfuscating the "compute-then-compare" functionality) [26,38] and encryption schemes that constitute counter-examples for circular security [27,30]. The latter include efficient attacks [15,17] on the key exchange and iO candidates described in [23]. One of the key distinctions between the two settings is whether an adversary can obtain encodings of zero from honest evaluations. For all the applications that can be based on LWE, the adversary cannot trivially obtain encodings of zero; whereas the attacks apply only to settings where the adversary can trivially obtain encodings of zero. There is much grey area in between, where we neither know how to obtain encodings of zero nor are we able to prove security based on LWE (e.g., in the setting of witness encryption).

This work. In this work, we explore the use of GGH15 encodings together with general (non-permutation) matrix branching programs. In particular, we present (i) new constructions of private constrained PRFs and lockable obfuscation from LWE, (ii) new attacks on iO candidates, and (iii) new candidates for iO and witness encryption that resist our new attacks as well as prior attacks. At the core of these results are new techniques and insights into the use of GGH15 encodings for a larger class of branching programs.

Most of the prior constructions and candidates for the primitives we consider follow the template laid out in [21]: start with the class of NC^1 circuits, represented using permutation branching programs, which are specified by a collection of permutation matrices $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$. Computation in such a program proceeds by taking a subset product of these matrices, where the choice of the subset is dictated by the input but the order in which the matrices are multiplied is oblivious to the input. To cryptographically "protect" this computation, we will first pre-process and randomize $\{\mathbf{M}_{i,b}\}$ to obtain a new collection of matrices $\{\hat{\mathbf{S}}_{i,b}\}$, and then encode the latter using graded encodings. Functionality (e.g. evaluation in lockable obfuscation and iO) relies on the fact that we can check whether some subset product of the $\hat{\mathbf{S}}_{i,b}$'s is zero (or the identity matrix) using the underlying graded encodings. Any security proof or attack would of course depend on the class of matrices $\mathbf{M}_{i,b}$'s we start out with, and how the $\hat{\mathbf{S}}_{i,b}$'s are derived.

Beyond permutation matrices. From a feasibility point of view, working with permutation matrices is without loss of generality. We know that any NC^1 circuit (or even a logspace computation) can be represented as a permutation matrix

branching program [5]. Moreover, any general branching program, where the underlying matrices are possibly low-rank, can be converted to a permutation branching program with a polynomial blow-up in the number and dimensions of these matrices. Nonetheless, there are advantages to working with more general, not necessarily permutation or full-rank, branching programs:

- The first is concrete efficiency. For instance, representing equality or point functions on ℓ-bit string would use $O(\ell^2)$ constant-width matrices with permutation branching programs, but just 2ℓ width-one matrices (i.e. entries) with general branching programs.
- The second is that in the read-once setting, general branching programs are more expressive than permutation branching programs. The restriction to read-once branching programs is useful in applications such as iO and witness encryption, as they allow us to disregard "multiplicative bundling" factors that protect against mixed-input attacks, which in turn yields much more efficient constructions. This was shown in a recent work of Halevi, Halevi, Shoup and Stephens-Davidowitz (HHSS) [29], which presented an iO candidate for read-once branching programs based on GGH15 encodings. Their candidate is designed for general read-once branching programs, as read-once permutation branching programs only capture an extremely limited class of functions.

This raises the natural question of the security of GGH15-based constructions when applied to general (non-permutation, possibly low-rank) matrix branching programs, as is exactly the focus of this work. Indeed, the afore-mentioned proof techniques *and* attacks break down in this setting. In particular, the HHSS iO candidate appears to resist the existing attacks in [15,17], thanks in part to the use of low-rank matrices (cf. [29, Section 1.2]).

We proceed to describe our results and techniques in more detail.

1.1 Our Results I: New Cryptographic Constructions from LWE

We present new constructions of private constrained PRFs and lockable obfuscation that work directly with general matrix branching programs. As with prior works, our constructions are secure under the LWE assumption with subexponential approximation factors. Our result generalizes the previous constructions in [13,26,38] which only work for permutation branching programs, and yields improved concrete efficiency for several interesting classes of functions that can be represented more efficiently using general branching programs, as described next.

- Lockable obfuscation [26,38] refers to the average-case secure virtual black-box (VBB) obfuscation for a class of functionalities $\mathbf{C}[f, y]$ which, on input x, output 1 if $f(x) = y$ and 0 otherwise. The average-case refers (only) to a uniformly random choice of y (more generally, y with sufficient min-entropy). For lockable obfuscation, we obtain improved constructions for a class of "compute" functions where each output bit is computed using a

general branching program applied to the input x (whereas [26, 38] require permutation branching programs). To illustrate the efficiency gain, consider the case where each output bit of the underlying function f computes a disjunction or conjunction of the ℓ input bits. In this case, we achieve up to a quadratic gain in efficiency due to our support for general branching programs. This class generalizes the distributional conjunction obfuscator studied in [10, 12, 38].

– Private puncturable PRFs are an important special case of constrained PRFs, with many applications such as 2-server private information retrieval (PIR) [6]. We obtain a very simple private puncturable PRF with a quadratic efficiency improvement over the recent GGH15-based construction of Canetti and Chen [13]. Nonetheless, our construction is admittedly less efficient –for most settings of parameters– than the more complex constructions in [6, 11] that combines techniques from both fully-homomorphic and attribute-based encryption.

Next, we provide a very brief overview of our techniques, and defer a more detailed technical overview to Section 2.

New constructions and proof techniques. A GGH15 encoding of a low-norm matrix $\hat{\mathbf{S}}$ w.r.t. two matrices $\mathbf{A}_0$ and $\mathbf{A}_1$ is defined to be along the edge $\mathbf{A}_0 \mapsto \mathbf{A}_1$ and is computed as

$$\mathbf{D} \leftarrow \mathbf{A}_0^{-1}(\hat{\mathbf{S}}\mathbf{A}_1 + \mathbf{E})$$

where for all $\mathbf{A}$, $\mathbf{Y}$ with proper dimensions, the notation $\mathbf{D} \leftarrow \mathbf{A}^{-1}(\mathbf{Y})$ means that $\mathbf{D}$ is a random low-norm matrix such that $\mathbf{AD} = \mathbf{Y} \bmod q$.

The constructions in [13, 26, 27, 38] encode any permutation matrix $\mathbf{M} \in \{0,1\}^{w \times w}$ as a GGH15 encoding of $\hat{\mathbf{S}} = \mathbf{M} \otimes \mathbf{S}$, i.e.

$$\mathbf{A}_0^{-1}((\mathbf{M} \otimes \mathbf{S})\mathbf{A}_1 + \mathbf{E})$$

for a random low-norm $\mathbf{S}$. The crux of the analysis is to show that $\mathbf{M}$ is hidden under the LWE assumption, namely: for any *permutation* matrix $\mathbf{M} \in \{0,1\}^{w \times w}$,

$$(\mathbf{A}_0, \mathbf{A}_0^{-1}((\mathbf{M} \otimes \mathbf{S})\mathbf{A}_1 + \mathbf{E})) \approx_c (\mathbf{A}_0, \mathbf{V}) \tag{1}$$

where $\mathbf{A}_0, \mathbf{A}_1$ are uniformly random over $\mathbb{Z}_q$, $\mathbf{S}, \mathbf{V}, \mathbf{E}$ are random low-norm matrices, $\approx_c$ stands for computational indistinguishable. The proof of (1) follows quite readily from the fact that given any permutation matrix $\mathbf{M} \in \{0,1\}^{w \times w}$, we have:

$$(\mathbf{A}, (\mathbf{M} \otimes \mathbf{S})\mathbf{A} + \mathbf{E}) \approx_c (\mathbf{A}, \mathbf{U})$$

under the LWE assumption, where $\mathbf{U}$ is uniformly random.

However, this statement is false for arbitrary matrices $\mathbf{M}$, take for instance $\mathbf{M} = \mathbf{0}^{w \times w}$, the all-0 matrix. Indeed, the reader can easily come up with rank-$(w-1)$ matrices $\mathbf{M}$ for which equation (1) fails to hold.

In our construction, we encode an arbitrary matrix $\mathbf{M}$ as a GGH15 encoding of

$$\hat{\mathbf{S}} = \begin{pmatrix} \mathbf{M} \otimes \mathbf{S} & \\ & \mathbf{S} \end{pmatrix}$$

That is, we append $\mathbf{S}$ along the diagonal. We then establish the following analogue of (1) under the LWE assumption: for any *arbitrary* $\mathbf{M} \in \{0,1\}^{w \times w}$,

$$\left(\mathbf{J}\mathbf{A}_0, \mathbf{A}_0^{-1}\left(\left(\begin{matrix} \mathbf{M} \otimes \mathbf{S} & \\ & \mathbf{S} \end{matrix}\right)\mathbf{A}_1 + \mathbf{E}\right)\right) \approx_c \left(\mathbf{J}\mathbf{A}_0, \mathbf{V}\right) \tag{2}$$

where $\mathbf{J}$ is any matrix of the form $[\star \mid \mathbf{I}]$, and $\mathbf{A}_0, \mathbf{A}_1, \mathbf{S}, \mathbf{V}, \mathbf{E}$ are distributed as in (1). This statement is qualitatively incomparable with (1): it is stronger in that it works for *arbitrary* $\mathbf{M}$, but weaker in that the distinguisher only sees partial information about $\mathbf{A}_0$.

Proving the statement in (2) requires a new proof strategy where we will treat $\mathbf{S}$ (instead of $\mathbf{A}_0, \mathbf{A}_1$) as a public matrix known to the distinguisher. In particular, we start with taking the bottom part of $\mathbf{A}_1$ as the LWE secret, in conjunction with the public $\mathbf{S}$ in the bottom-right diagonal; then use an extension of the trapdoor sampling lemma by Gentry et al. [25] to produce an "oblique" (while *statistically* indistinguishable) preimage sample using only the trapdoor of the top part of $\mathbf{A}_0$; finally argue that the "oblique" sample is *computationally* indistinguishable from random Gaussian using the top part of $\mathbf{A}_0$ as the LWE secret. Walking through these steps requires new techniques on analyzing the trapdoor sampling detailed in Section 4. We refer the readers to Sections 2.2 and 5.3 for further explanation of the proof techniques.

Next, we show that the weaker guarantee in (2) (in that the distinguisher gets $\mathbf{J}\mathbf{A}_0$ instead of $\mathbf{A}_0$) is sufficient for constructions of constrained PRFs and lockable obfuscation based on GGH15 encodings; this yields new constructions that are directly applicable to general, non-permutation matrix branching programs.

1.2 Our Results II: New Attacks on iO Candidates

Next, we turn our attention to iO, where adversaries can obtain encodings of zero through honest evaluations. Concretely, we focus on iO candidates that follow the [21] template described earlier in the introduction: start with a branching program $\{\mathbf{M}_{i,b}\}$, pre-process and randomize $\{\mathbf{M}_{i,b}\}$ to obtain a matrices $\{\hat{\mathbf{S}}_{i,b}\}$, and encode the latter using GGH15 encodings.

We present an attack that run in time $\mathsf{size}^{O(c)}$ for general read-c branching programs of size size. In particular, we have a polynomial-time attack when c is constant, as is the case for the iO candidate in [29] which corresponds to $c = 1$. Our attack covers various "safeguards" in the literature, such as Kilian-style randomization, multiplicative bundling, and diagonal padding.

Attack overview. Our attack is remarkably simple, and proceeds in two steps:

1. Compute a matrix $\mathbf{V}$ whose (i,j)'th entry correspond to an iO evaluation on input $x^{(i)} \mid y^{(j)}$ that yields an encoding of zero. The dimensions of $\mathbf{V}$ and the number of evaluations is polynomial in size^c.

2. Output the rank of $\mathbf{V}$ (over $\mathbb{Z}$). More precisely, check if $\mathsf{rank}(\mathbf{V})$ is above some threshold.

Step 1 was used in the attack of Coron et al. [17] and Chen et al. [15], both originated from the zeroizing attack of Cheon et al. [16] on CLT13 [19]. The novelty of our analysis lies in showing that $\mathsf{rank}(\mathbf{V})$ leaks information about the $\hat{\mathbf{S}}_{i,b}$'s and thus the plaintext branching program matrices $\mathbf{M}_{i,b}$'s. So we call the attack a "rank attack".

Our attack improves upon the previous attack of Chen et al. [15] on GGH15-based iO candidates in several ways: (i) we have a classical as opposed to a quantum attack, and (ii) it is applicable to a larger class of branching programs, i.e. branching programs that are not necessarily input-partitioned or using permutation matrices.

Why the rank-attack works? To get a taste of the rank-attack, let's consider an oversimplified description of the iO candidates based on GGH15 encodings. Let $\{\hat{\mathbf{S}}_{i,b}\}$ be the randomization of plaintext matrices $\{\mathbf{M}_{i,b}\}$. Then the obfuscated code is the GGH15 encodings of the $\hat{\mathbf{S}}_{i,b}$ matrices

$$\mathbf{A}_0, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}} \text{ where } \mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}\left(\hat{\mathbf{S}}_{i,b}\mathbf{A}_i + \mathbf{E}_{i,b}\right)$$

Evaluation proceeds by first computing the product of $\mathbf{A}_0$ with the subset product of the $\mathbf{D}_{i,b}$ matrices. As an example, for the obfuscation of a 3-step branching program that computes all-0 functionality, the evaluation on input $x = 000$ gives

$$\mathsf{Eval}(x) = \mathbf{A}_0\cdot\mathbf{D}_{1,0}\cdot\mathbf{D}_{2,0}\cdot\mathbf{D}_{3,0} = \hat{\mathbf{S}}_{1,0}\hat{\mathbf{S}}_{2,0}\mathbf{E}_{3,0}+\hat{\mathbf{S}}_{1,0}\mathbf{E}_{2,0}\mathbf{D}_{3,0}+\mathbf{E}_{1,0}\mathbf{D}_{2,0}\mathbf{D}_{3,0} \quad (3)$$

To give a sense of why computing the rank is useful in an attack, we make a further simplification, that suppose we manage to learn the monomial

$$\hat{\mathbf{S}}_{1,0}\mathbf{E}_{2,0}\mathbf{D}_{3,0} \in \mathbb{Z}^{t\times m}.$$

W.h.p., the Gaussians $\mathbf{E}_{2,0}, \mathbf{D}_{3,0}$ and therefore its product $\mathbf{E}_{2,0}\mathbf{D}_{3,0}$ are full rank (over $\mathbb{Z}$), so the rank of this term is that of $\hat{\mathbf{S}}_{1,0}$, which leaks some information about the rank of $\mathbf{M}_{1,0}$. Note that learning the rank of $\mathbf{M}_{1,0}$ leaks no useful information for permutation branching programs, but is sufficient to break iO for general branching programs.

In actuality, a single evaluation corresponding to an encoding of zero only provides a single value in $\mathbb{Z}$, which is a *sum* of products of the form above, multiplied by some left and right bookend vectors. To extract the important information out of the summation of random-looking terms, we will first form a matrix $\mathbf{V}$ of evaluations on appropriately chosen inputs. The matrix $\mathbf{V}$ has the property that it factors into the product of two matrices $\mathbf{V} = \mathbf{X}\cdot\mathbf{Y}$. We proceed analogously to the toy example in two steps with $\mathbf{X}, \mathbf{Y}$ playing the roles of $\hat{\mathbf{S}}_{1,0}$ and $\mathbf{E}_{2,0}\cdot\mathbf{D}_{3,0}$:

1. argue that $\mathbf{Y}$ is non-singular over $\mathbb{Q}$ so that $\mathsf{rank}(\mathbf{V}) = \mathsf{rank}(\mathbf{X})$, and

2. argue that $\mathsf{rank}(\mathbf{X})$ leaks information about the underlying branching program.

So far we have described what the analysis looks like for the read-once branching programs (i.e. $c = 1$). For the case of $c > 1$, the analysis has the flavor of converting the obfuscated code of a read-c branching program into the read-once setting, using the "tensor switching lemmas" from previous attacks [4,18] on iO candidates that use GGH13 and CLT13.

The code that demonstrates the attack as a proof-of-concept is available at https://github.com/wildstrawberry/cryptanalysesBPobfuscators.

1.3 Our Results III: New Candidates

Given the insights from our proofs and attacks, we present simple candidates for witness encryption and iO based on GGH15 encodings. Our witness encryption candidate relies on the observation from [24] that to build witness encryption for general NP relations, it suffices to build witness encryption for CNF formulas, and that we can represent CNF formulas using general, read-once branching programs. The ciphertext corresponding to a formula Ψ and a message $\mu \in \{0,1\}$ is of the form described in (2), namely

$$\mathbf{J}\mathbf{A}_0, \left\{ \mathbf{A}_{i-1}^{-1} \left(\begin{pmatrix} \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} & \\ & \mu \mathbf{S}_{i,b} \end{pmatrix} \mathbf{A}_i + \mathbf{E}_{i,b} \right) \right\}$$

where $\mathbf{J}$ is a specific matrix of the form $[\star \mid \mathbf{I}]$ and the $\mathbf{M}_{i,b}$'s are the read-once branching program representing Ψ.

Starting from the witness encryption candidate, we also present an iO candidate for NC^1 circuits that appear to resist our rank attack as well as all prior attacks. In order to thwart the rank attack, our iO candidate necessarily reads each input bit $\omega(1)$ times. To then prevent mixed-input attacks, we rely on an extension of multiplicative bundling factors used in prior works that uses matrices instead of scalars.

We stress that an important design goal in these candidates is simplicity so as to facilitate the security analysis. We believe and anticipate that any attacks or partial security analysis for these candidates (perhaps in some weak idealized model cf. [22]) would enhance our understanding of witness encryption and obfuscation.

1.4 Discussion and Open problems

Perspective. The proposal of candidate multilinear maps [20] from lattice-type assumptions in 2013 has triggered a major paradigm shift in cryptography and enabled numerous cryptographic applications, most notably indistinguishability obfuscation [21]. Among the three multilinear maps candidates [19, 20, 23], GGH15 is the only one that has served as a basis for new cryptographic applications based on established lattice problems, as demonstrated in e.g. [13,26,27,38].

We believe that extending the safe settings of GGH15 (where security can be based on the LWE assumption), as explored in this work through the generalized GGH15 framework as well as both proofs and attacks, will pave the way towards new cryptographic constructions.

Open problems. We conclude with a number of open problems:

- Study the security of our candidate for witness encryption, either prove security under instance-independent assumptions, or find a direct attack on the scheme. For the former (i.e., prove security), the only proof strategy in the existing literature is to build and prove a so-called positional witness encryption scheme [24], for which the security definition allows the adversary to obtain encodings of zeroes. Unfortunately, the natural extensions of our candidate witness encryption scheme to a positional variant are susceptible to the rank attack in the presence of encodings of zeroes. For the latter (i.e., directly attack the scheme), all existing attack strategies on GGH15 encodings as used in our candidate require encodings of zeroes, which are not readily available in the witness encryption setting.
- Find a polynomial-time attack for iO candidates for branching programs where every input repeats $c = O(\lambda)$ time where λ is the security parameter. The analysis of known attacks, including our rank attack, yields running times that grow exponentially with c. There are possibilities that the analysis is not tight and the rank attack or prior attacks could in fact succeed with a smaller running time. However we have not detected such a phenomenon with experiments for small c.
- Note that all our candidate constructions are of the form: $\mathbf{A}_J, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ and evaluation/decryption proceeds by first computing $\mathbf{A}_J\mathbf{D}_{\mathbf{x}'} := \mathbf{A}_J \prod_{i=1}^{h} \mathbf{D}_{i,x'_i}$ for some $\mathbf{x}' \in \{0,1\}^h$. Consider the following restricted class of adversaries that only gets oracle access to $\mathbf{x}' \mapsto \mathbf{A}_J\mathbf{D}_{\mathbf{x}'}$ instead of $\mathbf{A}_j, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}$. Note that our rank attack as well as various mixed-input and zeroizing attacks can all be implemented using this restricted adversaries. Can we prove (or break) security of our witness encryption or iO candidates against this restricted class of adversaries under some reasonable instance-independent assumptions?

Independent work. Variants of our new lemmas related to lattice preimage sampling in Section 4 were presented in an independent work of Goyal, Koppula and Waters [28], for different purposes from ours. In [28], the lemmas were used as intermediate building blocks en route a collusion resistant traitor tracing scheme based on the LWE assumption.

1.5 Reader's guide

The rest of the article is organized as follows. Section 2 provides a more detailed overview of our techniques. Section 4 provides new lemmas related to lattice

preimage sampling. Section 5 gives a formal construction of the generalized-GGH15 encoding, the security notions, and the main technical proof that suffices for the applications. Due to the page limitation we leave the applications, the attacks, and the witness encryption and iO candidates in the full version available at https://eprint.iacr.org/2018/360.

2 Technical Overview

In this section, we present a more detailed overview of our techniques. We briefly describe the notation used in this overview and the paper, and refer the reader to Section 3 for more details. We use boldface upper-case and lower-case letters for matrices and vectors respectively. Given a bit-string $\mathbf{x} \in \{0,1\}^h$, we use $\mathbf{M}_{\mathbf{x}}$ to denote matrix subset product $\prod_{i=1}^{h} \mathbf{M}_{i,x_i}$. Given matrices $\mathbf{A}, \mathbf{B}$, we use $\mathbf{A}^{-1}(\mathbf{B})$ to denote a random low-norm Gaussian $\mathbf{D}$ satisfying $\mathbf{AD} = \mathbf{B} \bmod q$. Two probability distributions are connected by $\approx_s$ or $\approx_c$ if they are statistically close or computationally indistinguishable.

2.1 Generalized GGH15 Encodings

In this work, we think of GGH15 as encoding two collections of matrices, one collection is arbitrary and the other one is random, and computing some function γ of a subset product of these matrices; we refer to this as (generalized) γ-GGH15 encodings.[4] That is, the γ-GGH15 encoding takes as input two collections of matrices $\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathbf{S}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$, an additional matrix $\mathbf{A}_\ell$, and the output is a collection of matrices

$$\mathbf{A}_0, \{\mathbf{D}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$$

such that for all $\mathbf{x} \in \{0,1\}^\ell$, we have

$$\mathbf{A}_0 \cdot \mathbf{D}_{\mathbf{x}} \approx \gamma(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_\ell$$

where $\mathbf{M}_{\mathbf{x}}, \mathbf{D}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}$ denotes subset-product of matrices as defined earlier. Here,

$$\mathbf{M}_{i,b} \in \{0,1\}^{w\times w}, \mathbf{S}_{i,b} \in \mathbb{Z}^{n\times n}, \mathbf{A}_0, \mathbf{A}_\ell \in \mathbb{Z}_q^{\gamma(w,n)\times m}, \mathbf{D}_{i,b} \in \mathbb{Z}^{m\times m}.$$

Intuitively, we also want to hide the $\mathbf{M}_{i,b}$'s, which we will come back to after describing the choices for γ and the construction.

Choices for γ. There are several instantiations for γ in the literature [12, 13, 21, 23, 26, 29, 38]:

$$\gamma_\times(\mathbf{M},\mathbf{S}) = \mathbf{MS},\ \gamma_\otimes(\mathbf{M},\mathbf{S}) := \mathbf{M}\otimes\mathbf{S},\ \gamma_{\text{diag}}(\mathbf{M},\mathbf{S}) := \begin{pmatrix}\mathbf{M} & \\ & \mathbf{S}\end{pmatrix}$$

[4] See Remark 5.2 for a comparison with the original GGH15 encodings.

where the first $\gamma_\times$ requires working with rings so that multiplication commutes. More generally, for the construction, we require that γ be multiplicatively homomorphic, so that

$$\gamma(\mathbf{M}, \mathbf{S})\gamma(\mathbf{M}', \mathbf{S}') = \gamma(\mathbf{M}\mathbf{M}', \mathbf{S}\mathbf{S}')$$

as is clearly satisfied by the three instantiations above.

The γ-GGH15 construction. We briefly describe the construction of γ-GGH15 encodings implicit in [23], from the view-point of "cascaded cancellations" [2, 27, 30]. The starting point of the construction is to expand $\gamma(\mathbf{M}_\mathbf{x}, \mathbf{S}_\mathbf{x}) \cdot \mathbf{A}_\ell$ using multiplicative homomorphism as a matrix product

$$\gamma(\mathbf{M}_\mathbf{x}, \mathbf{S}_\mathbf{x}) \cdot \mathbf{A}_\ell = \prod_{i=1}^{\ell} \gamma(\mathbf{M}_{i,x_i}, \mathbf{S}_{i,x_i}) \cdot \mathbf{A}_\ell$$

Next, it randomizes the product by sampling random (wide, rectangular) matrices $\mathbf{A}_0, \ldots, \mathbf{A}_{\ell-1}$ over $\mathbb{Z}_q$ along with their trapdoors, and rewrites the product as a series of "cascaded cancellations":

$$\gamma(\mathbf{M}_\mathbf{x}, \mathbf{S}_\mathbf{x}) \cdot \mathbf{A}_\ell = \mathbf{A}_0 \cdot \prod_{i=1}^{\ell} \mathbf{A}_{i-1}^{-1}(\gamma(\mathbf{M}_{i,x_i}, \mathbf{S}_{i,x_i})\mathbf{A}_i)$$

where $\mathbf{A}_{i-1}^{-1}(\cdot)$ denotes random low-norm Gaussian pre-images as defined earlier. [5]

For functionality, it suffices to define $\mathbf{D}_{i,b}$ to be $\mathbf{A}_{i-1}^{-1}(\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\mathbf{A}_i)$, but that would not be sufficient to hide the underlying $\mathbf{M}_{i,b}$'s. Instead, the construction introduces additional error terms $\{\mathbf{E}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$, and defines[6]

$$\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}(\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b})$$

Observe that for all $\mathbf{x} \in \{0,1\}^\ell$, we have

$$\mathbf{A}_0 \cdot \mathbf{D}_\mathbf{x} \approx \gamma(\mathbf{M}_\mathbf{x}, \mathbf{S}_\mathbf{x}) \cdot \mathbf{A}_\ell$$

where $\approx$ refers to an additive error term that depends on $|\ \mathbf{D}_{i,b}\ |, |\ \mathbf{E}_{i,b}\ |, |\ \gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\ |$, which we require to be small.

[5] A reader who is familiar with Kilian's randomization for branching programs should notice the similarity. In Kilian's randomization, we randomize the product

$$\mathbf{M}_\mathbf{x} = \prod_{i=1}^{\ell} \mathbf{R}_{i-1}^{-1}\mathbf{M}_{i,x_i}\mathbf{R}_i$$

by picking random invertible matrices $\mathbf{R}_1, \ldots, \mathbf{R}_{\ell-1}$ along with $\mathbf{R}_0 = \mathbf{R}_\ell = \mathbf{I}$. Here, we replace the square matrices $\mathbf{R}_i$'s with wide rectangular matrices $\mathbf{A}_i$'s, and change from left-multiplying $\mathbf{R}_{i-1}^{-1}$ to sampling a random Gaussian preimage of $\mathbf{A}_{i-1}$.

[6] In the GGH15 terminology, $\mathbf{D}_{i,b}$ would be an encoding of $\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})$ relative to the path $i-1 \mapsto i$.

Semantic security. Following [13,26,27,38], we consider the following notion of semantic security for γ-GGH15 encodings, namely that

> **(semantic security.)** The output $(\mathbf{A}_0, \{\mathbf{D}_{i,b}\}_{i\in[\ell],b\in\{0,1\}})$ computationally hides $\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$. We only require that security holds "on average" over random $\{\mathbf{S}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{A}_\ell$.

Prior works [13,26,38] showed that the $\gamma_\otimes$-GGH15 encodings achieve semantic security if we restrict the $\mathbf{M}_{i,b}$'s to be permutation matrices. That is,

> **Informal Lemma.** Under the LWE assumption, we have that for all *permutation* matrices $\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$,
>
> $$(\mathbf{A}_0, \{\mathbf{D}_{i,0}, \mathbf{D}_{i,1}\}_{i\in[\ell]}) \approx_c (\mathbf{A}_0, \{\mathbf{V}_{i,0}, \mathbf{V}_{i,1}\}_{i\in[\ell]}) \tag{4}$$
>
> where $\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}((\mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b})$, and $\mathbf{V}_{i,0}, \mathbf{V}_{i,1}$ are random low-norm Gaussians.

As mentioned earlier in the introduction, the proof of security crucially relies on the fact that any permutation matrix $\mathbf{M}$, LWE tells us that $(\mathbf{A}, (\mathbf{M}\otimes\mathbf{S})\mathbf{A}+\mathbf{E}) \approx_c (\mathbf{A}, \mathbf{U})$, where $\mathbf{U}$ is uniformly random. We sketch the proof of the semantic security of $\gamma_\otimes$-GGH15 for $\ell = 1$, which extends readily to larger ℓ (here the major changes in the hybrid arguments are highlighted with boxes):

$$\begin{aligned}
&\left(\mathbf{A}_0, \{\mathbf{A}_0^{-1}((\mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b})\mathbf{A}_1 + \mathbf{E}_{1,b})\}_{b\in\{0,1\}}\right) \\
\approx_c &\left(\mathbf{A}_0, \{\mathbf{A}_0^{-1}(\boxed{\mathbf{U}_{1,b}})\}_{b\in\{0,1\}}\right) && \text{// LWE} \\
\approx_s &\left(\mathbf{A}_0, \{\boxed{\mathbf{V}_{1,b}}\}_{b\in\{0,1\}}\right) && \text{// GPV}
\end{aligned}$$

2.2 This work: semantic security for arbitrary matrices

Without further modifications, γ-GGH15 encoding does not achieve semantic security for arbitrary matrices. Concretely, given $\mathbf{A}_0, \mathbf{D}_{1,0}$, we can compute

$$\mathbf{A}_0 \cdot \mathbf{D}_{1,0} = \gamma(\mathbf{M}_{1,0}, \mathbf{S}_{1,0})\mathbf{A}_1 + \mathbf{E}_{1,0}$$

which might leak information about the structure of $\mathbf{M}_{1,0}$. In particular, we can distinguish between $\mathbf{M}_{1,0}$ being $\mathbf{I}^{w\times w}$ versus $\mathbf{0}^{w\times w}$ for all of $\gamma_\times, \gamma_\otimes, \gamma_{\text{diag}}$.

The key to our new cryptographic constructions for general branching programs is a new technical lemma asserting semantic security for γ_{diag}-GGH15 encodings with arbitrary matrices where we replace $\mathbf{A}_0$ with $\mathbf{J}\mathbf{A}_0$ for some wide bookend matrix $\mathbf{J}$ that statistically "loses" information about $\mathbf{A}_0$:

> **New Lemma, Informal.** Under the LWE assumption, we have that for all matrices $\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$ over $\mathbb{Z}$,
>
> $$(\mathbf{J}\mathbf{A}_0, (\mathbf{D}_{i,0}, \mathbf{D}_{i,1})_{i\in\ell}) \approx_c (\mathbf{J}\mathbf{A}_0, (\mathbf{V}_{i,0}, \mathbf{V}_{i,1})_{i\in\ell}) \tag{5}$$
>
> where $\mathbf{J}$ is any matrix of the form $[\star \mid \mathbf{I}]$, $\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}(\begin{pmatrix}\mathbf{M}_{i,b} & \\ & \mathbf{S}_{i,b}\end{pmatrix}\mathbf{A}_i + \mathbf{E}_{i,b})$, and $\mathbf{V}_{i,0}, \mathbf{V}_{i,1}$ are random low-norm Gaussians.

New proof technique We prove a stronger statement for the semantic security of γ_{diag}-GGH15, namely the semantic security holds even given $\mathbf{S}_{1,0}, \mathbf{S}_{1,1}, \ldots, \mathbf{S}_{\ell,0}, \mathbf{S}_{\ell,1}$ (but not $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$). Our proof departs significantly from the prior analysis – in particular, we will treat $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ as LWE secrets. Let $\overline{\mathbf{A}}_i, \underline{\mathbf{A}}_i$ denote the top and bottom parts of $\mathbf{A}$, and define $\overline{\mathbf{E}}_{i,b}, \underline{\mathbf{E}}_{i,b}$ analogously. This means that

$$\mathbf{A}_{i-1}^{-1}(\gamma_{\mathrm{diag}}(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\mathbf{A}_i + \mathbf{E}_{i,b}) = \mathbf{A}_{i-1}^{-1}\begin{pmatrix}\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b} \\ \mathbf{S}_{i,b}\underline{\mathbf{A}}_i + \underline{\mathbf{E}}_{i,b}\end{pmatrix}$$

We will use $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ as LWE secrets in the following order: $\underline{\mathbf{A}}_\ell, \ldots, \underline{\mathbf{A}}_1, \overline{\mathbf{A}}_0, \ldots, \overline{\mathbf{A}}_{\ell-1}$. We sketch the proof for $\ell = 1$ (and it extends readily to larger ℓ):

$$\begin{aligned}
&\left(\mathbf{J}\mathbf{A}_0, \{\mathbf{A}_0^{-1}\begin{pmatrix}\mathbf{M}_{1,b}\overline{\mathbf{A}}_1 + \overline{\mathbf{E}}_{1,b} \\ \mathbf{S}_{1,b}\underline{\mathbf{A}}_1 + \underline{\mathbf{E}}_{1,b}\end{pmatrix}\}_{b\in\{0,1\}}\right) \\
\approx_c &\left(\mathbf{J}\mathbf{A}_0, \{\boxed{\overline{\mathbf{A}}_0^{-1}\left(\mathbf{M}_{1,b}\overline{\mathbf{A}}_1 + \overline{\mathbf{E}}_{1,b}\right)}\}_{b\in\{0,1\}}\right) \\
\approx_s &\left(\boxed{\mathbf{U}_0}, \{\overline{\mathbf{A}}_0^{-1}\left(\mathbf{M}_{1,b}\overline{\mathbf{A}}_1 + \overline{\mathbf{E}}_{1,b}\right)\}_{b\in\{0,1\}}\right) \\
\approx_c &\left(\mathbf{U}_0, \{\boxed{\mathbf{V}_{1,b}}\}_{b\in\{0,1\}}\right)
\end{aligned}$$

where the notations and analysis of hybrid arguments are as follows

- The first $\approx_c$ follow from a more general statement, namely for all i and for any $\mathbf{Z}_{i,b}$, we have

$$\left\{\mathbf{A}_{i-1}^{-1}\begin{pmatrix}\mathbf{Z}_{i,b} \\ \mathbf{S}_{i,b}\underline{\mathbf{A}}_i + \underline{\mathbf{E}}_{i,b}\end{pmatrix}\right\}_{b\in\{0,1\}} \approx_c \left\{\overline{\mathbf{A}}_{i-1}^{-1}\left(\mathbf{Z}_{i,b}\right)\right\}_{b\in\{0,1\}}$$

  even if the distinguisher gets $\mathbf{A}_{i-1}, \mathbf{S}_{i,b}, \mathbf{Z}_{i,b}$. The proof of this statement follows by first applying LWE with $\underline{\mathbf{A}}_i$ as the secret[7] to deduce that

$$\{\mathbf{S}_{i,b}, \mathbf{S}_{i,b}\underline{\mathbf{A}}_i + \underline{\mathbf{E}}_{i,b}\}_{b\in\{0,1\}} \approx_c \{\mathbf{S}_{i,b}, \mathbf{U}_{i,b}\}_{b\in\{0,1\}}$$

  where the $\mathbf{U}_{i,b}$ matrices are uniformly random over $\mathbb{Z}_q$, followed by a new statistical lemma about trapdoor sampling which tells us that for all but negligibly many $\mathbf{A}_{i-1}$, we have that for all $\mathbf{Z}_{i,b}$,

$$\mathbf{A}_{i-1}^{-1}\begin{pmatrix}\mathbf{Z}_{i,b} \\ \mathbf{U}_{i,b}\end{pmatrix} \approx_s \overline{\mathbf{A}}_{i-1}^{-1}\left(\mathbf{Z}_{i,b}\right)$$

- The $\approx_s$ follows from the structure of $\mathbf{J}$, which implies $(\overline{\mathbf{A}}_0, \mathbf{J}\mathbf{A}_0) \approx_s (\overline{\mathbf{A}}_0, \mathbf{U}_0)$, where $\mathbf{U}_0$ is a uniformly random matrix.

[7] Here, we could have used $\mathbf{S}_{i,0}, \mathbf{S}_{i,1}$ as the LWE secrets and $\underline{\mathbf{A}}_i$ as the public matrix; however, this strategy would break down when $\mathbf{M}_{i,b}$ depends on $\mathbf{S}_{i,b}$, which is needed in the applications.

– The final $\approx_c$ follows from a more general statement, which says that under the LWE assumption, we have that for any $\mathbf{Z}$,

$$\mathbf{A}^{-1}(\mathbf{Z}+\mathbf{E}) \approx_c \mathbf{A}^{-1}(\mathbf{U})$$

where the distributions are over random choices of $\mathbf{A}, \mathbf{E}, \mathbf{U}$, provided $\mathbf{A}$ is hidden from the distinguisher. The proof uses the Bonsai technique [14]. Suppose $\mathbf{A}$ is of the form $[\mathbf{A}_1 \mid \mathbf{A}_2]$ where $\mathbf{A}_1$ is uniformly random, $\mathbf{A}_2$ sampled with a trapdoor. Then, we have via the Bonsai technique [14]:

$$\mathbf{A}^{-1}(\mathbf{Z}+\mathbf{E}) \approx_s \begin{pmatrix} -\mathbf{V} \\ \mathbf{A}_2^{-1}(\mathbf{A}_1\mathbf{V}+\mathbf{E}+\mathbf{Z})) \end{pmatrix}$$

where $\mathbf{V}$ is a random low-norm Gaussian. We then apply the LWE assumption to $(\mathbf{V}, \mathbf{A}_1\mathbf{V}+\mathbf{E})$ with $\mathbf{A}_1$ as the LWE secret. Once we replace $\mathbf{A}_1\mathbf{V}+\mathbf{E}$ with a uniformly random matrix, the rest of the proof follows readily from the standard GPV lemma.

Extension: combining $\gamma_\otimes, \gamma_{diag}$. For the applications to private constrained PRFs and lockable obfuscation, we will rely on $\gamma_{\otimes\text{diag}}$-GGH15 encodings, where

$$\gamma_{\otimes\text{diag}}(\mathbf{M}, \mathbf{S}) := \begin{pmatrix} \mathbf{M} \otimes \mathbf{S} & \\ & \mathbf{S} \end{pmatrix}$$

We observe that our proof of semantic security for γ_{diag} also implies semantic security for $\gamma_{\otimes\text{diag}}$, where we give out $\mathbf{J}\mathbf{A}_0$ instead of $\mathbf{A}_0$. This follows from the fact that our proof for γ_{diag} goes through even if the $\mathbf{M}_{i,b}$'s depend on the $\mathbf{S}_{i,b}$'s, since we treat the latter as public matrices when we invoke the LWE assumption.

2.3 New Cryptographic Constructions from LWE

Using $\gamma_{\otimes\text{diag}}$-GGH15 encodings and the proof that semantic security of $\gamma_{\otimes\text{diag}}$ holds for arbitrary $\mathbf{M}$ matrices, we are ready to construct private constrained PRFs and lockable obfuscation where the constraint/function can be recognized by arbitrary matrix branching programs. Here we briefly explain the private constrained PRF construction as an example.

Before that we recall some terminologies for matrix branching programs. In the overview, we focus on read-once matrix branching programs for notational simplicity, although our scheme works for general matrix branching programs with any input pattern and matrix pattern (possibly low-rank matrices). A (read-once) matrix branching program for a function $f_\Gamma : \{0,1\}^\ell \to \{0,1\}$ is specified by $\Gamma := \left\{ \{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{P}_0, \mathbf{P}_1 \right\}$ such that for all $\mathbf{x} \in \{0,1\}^\ell$,

$$\mathbf{M}_\mathbf{x} = \prod_{i=1}^{\ell} \mathbf{M}_{i,x_i} = \mathbf{P}_{f_\Gamma(x)}$$

We will work with families of branching programs $\{\Gamma\}$, which share the same $\mathbf{P}_0, \mathbf{P}_1$.

Private constrained PRFs We proceed to provide an overview of our construction of private constrained PRFs using $\gamma_{\otimes\text{diag}}$-GGH15 encodings. As a quick overview of a private constrained PRF, a private constrained PRF allows the PRF master secret key holder to derive a constrained key given a constraint predicate C. The constrained key is required to randomize the output on every input x s.t. $C(x) = 0$, preserve the output on every input x s.t. $C(x) = 1$. In addition, the constraint C is required to be hidden given the description of the constrained key.

Let $\mathbf{e}_i \in \{0,1\}^{1\times w}$ denotes the unit vector with the i^{th} coordinate being 1, the rest being 0. Consider a class of constraints recognizable by branching programs

$$\mathit{\Gamma}_C := \left\{ \left\{ \mathbf{M}_{i,b} \in \{0,1\}^{w\times w} \right\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{P}_0, \mathbf{P}_1 \right\},$$

where the target matrices $\mathbf{P}_0, \mathbf{P}_1$ satisfy $\mathbf{e}_1\mathbf{P}_0 = \mathbf{e}_1$, $\mathbf{e}_1\mathbf{P}_1 = \mathbf{0}^{1\times w}$.

We use $\gamma_{\otimes\text{diag}}$ to encode $\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$, which means for $i = 0, ..., \ell$, $\mathbf{A}_i \in \mathbb{Z}_q^{(nw+n)\times m}$. Denote $\underline{\mathbf{A}}_0$ as the bottom n rows of $\mathbf{A}_i$, $\overline{\mathbf{A}}_i$ as the top nw rows of $\mathbf{A}_i$. Inside $\overline{\mathbf{A}}_i$ let $\overline{\mathbf{A}}_i^{(j)}$ denote the $(j-1)n^{th}$ to jn^{th} rows of $\overline{\mathbf{A}}_i$, for $j \in [w]$.

Define the output of the normal PRF evaluation as

$$\mathbf{x} \mapsto \lfloor \mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell \rceil_p$$

where $\lfloor \cdot \rceil_p$ denotes the rounding-to-$\mathbb{Z}_p$ operation used in previous LWE-based PRFs, which we suppress in the rest of this overview for notational simplicity.

We set $\mathbf{J} := (\mathbf{e}_1 \otimes \mathbf{I} \mid \mathbf{I})$ so that $\mathbf{J}\cdot\mathbf{A}_0 = \overline{\mathbf{A}}_0^{(1)} + \underline{\mathbf{A}}_0$, then

$$\mathbf{J}\cdot\gamma_{\otimes\text{diag}}(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}})\cdot\mathbf{A}_\ell = ((\mathbf{e}_1\cdot\mathbf{M}_{\mathbf{x}})\otimes\mathbf{S}_{\mathbf{x}})\cdot\overline{\mathbf{A}}_\ell + \mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell = \begin{cases} \mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell & \text{if } f_{\mathit{\Gamma}}(\mathbf{x}) = 1 \\ \mathbf{S}_{\mathbf{x}}\overline{\mathbf{A}}_\ell^{(1)} + \mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell & \text{if } f_{\mathit{\Gamma}}(\mathbf{x}) = 0 \end{cases}$$

Given $\mathit{\Gamma}$, the constrained key is constructed as

$$\overline{\mathbf{A}}_0^{(1)} + \underline{\mathbf{A}}_0, (\mathbf{D}_{i,0}, \mathbf{D}_{i,0})_{i\in[\ell]}$$

where $(\mathbf{A}_0, \{\mathbf{D}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}) \leftarrow \text{GGHEnc}_{\otimes\text{diag}}(\{\mathbf{M}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathbf{S}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{A}_\ell)$.

The constrained evaluation on an input $\mathbf{x}$ gives

$$(\overline{\mathbf{A}}_0^{(1)} + \underline{\mathbf{A}}_0)\cdot\mathbf{D}_{\mathbf{x}} \approx \mathbf{J}\cdot\gamma_{\otimes\text{diag}}(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}})\cdot\mathbf{A}_\ell$$

which equals to $\mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell$ if $f_{\mathit{\Gamma}}(\mathbf{x}) = 1$, $\mathbf{S}_{\mathbf{x}}\overline{\mathbf{A}}_\ell^{(1)} + \mathbf{S}_{\mathbf{x}}\underline{\mathbf{A}}_\ell$ if $f_{\mathit{\Gamma}}(\mathbf{x}) = 0$.

A special case: private puncturable PRFs. A private puncturable PRF can be obtained by simply using branching program with 1×1 matrices (i.e. let $w = 1$). The punctured key at $\mathbf{x}^*$ is given by

$$\overline{\mathbf{A}}_0 + \underline{\mathbf{A}}_0, \{\mathbf{D}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$$

where

$$\mathbf{D}_{i,x_i^*} \leftarrow \mathbf{A}_{i-1}^{-1}\left(\begin{pmatrix}\mathbf{S}_{i,x_i^*} & \\ & \mathbf{S}_{i,x_i^*}\end{pmatrix}\mathbf{A}_i + \mathbf{E}_{i,x_i^*}\right), \mathbf{D}_{i,1-x_i^*} \leftarrow \mathbf{A}_{i-1}^{-1}\left(\begin{pmatrix}\mathbf{0} & \\ & \mathbf{S}_{i,1-x^*}\end{pmatrix}\mathbf{A}_i + \mathbf{E}_{i,1-x_i^*}\right).$$

The construction extends naturally to allow us to puncture at sets of points specified by a wildcard pattern $\{0,1,\star\}^\ell$.

Security. In the security proof, we will use the fact that whenever $f_\Gamma(\mathbf{x}) = 0$, constrained evaluation outputs $\boxed{\mathbf{S}_\mathbf{x}\overline{\mathbf{A}}_\ell^{(1)}} + \mathbf{S}_\mathbf{x}\underline{\mathbf{A}}_\ell$, so that the normal PRF output is masked by the boxed term. More formally, in the security game, the adversary gets a constrained key for Γ_C, and oracle access to a PRF evaluation oracle Eval. We consider the following sequence of games:

- Replace the output of the Eval oracle by
$$(\overline{\mathbf{A}}_0^{(1)} + \underline{\mathbf{A}}_0) \cdot \mathbf{D}_\mathbf{x} - \mathbf{S}_\mathbf{x} \cdot \overline{\mathbf{A}}_\ell^{(1)}$$
This is statistically indistinguishable from the real game, since $(\overline{\mathbf{A}}_0^{(1)} + \underline{\mathbf{A}}_0) \cdot \mathbf{D}_\mathbf{x} \approx \mathbf{S}_\mathbf{x} \cdot \overline{\mathbf{A}}_\ell^{(1)} + \mathbf{S}_\mathbf{x} \cdot \underline{\mathbf{A}}_\ell$, and the approximation disappears w.h.p. after rounding.
- Apply semantic security to replace $(\mathbf{D}_{i,0}, \mathbf{D}_{i,0})_{i\in[\ell]}$ with random. Here, we require that semantic security holds even if the distinguisher gets $\{\mathbf{S}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \overline{\mathbf{A}}_\ell$, where the latter are needed in order to compute $\mathbf{S}_\mathbf{x} \cdot \overline{\mathbf{A}}_\ell^{(1)}$. This implies constraint-hiding.
- Now, we can apply the BLMR analysis to deduce pseudorandomness of $\mathbf{S}_\mathbf{x} \cdot \overline{\mathbf{A}}_\ell^{(1)}$, where we treat $\overline{\mathbf{A}}_\ell^{(1)}$ as the seed of the BLMR PRF [7]. This implies pseudorandomness of the output of the Eval oracle.

3 Preliminaries

Notations and terminology. In cryptography, the security parameter (denoted as λ) is a variable that is used to parameterize the computational complexity of the cryptographic algorithm or protocol, and the adversary's probability of breaking security. An algorithm is "efficient" if it runs in (probabilistic) polynomial time over λ.

When a variable v is drawn randomly from the set S we denote as $v \xleftarrow{\$} S$ or $v \leftarrow U(S)$, sometimes abbreviated as v when the context is clear. We use $\approx_s$ and $\approx_c$ as the abbreviation for statistically close and computationally indistinguishable.

Let $\mathbb{R}, \mathbb{Z}, \mathbb{N}$ be the set of real numbers, integers and positive integers. Denote $\mathbb{Z}/(q\mathbb{Z})$ by $\mathbb{Z}_q$. The rounding operation $\lfloor a \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ is defined as multiplying a by p/q and rounding the result to the nearest integer.

For $n \in \mathbb{N}$, $[n] := \{1, ..., n\}$. A vector in $\mathbb{R}^n$ (represented in column form by default) is written as a bold lower-case letter, e.g. $\mathbf{v}$. For a vector $\mathbf{v}$, the i^{th}

component of $\mathbf{v}$ will be denoted by v_i. A matrix is written as a bold capital letter, e.g. $\mathbf{A}$. The i^{th} column vector of $\mathbf{A}$ is denoted $\mathbf{a}_i$. In this article we frequently meet the situation where a matrix $\mathbf{A}$ is partitioned into two pieces, one stacking over the other. We denote it as $\mathbf{A} = \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}}$. The partition is not necessarily even. We will explicitly mention the dimension when needed.

The length of a vector is the ℓ_p-norm $\|\mathbf{v}\|_p = (\sum v_i^p)^{1/p}$. The length of a matrix is the norm of its longest column: $\|\mathbf{A}\|_p = \max_i \|\mathbf{a}_i\|_p$. By default we use ℓ_2-norm unless explicitly mentioned. When a vector or matrix is called "small", we refer to its norm.

Subset products (of matrices) appear frequently in this article. For a given $h \in \mathbb{N}$, a bit-string $\mathbf{v} \in \{0,1\}^h$, we use $\mathbf{X}_{\mathbf{v}}$ to denote $\prod_{i\in[h]} \mathbf{X}_{i,v_i}$ (it is implicit that $\{\mathbf{X}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ are well-defined).

The tensor product (Kronecker product) for matrices $\mathbf{A} \in \mathbb{R}^{\ell\times m}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B}, \ldots, a_{1,m}\mathbf{B} \\ \ldots, \quad \ldots, \quad \ldots \\ a_{\ell,1}\mathbf{B}, \ldots, a_{\ell,m}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{\ell n\times mp}. \tag{6}$$

The rank of the resultant matrix satisfies $\mathsf{rank}(\mathbf{A} \otimes \mathbf{B}) = \mathsf{rank}(\mathbf{A}) \cdot \mathsf{rank}(\mathbf{B})$. For matrices $\mathbf{A} \in \mathbb{R}^{\ell\times m}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{m\times u}$, $\mathbf{D} \in \mathbb{R}^{p\times v}$,

$$(\mathbf{AC}) \otimes (\mathbf{BD}) = (\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{C} \otimes \mathbf{D}). \tag{7}$$

Lemma 3.1 (Leftover hash lemma). *Let $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ be a 2-universal hash function family. Then for any random variable $X \in \mathcal{X}$, for $\epsilon > 0$ s.t. $\log(|\mathcal{Y}|) \le H_\infty(X) - 2\log(1/\epsilon)$, the distributions*

$$(h, h(X)) \text{ and } (h, U(\mathcal{Y}))$$

are ϵ-statistically close.

3.1 Lattices background

Smoothing parameter. We recall the definition of smoothing parameter and some useful facts.

Definition 3.2 (Smoothing parameter [32]). *For any n-dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real $\sigma > 0$ such that $\rho_{1/\sigma}(\Lambda^* \setminus \{\mathbf{0}\}) \le \epsilon$.*

Lemma 3.3 (Smoothing parameter bound from [25]). *For any n-dimensional lattice $\Lambda(\mathbf{B})$ and for any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon(n)$ for which*

$$\eta_\epsilon(\Lambda) \le \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$$

Lemma 3.4 (Smooth over the cosets [25]). *Let Λ, Λ' be n-dimensional lattices s.t. $\Lambda' \subseteq \Lambda$. Then for any $\epsilon > 0$, $\sigma > \eta_\epsilon(\Lambda')$, and $\mathbf{c} \in \mathbb{R}^n$, we have*

$$\Delta(D_{\Lambda,\sigma,\mathbf{c}} \bmod \Lambda',\ U(\Lambda \bmod \Lambda')) < 2\epsilon$$

Lemma 3.5 ([32, 35]). *Let* $\mathbf{B}$ *be a basis of an* n*-dimensional lattice* Λ*, and let* $\sigma \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\log n)$*, then* $\Pr_{\mathbf{x} \leftarrow D_{\Lambda,\sigma}}[\|\mathbf{x}\| \geq \sigma \cdot \sqrt{n} \vee \mathbf{x} = \mathbf{0}] \leq \mathrm{negl}(n)$.

Lemma 3.6 ([9, 25]). *There is a p.p.t. algorithm that, given a basis* $\mathbf{B}$ *of an* n*-dimensional lattice* $\Lambda(\mathbf{B})$*,* $\mathbf{c} \in \mathbb{R}^n$*,* $\sigma \geq \|\tilde{\mathbf{B}}\| \cdot \sqrt{\ln(2n+4)/\pi}$*, outputs a sample from* $D_{\Lambda,\sigma,\mathbf{c}}$.

Learning with errors. We recall the learning with errors problem.

Definition 3.7 (Decisional learning with errors (LWE) [37]). *For* $n, m \in \mathbb{N}$ *and modulus* $q \geq 2$*, distributions for secret vectors, public matrices, and error vectors* $\theta, \pi, \chi \subseteq \mathbb{Z}_q$*. An LWE sample is obtained from sampling* $\mathbf{s} \leftarrow \theta^n$*,* $\mathbf{A} \leftarrow \pi^{n \times m}$*,* $\mathbf{e} \leftarrow \chi^m$*, and outputting* $(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}^T \pmod q)$.

We say that an algorithm solves $\mathsf{LWE}_{n,m,q,\theta,\pi,\chi}$ *if it distinguishes the LWE sample from a random sample distributed as* $\pi^{n \times m} \times U(\mathbb{Z}_q^{1 \times m})$ *with probability bigger than* $1/2$ *plus non-negligible.*

Lemma 3.8 (Regularity of Ajtai function [37]). *Fix a constant* $c > 1$*, let* $m \geq cn \log q$*. Then for all but* $q^{\frac{-(c-1)n}{4}}$ *fraction of* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, the statistical distance between a random subset-sum of the columns of* $\mathbf{A}$ *and uniform over* $\mathbb{Z}_q^n$ *is less than* $q^{\frac{-(c-1)n}{4}}$.

Lemma 3.9 (Standard form [9, 33, 34, 37]). *Given* $n \in \mathbb{N}$*, for any* $m = \mathrm{poly}(n)$*,* $q \leq 2^{\mathrm{poly}(n)}$*. Let* $\theta = \pi = U(\mathbb{Z}_q)$*,* $\chi = D_{\mathbb{Z},\sigma}$ *where* $\sigma \geq 2\sqrt{n}$*. If there exists an efficient (possibly quantum) algorithm that breaks* $\mathsf{LWE}_{n,m,q,\theta,\pi,\chi}$*, then there exists an efficient (possibly quantum) algorithm for approximating* SIVP *and* GapSVP *in the* ℓ_2 *norm, in the worst case, to within* $\tilde{O}(nq/\sigma)$ *factors.*

We drop the subscripts of LWE when referring to standard form of LWE with the parameters specified in Lemma 3.9. In this article we frequently use the following variant of LWE that is implied by the standard form.

Lemma 3.10 (LWE with small public matrices [7]). *For* n, m, q, σ *chosen as was in Lemma 3.9,* $\mathsf{LWE}_{n',m,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$ *is as hard as* $\mathsf{LWE}_{n,m,q,U(\mathbb{Z}_q),U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma}}$ *for* $n' \geq 2 \cdot n \log q$.

Trapdoor and preimage sampling. Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, denote the kernel lattice of $\mathbf{A}$ as

$$\Lambda^{\perp}(\mathbf{A}) := \{\mathbf{c} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{c} = 0^n \pmod q\}.$$

Given any $\mathbf{y} \in \mathbb{Z}_q^n$, $\sigma > 0$, we use $\mathbf{A}^{-1}(\mathbf{y}, \sigma)$ to denote the distribution of a vector $\mathbf{d}$ sampled from $D_{\mathbb{Z}^m,\sigma}$ conditioned on $\mathbf{A}\mathbf{d} = \mathbf{y} \pmod q$. We sometimes suppress σ when the context is clear.

Lemma 3.11 ([1, 3, 31]). *There is a p.p.t. algorithm* $\mathsf{TrapSam}(1^n, 1^m, q)$ *that, given the modulus* $q \geq 2$*, dimensions* n*,* m *such that* $m \geq 2n \log q$*, outputs* $\mathbf{A} \approx_s U(\mathbb{Z}_q^{n \times m})$ *with a trapdoor* τ.

Following Lemmas 3.6 and 3.11,

Lemma 3.12. *There is a p.p.t. algorithm that for $\sigma \geq 2\sqrt{n \log q}$, given $(\mathbf{A}, \tau) \leftarrow \mathsf{TrapSam}(1^n, 1^m, q)$, $\mathbf{y} \in \mathbb{Z}_q^n$, outputs a sample from $\mathbf{A}^{-1}(\mathbf{y}, \sigma)$.*

Lemma 3.13 ([25]). *For all but negligible probability over $(\mathbf{A}, \tau) \leftarrow \mathsf{TrapSam}(1^n, 1^m, q)$, for sufficiently large $\sigma \geq 2\sqrt{n \log q}$, the following distributions are efficiently samplable and statistically close:*

$$\left\{\mathbf{A}, \mathbf{x}, \mathbf{y} : \mathbf{y} \leftarrow U(\mathbb{Z}_q^n), \mathbf{x} \leftarrow \mathbf{A}^{-1}(\mathbf{y}, \sigma)\right\} \approx_s \left\{\mathbf{A}, \mathbf{x}, \mathbf{y} : \mathbf{x} \leftarrow D_{\mathbb{Z}^m, \sigma}, \mathbf{y} = \mathbf{A}\mathbf{x}\right\}.$$

Lemma 3.14 (Bonsai technique [14]). *Let $n, m, m_1, m_2, q \in \mathbb{N}, \sigma \in \mathbb{R}$ satisfy $m = m_1 + m_2$, $m_2 \geq 2n \log q$, $\sigma > 2\sqrt{n \log q}$. For any $\mathbf{y} \in \mathbb{Z}_q^n$, the following two distributions are efficiently samplable and statistically close.*

1. *Let $(\mathbf{A}, \tau) \leftarrow \mathsf{TrapSam}(1^n, 1^m, q)$, $\mathbf{d} \leftarrow \mathbf{A}^{-1}(\mathbf{y}, \sigma)$. Output $(\mathbf{A}, \mathbf{d})$.*
2. *Let $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{n \times m_1})$, $(\mathbf{A}_2, \tau_2) \leftarrow \mathsf{TrapSam}(1^n, 1^{m_2}, q)$; $\mathbf{d}_1 \leftarrow D_{\mathbb{Z}^{m_1}, \sigma}$, $\mathbf{d}_2 \leftarrow \mathbf{A}_2^{-1}(\mathbf{y} - \mathbf{A}_1 \cdot \mathbf{d}_1, \sigma)$. Let $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2]$, $\mathbf{d} = [\mathbf{d}_1^T, \mathbf{d}_2^T]^T$. Output $(\mathbf{A}, \mathbf{d})$.*

4 New Lemmas on Preimage Sampling

In this section, we present new lemmas related to lattice preimage sampling. These lemmas are essential to the proof of semantic security for non-permutation branching programs, as outlined in Section 2.2.

The first is a statistical lemma which states that for all but negligibly many matrix $\mathbf{A}$ (with proper dimensions), for any matrix $\mathbf{Z}$, the following two distributions are statistically indistinguishable:

$$\left(\mathbf{A}, \mathbf{A}^{-1}\begin{pmatrix}\mathbf{Z} \\ \mathbf{U}\end{pmatrix}\right) \approx_s \left(\mathbf{A}, \overline{\mathbf{A}}^{-1}(\mathbf{Z})\right)$$

where the distributions are over random choices of a matrix $\mathbf{U}$ and probability distributions $\mathbf{A}^{-1}(\cdot)$ and $\overline{\mathbf{A}}^{-1}(\cdot)$. This is in essence an extension of the trapdoor sampling lemma from Gentry, Peikert and Vaikuntanathan [25].

The second is a computational lemma which states that for any matrix $\mathbf{Z}$, the following two distributions are computationally indistinguishable:

$$\mathbf{A}^{-1}(\mathbf{Z} + \mathbf{E}) \approx_c \mathbf{A}^{-1}(\mathbf{U})$$

where the distributions are over random private choices of $\mathbf{A}, \mathbf{E}$ and $\mathbf{U}$ and the coins of $\mathbf{A}^{-1}(\cdot)$. The computational indistinguishability relies on the hardness of the decisional learning with errors (LWE) problem.

4.1 The Statistical Lemma

We prove the above statistical lemma for vectors; the setting for matrices follow readily via a hybrid argument.

Lemma 4.1. *Let $\epsilon > 0$. Given $\sigma \in R^+$, $n', n, m, q \in \mathbb{N}$. For all but a $q^{-2n'}$ fraction of $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n' \times m}$, all but a q^{-2n} fraction of $\underline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, let $\mathbf{A} := \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}}$. For $\sigma > \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$, $m \geq 9(n'+n)\log q$. For a fixed $\mathbf{z} \in \mathbb{Z}_q^{n'}$, for $\mathbf{u} \leftarrow U(\mathbb{Z}_q^n)$, we have*

$$\mathbf{A}^{-1}(\begin{pmatrix}\mathbf{z}\\ \mathbf{u}\end{pmatrix}, \sigma) \text{ and } \overline{\mathbf{A}}^{-1}(\mathbf{z}, \sigma)$$

are 2ϵ-statistically close.

Proof. We need two lemmas to assist the proof of Lemma 4.1.

Lemma 4.2. *Let $c > 9$. For $n', n, m, q \in \mathbb{N}$ such that $m \geq c(n'+n)\log q$. For all but $q^{-2n'}$ fraction of $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n' \times m}$, all but q^{-2n} fraction of $\underline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, we have $\left\{\underline{\mathbf{A}} \cdot \mathbf{x} \mid \mathbf{x} \in \{0,1\}^m \cap \Lambda^\perp(\overline{\mathbf{A}})\right\} = \mathbb{Z}_q^n$.*

Proof. From Lemma 3.8, we have for all but $q^{-2n'}$ fraction of $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n' \times m}$

$$\left| \Pr_{\mathbf{x} \in \{0,1\}^m}[\overline{\mathbf{A}} \cdot \mathbf{x} = 0^{n'}] - q^{-n'} \right| < 2q^{-2n'} \Rightarrow \Pr_{\mathbf{x} \in \{0,1\}^m}[\overline{\mathbf{A}} \cdot \mathbf{x} = 0^{n'}] > 0.99 \cdot q^{-n'} \quad (8)$$

Let $\mathbf{x} \leftarrow U(\{0,1\}^m \cap \Lambda^\perp(\overline{\mathbf{A}}))$, we have $H_\infty(\mathbf{x}) > m - 2n'\log q$. For $\delta > 0$, by setting $m \geq n\log q + 2n'\log q + 2\log(1/\delta)$, we have that for $\underline{\mathbf{A}} \leftarrow U(\mathbb{Z}_q^{n \times m})$,

$$(\underline{\mathbf{A}}, \underline{\mathbf{A}} \cdot \mathbf{x}) \text{ and } (\underline{\mathbf{A}}, U(\mathbb{Z}_q^n))$$

are δ-statistically close following leftover hash lemma (cf. Lemma 3.1).

Then Lemma 4.2 follows by setting $\delta = q^{-4n}$ and take a union bound for $\underline{\mathbf{A}}$.

Lemma 4.3. *For $n', n, m, q \in \mathbb{N}$, $\sigma > 0$. $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n' \times m}$, $\underline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$. Assuming the columns of $\mathbf{A} := \binom{\overline{\mathbf{A}}}{\underline{\mathbf{A}}}$ generate $\mathbb{Z}_q^{n'+n}$. For any vectors $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{z} \in \mathbb{Z}_q^{n'}$, and $\mathbf{c} \in \mathbb{Z}^m$ where $\mathbf{A} \cdot \mathbf{c} = \binom{\mathbf{z}}{\mathbf{u}} \mod q$. The conditional distribution D of $\mathbf{x} \leftarrow \mathbf{c} + D_{\Lambda^\perp(\overline{\mathbf{A}}), \sigma, -\mathbf{c}}$ given $\underline{\mathbf{A}}\mathbf{x} = \mathbf{u} \mod q$ is exactly $\mathbf{c} + D_{\Lambda^\perp(\mathbf{A}), \sigma, -\mathbf{c}}$.*

Proof. Observe that the support of D is $\mathbf{c} + \Lambda^\perp(\mathbf{A})$. We compute the distribution D: for all $\mathbf{x} \in \mathbf{c} + \Lambda^\perp(\mathbf{A})$,

$$D(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\mathbf{c} + \Lambda^\perp(\mathbf{A}))} = \frac{\rho_{\sigma,-\mathbf{c}}(\mathbf{x} - \mathbf{c})}{\rho_{\sigma,-\mathbf{c}}(\Lambda^\perp(\mathbf{A}))} = D_{\Lambda^\perp(\mathbf{A}), \sigma, -\mathbf{c}}(\mathbf{x} - \mathbf{c}). \quad (9)$$

Finally from Lemma 3.4, let $\Lambda = \Lambda^\perp(\overline{\mathbf{A}})$, $\Lambda' = \Lambda^\perp(\mathbf{A})$, we have $\Lambda' \subseteq \Lambda$. Since $\sigma > \eta_\epsilon(\Lambda')$, $D_{\Lambda^\perp(\overline{\mathbf{A}}), \sigma, -\mathbf{c}}$ is 2ϵ-statistically close to uniform over the cosets of the quotient group $(\Lambda^\perp(\overline{\mathbf{A}})/\Lambda^\perp(\mathbf{A}))$. The rest of the proof of Lemma 4.1 follows Lemma 4.3 and Lemma 4.2.

4.2 The Computational Lemma

Lemma 4.4. *Given $n, m, k, q \in \mathbb{N}, \sigma \in \mathbb{R}$ such that $n, m, k \in \mathrm{poly}(\lambda)$, $m \geq 4n \log q$, $\sigma \geq 2\sqrt{n \log q}$. For arbitrary matrix $\mathbf{Z} \in \mathbb{Z}_q^{n \times k}$, the following two distributions are computationally indistinguishable assuming $\mathsf{LWE}_{m,k,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$.*

Dist. 1 *Let $\mathbf{A}, \tau \leftarrow \mathsf{TrapSam}(1^n, 1^m, q)$, $\mathbf{E} \leftarrow D_{\mathbb{Z},\sigma}^{n \times k}$. Sample $\mathbf{D} \leftarrow \mathbf{A}^{-1}(\mathbf{Z} + \mathbf{E}, \sigma)$ using τ. Output $\mathbf{D}$.*

Dist. 2 *Sample $\mathbf{D} = D_{\mathbb{Z},\sigma}^{m \times k}$. Output $\mathbf{D}$.*

Proof. We prove a stronger statement where the computational indistinguishability holds even when $\mathbf{Z}$ is given to the adversary. The proof uses the Bonsai technique [14]. Let $m = m_1 + m_2$ such that $m_1, m_2 \geq 2n \log q$. We introduce 2 intermediate distributions,

Dist. 1.1 Let $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{n \times m_1})$, $(\mathbf{A}_2, \tau_2) \leftarrow \mathsf{TrapSam}(1^n, 1^{m_2}, q)$. Sample $\mathbf{D}_1 \leftarrow D_{\mathbb{Z},\sigma}^{m_1 \times k}$. Let $\mathbf{E} \leftarrow D_{\mathbb{Z},\sigma}^{n \times k}$, sample $\mathbf{D}_2 \leftarrow \mathbf{A}_2^{-1}((-\mathbf{A}_1 \cdot \mathbf{D}_1 + \mathbf{E} + \mathbf{Z}), \sigma)$ using τ_2. Let $\mathbf{D} := \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}$. Output $\mathbf{D}$.

Dist. 1.2 Let $\mathbf{A}_1 \leftarrow U(\mathbb{Z}_q^{n \times m_1})$, $(\mathbf{A}_2, \tau_2) \leftarrow \mathsf{TrapSam}(1^n, 1^{m_2}, q)$. Sample $\mathbf{D}_1 \leftarrow D_{\mathbb{Z},\sigma}^{m_1 \times k}$. Let $\mathbf{U} \leftarrow U(\mathbb{Z}_q^{n \times k})$, sample $\mathbf{D}_2 \leftarrow \mathbf{A}_2^{-1}((\mathbf{U} + \mathbf{Z}), \sigma)$ using τ_2. Let $\mathbf{D} := \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}$. Output $\mathbf{D}$.

Then Distributions 1 and 1.1 are statistically close following Lemma 3.14. Distributions 2 and 1.2 are statistically close following Lemma 3.13.

It remains to prove that Dist. 1.1 $\approx_c$ Dist. 1.2 assuming $\mathsf{LWE}_{m_1,k,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$. This follows by taking $(\mathbf{D}_1, -\mathbf{A}_1 \cdot \mathbf{D}_1 + \mathbf{E})$ as the LWE sample, where $\mathbf{A}_1$ is the concatenation of n independent uniform secret vectors, $\mathbf{D}_1$ is the low-norm public matrix and $\mathbf{E}$ is the error matrix.

Formally, suppose there exists a p.p.t. distinguisher A for Dist. 1.1 and Dist. 1.2, we build a distinguisher A' for $\mathsf{LWE}_{m_1,k,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$. Given the challenge sample $(\mathbf{D}_1, \mathbf{Y}_1)$, A' runs $(\mathbf{A}_2, \tau_2) \leftarrow \mathsf{TrapSam}(1^n, 1^{m_2}, q)$, samples $\mathbf{D}_2 \leftarrow \mathbf{A}_2^{-1}((\mathbf{Y}_1 + \mathbf{Z}), \sigma)$ using τ_2, send $\mathbf{D} := \begin{pmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{pmatrix}$ to the adversary A. If A says it is from Dist. 1.1, then A' chooses "LWE"; if A says Dist. 1.2, then A' chooses "random". The success probability of A' is same to the success probability of A.

5 Generalized GGH15 Encodings

We present the abstraction of *generalized GGH15 encodings*. The abstraction includes a construction framework and definitions of security notions.

5.1 The construction framework

We begin with a description of the construction:

Construction 5.1 (γ-GGH15 Encodings) *The randomized algorithm* ggh.encode *takes the following inputs*

- *Parameters*[8] 1^λ, $h, n, m, q, t, w \in \mathbb{N}$, $\sigma \in \mathbb{R}^*$ *and the description of a distribution* χ *over* $\mathbb{Z}$.
- *A function* $\gamma : \mathbb{Z}^{w\times w} \times \mathbb{Z}^{n\times n} \to \mathbb{Z}^{t\times t}$.
- *Matrices* $\left\{\mathbf{M}_{i,b} \in \mathbb{Z}_{i,b}^{w\times w}\right\}_{i\in[h],b\in\{0,1\}}, \left\{\mathbf{S}_{i,b} \in \mathbb{Z}_{i,b}^{n\times n}\right\}_{i\in[h],b\in\{0,1\}}$.
- *A matrix* $\mathbf{A}_h \in \mathbb{Z}_q^{t\times m}$.

It generates the output as follows

- *Samples* $\{\mathbf{A}_i, \tau_i \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)\}_{i\in\{0,1,\ldots,h-1\}}$.
- *Samples* $\{\mathbf{E}_{i,b} \leftarrow \chi^{t\times m}\}_{i\in[h],b\in\{0,1\}}$.
- *For* $i \in [h], b \in \{0,1\}$, *let* $\hat{\mathbf{S}}_{i,b} := \gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})$, *then samples*

$$\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}(\hat{\mathbf{S}}_{i,b} \cdot \mathbf{A}_i + \mathbf{E}_{i,b}, \sigma)$$

  using τ_{i-1}.
- *Outputs* $\mathbf{A}_0$, $\{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}$.

We require γ to be multiplicatively homomorphic:

$$\gamma(\mathbf{M}, \mathbf{S}) \cdot \gamma(\mathbf{M}', \mathbf{S}') = \gamma(\mathbf{M} \cdot \mathbf{M}', \mathbf{S} \cdot \mathbf{S}')$$

Remark 5.2 (Comparison with GGH15). The goal of the original GGH15 graded encodings in [23] was to emulate the functionality provided by multi-linear maps with respect to some underlying directed acyclic graph. The basic unit of the construction is an encoding of a low-norm matrix $\hat{\mathbf{S}}$ along $\mathbf{A}_0 \mapsto \mathbf{A}_1$ given by $\mathbf{A}_0^{-1}(\hat{\mathbf{S}}\mathbf{A}_1 + \mathbf{E})$, where $\hat{\mathbf{S}}$ must be drawn from some high-entropy distribution to achieve any meaningful notion of security.

Following [13, 26, 27, 38], we think of $\hat{\mathbf{S}}$ as being deterministically derived from an arbitrary low-norm matrix $\mathbf{M}$ and a random low-norm matrix $\mathbf{S}$ via some fixed function γ given by $\gamma : (\mathbf{M}, \mathbf{S}) \mapsto \mathbf{M} \otimes \mathbf{S}$ in the afore-mentioned constructions. Here, we make γ an explicit parameter to the construction, so that we obtain a family of constructions parameterized by γ, which we refer to as the "γ-GGH15 encodings".

Looking ahead to Section 5.2, another advantage of decoupling $\hat{\mathbf{S}}$ into $\mathbf{M}$ and $\mathbf{S}$ is that we can now require semantic security for arbitrary inputs $\mathbf{M}$ and random choices of $\mathbf{S}$ (more precisely, arbitrary $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ and random $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$), as considered in [38]. Moreover, this notion of semantic security can be achieved under the LWE assumption for some specific γ and classes of matrices $\mathbf{M}$. Here, we make explicit the idea that semantic security should be defined with respect to some fixed auxiliary function aux of the matrices $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_0, \ldots, \mathbf{A}_h$.

[8] In the rest of the presentation, these parameters are omitted in the input of ggh.encode.

Functionality. The next lemma captures the functionality provided by the construction, namely that for all $\mathbf{x} \in \{0,1\}^h$,

$$\mathbf{A}_0 \cdot \mathbf{D}_{\mathbf{x}} \approx \gamma(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_h$$

Lemma 5.3 (Functionality of γ-GGH15 encodings). *Suppose γ is multiplicatively homomorphic. For all inputs to the Construction 5.1 s.t. $\sigma > \Omega(\sqrt{t \log q})$, $m > \Omega(t \log q)$, $\|\chi\| \le \sigma$; we have for all $\mathbf{x} \in \{0,1\}^h$, with all but negligible probability over the randomness in Construction 5.1,*

$$\|\mathbf{A}_0 \cdot \mathbf{D}_{\mathbf{x}} - \gamma(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_h\|_\infty \le h \cdot \left(m\sigma \cdot \max_{i,b} \|\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\| \right)^h.$$

Proof. Recall $\hat{\mathbf{S}}_{i,b} = \gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})$. It is straight-forward to prove by induction that for all $h' = 0, 1, \ldots, h$:

$$\mathbf{A}_0 \cdot \prod_{k=1}^{h'} \mathbf{D}_{k,x_k} = \left(\prod_{i=1}^{h'} \hat{\mathbf{S}}_{i,x_i} \right) \mathbf{A}_{h'} + \sum_{j=1}^{h'} \left(\left(\prod_{i=1}^{j-1} \hat{\mathbf{S}}_{i,x_i} \right) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \tag{10}$$

The base case $h' = 0$ holds trivially. The inductive step uses the fact that for all $h' = 1, \ldots, h$:

$$\mathbf{A}_{h'-1} \cdot \mathbf{D}_{h',x_{h'}} = \hat{\mathbf{S}}_{h',x_{h'}} \cdot \mathbf{A}_{h'} + \mathbf{E}_{h',x_{h'}}$$

From the homomorphic property of γ we can deduce that

$$\prod_{i=1}^{h} \hat{\mathbf{S}}_{i,x_i} = \prod_{i=1}^{h} \gamma(\mathbf{M}_{i,x_i}, \mathbf{S}_{i,x_i}) = \gamma(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}})$$

Finally, we bound the error term as follows:

$$\begin{aligned}
\|\mathbf{A}_0 \cdot \mathbf{D}_{\mathbf{x}} - \gamma(\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}) \cdot \mathbf{A}_h\|_\infty &= \left\| \sum_{j=1}^{h} \left(\prod_{i=1}^{j-1} (\hat{\mathbf{S}}_{i,x_i}) \cdot \mathbf{E}_{j,x_j} \cdot \prod_{k=j+1}^{h} \mathbf{D}_{k,x_k} \right) \right\|_\infty \\
&\le h \cdot \sqrt{t} \cdot \sigma \cdot \left(\sqrt{t} \cdot \max_{i,b} \|\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\| \cdot \sigma \cdot \sqrt{m} \right)^{h-1} \\
&\le h \cdot \left(\max_{i,b} \|\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\| \cdot \sigma \cdot m \right)^h
\end{aligned}$$

Looking ahead, in the applications we will set the parameters to ensure that the threshold $B := h \cdot (m\sigma \cdot \max_{i,b} \|\gamma(\mathbf{M}_{i,b}, \mathbf{S}_{i,b})\|)^h$ is relatively small compared to the modulus q.

Remark 5.4 (Dimensions of $\mathbf{A}_h$). The construction and many analyses in this article can be obviously generalized to the cases where the dimensions of matrices are more flexible. As an example, the matrix $\mathbf{A}_h$ can be chosen from $\mathbb{Z}_q^t$ instead of $\mathbb{Z}_q^{t \times m}$ (as a result, $\mathbf{D}_{h,0}$, $\mathbf{D}_{h,1}$ are from $\mathbb{Z}^m$ instead of $\mathbb{Z}^{m \times m}$). This change maintains necessary functionalities, reduce the size of the construction, and is (more importantly) necessary for one of the proofs in the paper. For the ease of presentation we keep all the $\mathbf{A}$ matrices with the same dimension, all the $\mathbf{D}$ matrices with the same dimension, and mention the exceptions as they arise.

Interesting γ functions. We are interested in the following 3 γ functions:

- $\gamma_\otimes : \{0,1\}^{w\times w} \times \mathbb{Z}^{n\times n} \to \mathbb{Z}^{(wn)\times(wn)}$, $\mathbf{M}, \mathbf{S} \mapsto \mathbf{M} \otimes \mathbf{S}$.
  $\gamma_\otimes$ with permutation matrices $\mathbf{M}$ was introduced and studied in [13, 26, 27, 38].
- $\gamma_{\text{diag}} : \mathbb{Z}^{w\times w} \times \mathbb{Z}^{n\times n} \to \mathbb{Z}^{(w+n)\times(w+n)}$, $\mathbf{M}, \mathbf{S} \mapsto \begin{pmatrix} \mathbf{M} & \\ & \mathbf{S} \end{pmatrix}$.
  γ_{diag} is implicit in the constructions in [21, 29] and is central to the security analysis in this work.
- $\gamma_{\otimes\text{diag}} : \{0,1\}^{w\times w} \times \mathbb{Z}^{n\times n} \to \mathbb{Z}^{(wn+n)\times(wn+n)}$, $\mathbf{M}, \mathbf{S} \mapsto \begin{pmatrix} \mathbf{M}\otimes\mathbf{S} & \\ & \mathbf{S} \end{pmatrix}$.
  We introduce $\gamma_{\otimes\text{diag}}$ in this work, which would be central to the applications in this paper.

Note that all of the three γ functions are multiplicatively homomorphic and norm-preserving.

5.2 Security notions

Intuitively, semantic security says that for all $\mathbf{M}$, the output of the γ-GGH15 encodings

$$\mathbf{A}_0, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}$$

hides $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$, for random choices of $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ and $\mathbf{A}_0, \ldots, \mathbf{A}_h$. We consider a more general notion parameterized by some fixed function aux of $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_0, \ldots, \mathbf{A}_h$, and we require that $\mathsf{aux}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ hides $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$.

Definition 5.5 (Semantic security with auxiliary input). *We say that the γ-GGH15 encodings satisfies semantic security with auxiliary input* aux *for a family of matrices $\mathcal{M} \subseteq \mathbb{Z}^{w\times w}$ if for all $\{\mathbf{M}_{i,b} \in \mathcal{M}\}_{i\in[h],b\in\{0,1\}}$, we have*

$$\mathsf{aux}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}} \approx_c \mathsf{aux}, \left\{(D_{\mathbb{Z},\sigma}^{m\times m})_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$$

where

$$\mathbf{S}_{i,b} \leftarrow D_{\mathbb{Z},\sigma}^{n\times n}, \mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m}), \{\mathbf{D}_{i,b}\} \leftarrow \mathsf{ggh.encode}(\gamma, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_h)$$

and aux *is a fixed function of $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_0, \ldots, \mathbf{A}_h$.*

Remark 5.6 ($\gamma_\otimes$-GGH encodings with permutation matrices). Canetti and Chen [13] (also, [26, 38]) showed that the $\gamma_\otimes$-GGH15 encoding satisfies semantic security with auxiliary input $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_h)$ for the family of permutation matrices in $\{0,1\}^{w\times w}$.

We can prove that the $\gamma_\otimes$-GGH15 encoding satisfies semantic security with auxiliary input $(\mathbf{A}_0, \{\mathbf{S}_{i,b}\}_{i\in[\ell],b\in\{0,1\}})$ for the family of permutation matrices in $\{0,1\}^{w\times w}$, by using the LWE assumption with the $\mathbf{S}_{i,b}$ as the public matrices.

Such a proof requires a multiplicative blow-up (of roughly $O(\log q)$) in the dimensions of the $\mathbf{S}_{i,b}$ matrices. One of the advantages of using the $\mathbf{S}$ matrices as the public matrices is that we can use the same $\mathbf{S}_0, \mathbf{S}_1$ across all the h levels, similar to the PRF construction in [7].

5.3 Semantic security for $\gamma_{\mathbf{diag}}$-GGH15 and $\gamma_{\otimes\mathbf{diag}}$-GGH15 encodings

In this section, we prove semantic security of the γ_{diag}-GGH15 and $\gamma_{\otimes\text{diag}}$-GGH15 encodings in Construction 5.1 under the LWE assumption, where

$$\gamma_{\text{diag}}(\mathbf{M}, \mathbf{S}) = \begin{pmatrix} \mathbf{M} & \\ & \mathbf{S} \end{pmatrix}, \ \gamma_{\otimes\text{diag}}(\mathbf{M}, \mathbf{S}) = \begin{pmatrix} \mathbf{M} \otimes \mathbf{S} & \\ & \mathbf{S} \end{pmatrix}.$$

In fact, we show that this holds given auxiliary input about $\mathbf{A}_0$ and $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$.

$\mathbf{S}$-*dependent security.* Concretely, we will derive semantic security of $\gamma_{\otimes\text{diag}}$ from that of γ_{diag} by showing that the construction γ_{diag} satisfies a stronger notion of $\mathbf{S}$-dependent security where the matrices $\{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}$ may depend on $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}$:

Definition 5.7 ($\mathbf{S}$-dependent semantic security with auxiliary input). *We say that the γ-GGH15 encodings satisfies $\mathbf{S}$-dependent semantic security with auxiliary input* aux *for a family of matrices $\mathcal{M} \subseteq \mathbb{Z}^{w\times w}$ if for every polynomial-size circuit $f : (\mathbb{Z}^{n\times n})^{2h} \to \mathcal{M}^{2h}$, we have*

$$\mathsf{aux}, \{\mathbf{D}_{i,b}\}_{i\in[h],b\in\{0,1\}} \approx_c \mathsf{aux}, \left\{(D_{\mathbb{Z},\sigma}^{m\times m})_{i,b}\right\}_{i\in[h],b\in\{0,1\}}$$

where

$$\mathbf{S}_{i,b} \leftarrow D_{\mathbb{Z},\sigma}^{n\times n}, \mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m}), \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}} = f(\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}),$$
$$\{\mathbf{D}_{i,b}\} \leftarrow \mathsf{ggh.encode}(\gamma, \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_h)$$

and aux *is a fixed function of $\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{A}_0, \ldots, \mathbf{A}_h$.*

Theorem 5.8 ($\mathbf{S}$-dependent semantic security of $\gamma_{\mathbf{diag}}$). *Assuming $\mathsf{LWE}_{n,2m,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$, the γ_{diag}-GGH15 encodings in Construction 5.1 satisfies $\mathbf{S}$-dependent semantic security for $\mathcal{M} = \mathbb{Z}^{w\times w}$ with auxiliary input*

$$\mathsf{aux} = \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{J}\cdot\mathbf{A}_0, \overline{\mathbf{A}}_h$$

where $\overline{\mathbf{A}}_h \in \mathbb{Z}_q^{w\times m}$ is the top w rows of $\mathbf{A}_h$ and $\mathbf{J} \in \{0,1\}^{n\times(t-n)} \mid \mathbf{I}^{n\times n}$.

Remark 5.9 (Necessity of $\mathbf{JA}_0$). Ideally, we would liked to have shown that semantic security holds with auxiliary input $\mathbf{A}_0$ (as opposed to $\mathbf{JA}_0$). However, such a statement is false for general $\mathcal{M} \in \mathbb{Z}^{w\times w}$. Concretely, given $\mathbf{A}_0, \mathbf{D}_{1,0}$, we can compute $\mathbf{A}_0 \cdot \mathbf{D}_{1,0}$ which leaks information about the structure of $\mathbf{M}_{1,0}$. In particular, we can distinguish between $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$.

As an immediate corollary, we then have:

Corollary 5.10 (semantic security of $\gamma_{\otimes\mathbf{diag}}$). *Assuming* $\mathsf{LWE}_{n,2m,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$, *the* $\gamma_{\otimes diag}$*-GGH15 encodings in Construction 5.1 satisfies semantic security for* $\mathcal{M} = \mathbb{Z}^{w\times w}$ *with auxiliary input*

$$\mathsf{aux} = \{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}}, \mathbf{J}\cdot\mathbf{A}_0, \overline{\mathbf{A}}_h$$

where $\overline{\mathbf{A}}_h \in \mathbb{Z}_q^{wn\times m}$ *is the top* wn *rows of* $\mathbf{A}_h$ *and* $\mathbf{J} \in \{0,1\}^{n\times(t-n)} \mid \mathbf{I}^{n\times n}$.

5.4 Proof of the main theorem

Proof (Proof of Theorem 5.8). For $t, n, w \in \mathbb{N}$ such that $t = w + n$. For any matrix $\mathbf{X} \in \mathbb{Z}^{t\times *}$, let $\mathbf{X} = \begin{pmatrix}\overline{\mathbf{X}}\\ \underline{\mathbf{X}}\end{pmatrix}$, where $\overline{\mathbf{X}} \in \mathbb{Z}^{w\times *}$, $\underline{\mathbf{X}} \in \mathbb{Z}^{n\times *}$. For the sake of completeness we spell out the details of the real and simulated distributions which will be proven indistinguishable.

The real and simulated distributions. In the real distribution the adversary is given

$$\mathbf{J}\cdot\mathbf{A}_0, \left\{\boxed{\mathbf{D}_{i,b}}, \mathbf{S}_{i,b}, \mathbf{M}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}, \overline{\mathbf{A}}_h$$

where

- $\{\mathbf{A}_i, \tau_i \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)\}_{i\in\{0,1,\ldots,h-1\}}, \mathbf{A}_h \leftarrow U(\mathbb{Z}_q^{t\times m})$
- $\mathbf{S}_{i,b} \leftarrow D_{\mathbb{Z},\sigma}^{n\times n}, \{\mathbf{M}_{i,b}\}_{i\in[h],b\in\{0,1\}} \leftarrow f(\{\mathbf{S}_{i,b}\}_{i\in[h],b\in\{0,1\}})$
- $\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}\begin{pmatrix}\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b}\\ \mathbf{S}_{i,b}\underline{\mathbf{A}}_i + \underline{\mathbf{E}}_{i,b}\end{pmatrix}, \mathbf{E}_{i,b} \leftarrow \chi^{t\times m}$

The simulated distribution is generated in the same way except that the adversary is given

$$\mathbf{J}\cdot\mathbf{A}_0, \left\{\boxed{\mathbf{V}_{i,b}}, \mathbf{S}_{i,b}, \mathbf{M}_{i,b}\right\}_{i\in[h],b\in\{0,1\}}, \overline{\mathbf{A}}_h$$

where $\mathbf{V}_{i,b} \leftarrow D_{\mathbb{Z},\sigma}^{m\times m}$.

To show that the real distribution is computationally indistinguishable from the simulated one, we introduce the following intermediate distributions.

Distributions 1.i, for $i \in \{h+1, h, ..., 1\}$. Let Distribution $1.(h+1)$ be identical to the real distribution. For $i = h$ down to 1, let Distributions $1.i$ be the same to Distributions $1.(i+1)$, except that $\mathbf{A}_{i-1}$, $\mathbf{D}_{i,0}$, $\mathbf{D}_{i,1}$ are sampled differently. Let $(\overline{\mathbf{A}}_{i-1}, \tau_{i-1}) \leftarrow \mathsf{TrapSam}(1^w, 1^m, q)$, $\underline{\mathbf{A}}_{i-1} \leftarrow U(\mathbb{Z}_q^{n\times m})$. Sample $\mathbf{D}_{i,b} \leftarrow \overline{\mathbf{A}}_{i-1}^{-1}((\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b}), \sigma)$ using τ_{i-1}, $b \in \{0,1\}$.

Distributions 2.0. Distribution 2.0 is sampled identically to Distribution 1.1, except that $\mathbf{J}\cdot\mathbf{A}_0$ is replaced with a uniformly random matrix $\mathbf{U} \xleftarrow{\$} \mathbb{Z}^{n\times m}$. Since $\mathbf{J} \in \{0,1\}^{n\times(t-n)} \mid \mathbf{I}^{n\times n}$, $\mathbf{U} \approx_s \mathbf{J}\cdot\mathbf{A}_0$ for $\mathbf{A}_0, \tau_0 \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$ due to Lemma 3.11.

Distributions 2.j, for $j \in \{1, ..., h\}$. For $j = 1, 2, ..., h$, let Distributions 2.j be the same to Distributions 2.$(j-1)$, except that $\mathbf{D}_{j,0}$, $\mathbf{D}_{j,1}$ are sampled simply from $D_{\mathbb{Z},\sigma}^{m\times m}$. Note that Dist. 2.$h$ is identical to the simulated distribution, except that in Dist. 2.h, $\mathbf{U} \xleftarrow{\$} \mathbb{Z}^{n\times m}$ is in the place where $\mathbf{J}\cdot\mathbf{A}_0$ is in the simulated distribution, so they are statistically close again due to Lemma 3.11.

The sequence. We will show that:

$$\text{Real} = 1.(h+1) \approx_c 1.h \approx_c \cdots \approx_c 1.1 \approx_s 2.0 \approx_c 2.1 \approx_c \cdots \approx_c 2.h \approx_s \text{Simulated}$$

In particular, the $\approx_c$'s will rely on the LWE assumption, using $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ as LWE secrets in the following order: $\underline{\mathbf{A}}_\ell, \ldots, \underline{\mathbf{A}}_1, \overline{\mathbf{A}}_0, \ldots, \overline{\mathbf{A}}_{\ell-1}$.

Lemma 5.11. *For $i \in [h]$, Distribution 1.$(i+1)$ $\approx_c$ Distribution 1.i assuming $\mathsf{LWE}_{n,2n,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$.*

Roughly speaking, we will show that for all $i \in [h]$,

$$\left\{\mathbf{A}_{i-1}^{-1}\begin{pmatrix}\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b}\\ \mathbf{S}_{i,b}\underline{\mathbf{A}}_i + \underline{\mathbf{E}}_{i,b}\end{pmatrix}\right\}_{b\in\{0,1\}} \approx_c \left\{\overline{\mathbf{A}}_{i-1}^{-1}(\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b})\right\}_{b\in\{0,1\}}$$

where the distinguisher is also given $\mathbf{A}_{i-1}, \tau_{i-1}, \mathbf{S}_{i,0}, \mathbf{S}_{i,1}, \mathbf{M}_{i,0}, \mathbf{M}_{i,1}, \overline{\mathbf{A}}_i$, but not $\underline{\mathbf{A}}_i$, so that we can treat $\underline{\mathbf{A}}_i$ as a LWE secret, cf. Lemma 4.4.

Proof. We introduce an intermediate distribution 1.i^*, which is generated in the same way as Distributions 1.$(i+1)$, except that $\mathbf{D}_{i,0}$, $\mathbf{D}_{i,1}$ are sampled as:

$$\mathbf{D}_{i,b} \leftarrow \mathbf{A}_{i-1}^{-1}\left(\begin{pmatrix}\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b}\\ \mathbf{U}_{i,b}\end{pmatrix}, \sigma\right), b \in \{0,1\}.$$

where $(\mathbf{U}_{i,0}, \mathbf{U}_{i,1}) \leftarrow U(\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m})$.

The intermediate distribution 1.i^* is statistically close to Distribution 1.i due to Lemma 4.1. It remains to prove that 1.i^* is computationally indistinguishable from Distribution 1.$(i+1)$. This follows Lemma 3.10, by treating $\underline{\mathbf{A}}_i$ as the LWE secret, and $\mathbf{S}_{i,0}, \mathbf{S}_{i,1}$ as the public matrices.

Formally, if there's an adversary A that distinguishes Distributions 1.$(i+1)$ and 1.i^*, we build a distinguisher A' for $\mathsf{LWE}_{n,2n,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$ as follows. Once given the LWE challenge

$$\mathbf{S}_{i,0}, \mathbf{S}_{i,1}, \underline{\mathbf{Y}}_{i,0}, \underline{\mathbf{Y}}_{i,1}$$

where $\mathbf{S}_{i,0}, \mathbf{S}_{i,1}$ are the low-norm public matrices, $\underline{\mathbf{Y}}_{i,0}, \underline{\mathbf{Y}}_{i,1}$ are either the $\mathsf{LWE}_{n,2n,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$ samples with the common secret $\underline{\mathbf{A}}_i \leftarrow U(\mathbb{Z}_q^{n\times m})$, or independent uniform samples from $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m}$. The LWE distinguisher A' proceeds as follows:

1. Sample $\left\{\mathbf{S}_{k,b} \leftarrow D_{\mathbb{Z},\sigma}^{n\times n}\right\}_{k\in[h],k\neq i,b\in\{0,1\}}$.
2. For $k \in [h], b \in \{0,1\}$, compute $\mathbf{M}_{k,b} \in \mathbb{Z}^{w\times w}$ using $f(\{\mathbf{S}_{k,b}\}_{k\in[h],b\in\{0,1\}})$.

3. For $k \in \{0, 1, ..., i-1\}$, sample $\mathbf{A}_k, \tau_k \leftarrow \mathsf{TrapSam}(1^t, 1^m, q)$. For $k \in \{i, i+1, ..., h-1\}$, sample $\overline{\mathbf{A}}_k, \bar{\tau}_k \leftarrow \mathsf{TrapSam}(1^w, 1^m, q)$. Sample $\overline{\mathbf{A}}_h \leftarrow U(\mathbb{Z}_q^{t \times m})$.
4. For $k \in [h], b \in \{0,1\}$, samples

$$\mathbf{D}_{k,b} \leftarrow \begin{cases} \mathbf{A}_{k-1}^{-1}\left(\begin{smallmatrix}\mathbf{M}_{k,b}\overline{\mathbf{A}}_k + \overline{\mathbf{E}}_{k,b} \\ \mathbf{S}_{k,b}\underline{\mathbf{A}}_k + \underline{\mathbf{E}}_{k,b}\end{smallmatrix}\right) & \text{using } \tau_{k-1} \text{ if } k \le i-1 \\ \mathbf{A}_{i-1}^{-1}\left(\begin{smallmatrix}\mathbf{M}_{i,b}\overline{\mathbf{A}}_i + \overline{\mathbf{E}}_{i,b} \\ \underline{\mathbf{Y}}_{i,b}\end{smallmatrix}\right) & \text{using } \tau_{i-1} \text{ if } k = i \\ \overline{\mathbf{A}}_{k-1}^{-1}(\mathbf{M}_{k,b}\overline{\mathbf{A}}_k + \overline{\mathbf{E}}_{k,b})) & \text{using } \bar{\tau}_{k-1} \text{ if } k \ge i+1 \end{cases}$$

with standard deviation σ.

The LWE distinguisher A' then sends

$$\mathbf{J} \cdot \mathbf{A}_0, \left\{\boxed{\mathbf{D}_{k,b}}, \mathbf{S}_{k,b}, \mathbf{M}_{k,b}\right\}_{k \in [h], b \in \{0,1\}}, \overline{\mathbf{A}}_h.$$

to the adversary A. If A says it is Dist. $1.(i+1)$, it corresponds to the LWE samples with low-norm public matrices; if A says Dist. $1.i^*$, it corresponds to the uniform distribution.

Lemma 5.12. *For $j \in [h]$, Distribution $2.(j-1) \approx_c$ Distributions $2.j$ assuming $\mathsf{LWE}_{m,2m,q,U(\mathbb{Z}_q),D_{\mathbb{Z},\sigma},D_{\mathbb{Z},\sigma}}$.*

Roughly speaking, we will show that for all $j \in [h]$,

$$\left\{\overline{\mathbf{A}}_{j-1}^{-1}(\mathbf{M}_{j,b}\overline{\mathbf{A}}_j + \overline{\mathbf{E}}_{j,b})\right\}_{b \in \{0,1\}} \approx_c \left\{D_{\mathbb{Z},\sigma}^{m \times m}\right\}_{b \in \{0,1\}}$$

where the distinguisher is also given $\mathbf{M}_{j,0}, \mathbf{M}_{j,1}, \overline{\mathbf{A}}_j$, but not $\overline{\mathbf{A}}_{j-1}$, so as to trigger Lemma 4.4.

Proof. For $j \in [h]$, suppose there exists an adversary A that distinguishes Distributions $2.(j-1)$ and $2.j$, we build a distinguisher A' for Distributions 1 and 2 in Lemma 4.4 as follows. Given challenging samples

$$\mathbf{D}_{j,0} \mid \mathbf{D}_{j,1} \in \mathbb{Z}^{m \times 2m}$$

either obtained from $\overline{\mathbf{A}}_{j-1}^{-1}(\left[\mathbf{M}_{j,0}\overline{\mathbf{A}}_j + \overline{\mathbf{E}}_{j,0} \mid \mathbf{M}_{j,1}\overline{\mathbf{A}}_j + \overline{\mathbf{E}}_{j,1}\right])$ which corresponds to Dist. 1 in Lemma 4.4 (by treating $\left[\mathbf{M}_{j,0}\overline{\mathbf{A}}_j \mid \mathbf{M}_{j,1}\overline{\mathbf{A}}_j\right]$ as the arbitrary matrix $\mathbf{Z}$); or from $D_{\mathbb{Z},\sigma}^{m \times 2m}$ which corresponds to Dist. 2 in Lemma 4.4. The distinguisher A' proceeds as follows:

1. For $k \in [h], b \in \{0,1\}$, sample $\mathbf{S}_{k,b} \leftarrow D_{\mathbb{Z},\sigma}^{n \times n}$.
2. For $k \in [h], b \in \{0,1\}$, compute $\mathbf{M}_{k,b} \in \mathbb{Z}^{w \times w}$ using $f(\{\mathbf{S}_{k,b}\}_{k \in [h], b \in \{0,1\}})$.
3. For $k \in \{j, j+1, ..., h-1\}$, sample $\overline{\mathbf{A}}_k, \bar{\tau}_k \leftarrow \mathsf{TrapSam}(1^w, 1^m, q)$. Sample $\overline{\mathbf{A}}_h \leftarrow U(\mathbb{Z}_q^{t \times m})$.
4. For $k \in \{1, 2, ..., j-1, j+1, ..., h\}, b \in \{0,1\}$, samples

$$\mathbf{D}_{k,b} \leftarrow \begin{cases} D_{\mathbb{Z},\sigma}^{m \times m} \text{ if } k \le j-1 \\ \overline{\mathbf{A}}_{k-1}^{-1}(\mathbf{M}_{k,b}\overline{\mathbf{A}}_k + \overline{\mathbf{E}}_{k,b}, \sigma) \text{ using } \bar{\tau}_{k-1} \text{ if } k \ge j+1 \end{cases}.$$

5. Sample $\mathbf{U} \leftarrow U(\mathbb{Z}_q^{n \times m})$.

A' then sends

$$\mathbf{U}, \left\{\boxed{\mathbf{D}_{k,b}}, \mathbf{S}_{k,b}, \mathbf{M}_{k,b}\right\}_{k\in[h],b\in\{0,1\}}, \overline{\mathbf{A}}_h.$$

to the adversary A. Note that A' correctly produce the output without $\overline{\mathbf{A}}_{j-1}$. So if A determines that the samples are from Distribution $2.(j-1)$, A' chooses Dist. 1 in Lemma 4.4; if A determines that the samples are from Distribution $2.j$, A' chooses Dist. 2 in Lemma 4.4.

Theorem 5.8 follows from Lemmas 5.11 and 5.12.

Acknowledgments

Y.C. is supported by the NSF MACS project. Part of this work was done while visiting ENS. V.V. is supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, the NEC Corporation and a Steven and Renee Finn Career Development Chair from MIT. This work was also sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236. H.W. is supported by ERC Project aSCEND (H2020 639554). Part of this work was done while visiting CQT.

References

1. Miklós Ajtai. Generating hard instances of the short basis problem. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999.
2. Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (ring-)LWE. In *CRYPTO, Part II*, pages 659–680, 2016.
3. Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, 2011.
4. Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. In *ICALP*, volume 80 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
5. David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. In Juris Hartmanis, editor, *STOC*, pages 1–5. ACM, 1986.
6. Dan Boneh, Sam Kim, and Hart William Montgomery. Private puncturable prfs from standard lattice assumptions. In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 415–445, 2017.

7. Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 410–428, 2013.
8. Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.
9. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.
10. Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. In *CRYPTO, Part II*, pages 416–434, 2013.
11. Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained prfs (and more) from LWE. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, pages 264–302, 2017.
12. Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In *ITCS*, pages 147–156. ACM, 2016.
13. Ran Canetti and Yilei Chen. Constraint-hiding constrained PRFs for NC^1 from LWE. In *EUROCRYPT 2017, Part I*, pages 446–476, 2017.
14. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of cryptology*, 25(4):601–639, 2012.
15. Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In *EUROCRYPT (3)*, volume 10212 of *Lecture Notes in Computer Science*, pages 278–307, 2017.
16. Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *EUROCRYPT (1)*, volume 9056 of *LNCS*, pages 3–12. Springer, 2015.
17. Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Cryptanalysis of GGH15 multilinear maps. In *CRYPTO (2)*, volume 9815 of *LNCS*, pages 607–628. Springer, 2016.
18. Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. In *Public Key Cryptography (1)*, volume 10174 of *Lecture Notes in Computer Science*, pages 41–58. Springer, 2017.
19. Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.
20. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.
21. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013.
22. Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In *TCC (B2)*, volume 9986 of *Lecture Notes in Computer Science*, pages 241–268, 2016.
23. Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *TCC 2015, Part II*, pages 498–527, 2015.
24. Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In *CRYPTO (1)*, volume 8616 of *Lecture Notes in Computer Science*, pages 426–443. Springer, 2014.

25. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
26. Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *FOCS*, pages 612–621, 2017.
27. Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In *EUROCRYPT (2)*, pages 528–557, 2017.
28. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *STOC*, 2018.
29. Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing BP-obfuscation using graph-induced encoding. In *ACM CCS*, pages 783–798, 2017.
30. Venkata Koppula and Brent Waters. Circular security separations for arbitrary length cycles from LWE. In *CRYPTO (2)*, pages 681–700, 2016.
31. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.
32. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing*, 37(1):267–302, 2007.
33. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.
34. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *STOC*, pages 461–473. ACM, 2017.
35. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography*, pages 145–166. Springer, 2006.
36. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.
37. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
38. Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *FOCS*, pages 600–611, 2017.