Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions

Daniele Micciancio Petros Mol

University of California, San Diego
Computer Science and Engineering department
9500 Gilman Dr, Mail code 0404, La Jolla, CA 92093
Email: {daniele, pmol}@cs.ucsd.edu

September 26, 2011

Abstract

We study under what conditions the conjectured one-wayness of the knapsack function (with polynomially bounded inputs) over an arbitrary finite abelian group implies that the output of the function is pseudorandom, i.e., computationally indistinguishable from a uniformly chosen group element. Previous work of Impagliazzo and Naor (J. Cryptology 9(4):199-216, 1996) considers only specific families of finite abelian groups and uniformly chosen random *binary* inputs. Our work substantially extends previous results and provides a much more general reduction that applies to arbitrary finite abelian groups and input distributions with polynomially bounded coefficients. As an application of the new result, we give *sample preserving* search-to-decision reductions for the Learning With Errors (LWE) problem, introduced by Regev (J. ACM 56(6):34, 2009) and widely used in lattice-based cryptography.

Keywords: bounded knapsacks, LWE, Fourier analysis, pseudorandomness, sample complexity

Contents

1 Introduction **3**
1.1 Bounded Knapsacks over Abelian Groups 4
1.2 Pseudorandomness of the LWE Function 5

2 Preliminaries **6**
2.1 Probability 6
2.2 Groups and Knapsack Function Families. 7
2.3 Lattices and Gaussian Distributions 9
2.4 Fourier Analysis and Learning 11

3 Pseudorandomness of Knapsack Functions **12**
3.1 From One-wayness to Unpredictability 13
3.2 From Unpredictability to Pseudorandomness 15

4 Implications and applications **19**
4.1 Specific Groups and Input Distributions 20
4.2 Applications to LWE 22

5 Open Problems **24**

6 Acknowledgments **24**

1 Introduction

The Learning With Errors (LWE) problem, introduced by Regev in [38], is the problem of recovering a secret n-dimensional integer vector $\mathbf{s} \in \mathbb{Z}_q^n$, given a collection of perturbed random equations $\mathbf{a}_i\mathbf{s} \approx b_i$ where $\mathbf{a}_i \in \mathbb{Z}_q^n$ is chosen uniformly at random and $b_i = \mathbf{a}_i\mathbf{s} + e_i$ for some small, randomly chosen error term e_i. In recent years, LWE has been used to substantially expand the scope of lattice based cryptography, yielding solutions to many important cryptographic tasks, including public key encryption secure against passive [38, 24, 36] and active attacks [37, 35], (hierarchical) identity based encryption [17, 12, 1, 2], digital signatures [17, 12], oblivious transfer protocols [36], several forms of leakage resilient encryption [5, 6, 13, 20], (fully) homomorphic encryption [16, 15, 11] and more. The versatility of the LWE problem in the construction of a plethora of cryptographic applications is due in large part to its pseudorandomness properties: as proved in [38], if recovering (with high[1] probability) the secret $\mathbf{s}$ from the samples $(\mathbf{a}_i, \mathbf{a}_i\mathbf{s} + e_i)$ is computationally hard, then it is also hard to distinguish the LWE samples $(\mathbf{a}_i, \mathbf{a}_i\mathbf{s} + e_i)$ from randomly chosen ones $(\mathbf{a}_i, b_i)$ where the $b_i \in \mathbb{Z}_q$ are uniformly and independently distributed. In other words, any efficient distinguisher (between the LWE and the uniform distributions) can be turned into an inverter that recovers the secret $\mathbf{s}$, with only a polynomial slow-down.

On the theoretical side, cryptography based on LWE is supported by deep worst-case/average-case connections [38, 35], showing that any algorithm that solves LWE (on the average) can be efficiently converted into a (quantum) algorithm that solves the hardest (worst-case) instances of several famous lattice approximation problems which are believed to be intractable, like approximating the minimum distance of a lattice within factors that grow polynomially in the dimension, and various other related problems [27]. It should be remarked that, while such proofs of security based on worst-case lattice assumptions provide a solid theoretical justification for the probability distributions used in LWE cryptography, they are quite loose in their parameter settings. As a result, these reductions are hardly useful in practice, and in order to get meaningful estimates on the hardness of breaking LWE cryptography, it is generally more useful and appropriate to *conjecture* the average-case hardness of solving LWE, and use that as a starting point. (See [29, pp. 446-450] for a discussion of this and related issues.) In fact, all recent work aimed at determining appropriate key sizes and security parameters [33, 26, 40] follows this approach, and investigates experimentally the concrete hardness of solving LWE on the average.

In light of that, LWE is best formulated as the problem of inverting the one-way function family (indexed by a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, where m is the number of samples) that maps the secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error vector $\mathbf{e} \in \mathbb{Z}_q^m$ to $\mathbf{As} + \mathbf{e} \in \mathbb{Z}_q^m$. The search-to-decision reduction of [38] shows that if the LWE function family is one-way, then it is also a good pseudorandom generator. However, the reduction in [38] somehow hides an important detail: the value of m for which the function is assumed to be one-way is much higher than (still polynomially related to) the value of m for which the pseudorandomness of the function's output is proven.

While theoretical results based on worst-case lattice problems are fairly insensitive to the value of m, (i.e., the number of samples used in the LWE instance,) this number becomes more important and relevant when considering concrete attacks on the average-case hardness of LWE. For instance, recent algorithmic results [8] show that, when the errors e_i are sufficiently small, the LWE problem can be solved in subexponential (or even polynomial) time, provided a sufficiently large number of samples is available. Therefore, for certain ranges of the parameters, the number of available samples can have a significant impact on the computational hardness of the LWE problem. Likewise, some lattice attacks perform better in practice when given many (typically $\omega(n)$) samples [33]. However, LWE-based encryption schemes (e.g., see [26]) typically expose only a small number of samples (say, comparable to the dimension n of the LWE secret $\mathbf{s}$) during key generation and encryption. Fixing the number of available samples to a small value may significantly reduce the effectiveness of attacks, and increase our confidence in the concrete security of the schemes.

It should also be noted that when the number of available samples is above a certain threshold, one can efficiently generate an arbitrary number of additional samples [17, 6, 39], but at the cost of increasing the

[1] Due to the self-reducibility properties of the LWE problem, here "high" can be equivalently interpreted in a variety of ways, ranging from "nonnegligible" to "very close to 1".

magnitude of the errors. So, for certain other ranges of the parameters the impact of increasing the number of samples may not be as critical as in [8]. Still, even in such situations, using a large number of samples comes at the price of lowering the quality of the samples, which can negatively impact the concrete security and performance of LWE-based cryptographic functions.

This motivates the following question: how big of a blow-up in the number of samples is required to prove the pseudorandomness of the LWE output distribution, based on the conjectured hardness of the LWE search (secret recovery) problem? The main result of this paper is that, perhaps surprisingly, in most common applications of LWE in cryptography, no such blow-up is necessary at all: there is a *sample preserving* reduction from solving the search LWE problem (with nonnegligible success probability) to the problem of distinguishing the LWE distribution from random (with nonnegligible advantage). At the core of our result is a general theorem about the pseudorandomness of bounded knapsacks over arbitrary groups that substantially extends previous work in the area and might be of independent interest.

ROADMAP. In the next subsections, we give an informal overview of our results and techniques for bounded knapsack functions, and their application to the LWE problem. We review the background required in the rest of the paper in Section 2. Section 3 is devoted to the proof of our main technical result, the search-to-decision reduction for bounded knapsack families defined over arbitrary abelian groups. In Section 4, we describe applications of our main theorem to a broad range of bounded knapsack families, and explain how our result implies *sample-preserving* search-to-decision reductions for the LWE problem. We conclude in Section 5 with some interesting open problems.

1.1 Bounded Knapsacks over Abelian Groups

Let $(G, +)$ be a finite abelian group, and $\mathbf{g} = (g_1, \ldots, g_m) \in G^m$ a sequence of group elements chosen uniformly at random. The group elements $\mathbf{g}$ define a knapsack function $f_{\mathbf{g}}(\mathbf{x})$ that maps the integer vector $\mathbf{x} \in \mathbb{Z}^m$ to the group element $f_{\mathbf{g}}(\mathbf{x}) = \sum_i x_i g_i$. If the input $\mathbf{x}$ is restricted to vectors with small entries, then for a large variety of groups G, $f_{\mathbf{g}}$ is conjectured to be a one-way function family, i.e., a family of functions that are hard to invert on average when the key $\mathbf{g}$ is chosen uniformly at random. For example, when the input $\mathbf{x}$ is restricted to the set $\{0,1\}^m$ of binary vectors, inverting $f_{\mathbf{g}}$ is the famous subset-sum problem, which is conjectured to be hard to solve on average, and has been extensively studied in cryptography. In a classic paper [22], Impagliazzo and Naor showed that for some specific, but representative, choices of the group G, if the subset-sum function is one-way, then it is also a pseudorandom generator, i.e., it is computationally hard to distinguish $(\mathbf{g}, f_{\mathbf{g}}(\mathbf{x}))$ from a uniformly random element of G^{m+1}, when $\mathbf{g} \in G^m$ and $\mathbf{x} \in \{0,1\}^m$ are chosen uniformly at random. We generalize the results of [22] in two respects:

- We consider functions over arbitrary finite groups G. Only groups of the form $\mathbb{Z}_N$ were considered in [22], and for two specific (but representative) choices of N (prime and power of 2).
- We consider input coefficients x_i that take values from a set $\{0, \ldots, s\}$ (or, more generally $\{-s, \ldots, s\}$) for any (*polynomially bounded*) s. Moreover, we consider *arbitrary* input distributions. By contrast, the results in [22] hold for inputs $\mathbf{x}$ distributed *uniformly* with coefficients in $\{0,1\}$.

Both extensions are essential for the sample-preserving search-to-decision LWE reduction presented in Section 4.2, which requires the pseudorandomness of the knapsack function over vector groups $G = \mathbb{Z}_q^k$, and for inputs $\mathbf{x}$ following a nonuniform (Gaussian) distribution over a sufficiently large set $\{-s, \ldots, s\}$. We remark that our generalization is nontrivial, as there are choices of the group G and input distribution, for which the bounded knapsack function is presumably one-way, but *not* pseudorandom. (See Lemma 4.1 for an example.) Our main technical result (Theorem 3.1) shows that for any finite abelian group G and input distribution $\mathcal{X}$, the output of the knapsack function is pseudorandom provided the following two conditions hold:

1. $f_{\mathbf{g}}$ is computationally hard to invert with respect to input distribution $\mathcal{X}$, and
2. certain folded versions of $f_{\mathbf{g}}$ (where both the key $\mathbf{g}$ and the output $f_{\mathbf{g}}(\mathbf{x})$ are projected onto a quotient group $G_d = G/dG$ for some $d \in \mathbb{Z}$,) have pseudorandom output.

The second condition above may seem to make the statement in the theorem vacuous, as it asserts the pseudorandomness of $f_{\mathbf{g}}$ assuming the pseudorandomness of (certain other versions of) $f_{\mathbf{g}}$. The power of the theorem comes from the fact that the quotient groups G_d considered are very small. So small that for many interesting groups and input distributions the folded knapsack function $f_{\mathbf{g}}(\mathbf{x}) \bmod dG$ *compresses* the input (rather than stretching it) and produces an output which is *statistically* close to uniform. Therefore, for all such groups and input distributions, the one-wayness of the bounded knapsack function directly implies that knapsacks are good pseudorandom generators. Specific groups and input distributions for which this holds include:

- Groups whose order contains only large prime factors, larger than the maximum value of the input coefficients. Cyclic groups with prime order and vector groups $\mathbb{Z}_p^k$ for prime p fall into this category. This result generalizes those in [22] from uniform binary input to arbitrary input distributions.
- Distributions that, when folded (modulo small divisors of the order of G,) maintain high entropy relative to the size of the quotient group G/dG. (See Theorem 4.3.) Groups of the form $G = \mathbb{Z}_{2^\ell}^k$ and uniform input distribution over $\mathbb{Z}_{2^i}^m$ for some $i < \ell$ satisfy this requirement.

This last parameter set is a very attractive choice in practice since both group operations and input sampling are particularly efficient and easy to implement using arithmetic modulo powers of 2.

1.2 Pseudorandomness of the LWE Function

Our results for LWE are obtained using the duality between LWE and the knapsack function over vector groups[2]. Specifically, the LWE problem with secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ and m samples, can be shown to be essentially equivalent to the knapsack problem over the vector group $\mathbb{Z}_q^{m-n}$ when the input $\mathbf{x} \in \mathbb{Z}_q^m$ follows the same distribution as the LWE error $\mathbf{e}$. This duality is by no means new, and has been noticed and used in different settings [42, 30]. Here we observe that the duality holds both for the search and decision variants of both problems, so that search-to-decision reductions for the knapsack functions can be readily translated into corresponding *sample-preserving* search-to-decision reductions for LWE. As a corollary to our main theorem, we get search-to-decision reductions for the following interesting cases (among others):

- Binary modulus $q = 2$ and *any* error distribution. This directly proves the pseudorandomness of the well-known Learning Parity with Noise (LPN) problem, as already established in [10, 7, 23].
- Prime modulus q and *any polynomially bounded* error distribution.
- Prime power modulus $q = p^e$ for prime $p = \mathsf{poly}(n)$ large enough such that the error distribution is concentrated over $\{-(p-1)/2, \ldots, (p-1)/2\}$.
- Prime power modulus $q = p^e$ for arbitrary (possibly small) prime p and *uniform* error distribution over $\mathbb{Z}_{p^i}$ for some $i < e$ such that $p^i = \mathsf{poly}(n)$.

These results subsume (see below) several previous pseudorandomness results for LWE [38, 6] and LPN [23] but with an important difference. While the proofs in [38, 6, 23] require that LWE (resp. LPN) is hard to solve (invert) for a very large number of samples, our reductions are *sample preserving*: the pseudorandomness of LWE (resp. LPN) holds, provided the same problem is computationally hard to solve in its search version with the *same* number of samples[3]. We remark that previous results are often phrased as reductions from solving the LWE search problem *with high probability*, to solving the LWE decision problem *with*

[2] We stretch that search LWE can be *directly* reduced (i.e. without exploiting the duality with the knapsack function) to its decision variant. Nevertheless, we still choose to use the aforementioned duality for the following reasons: i) our results for knapsack functions are much more general. Indeed, the LWE function family can be seen as the dual of the particular knapsack family where the underlying group is the vector group $\mathbb{Z}_q^k$. ii) in the general case where $\mathbb{Z}_q$ has *composite* order, i.e. q is composite, proving a direct search-to-decision reduction is no less technical than proving an indirect one (using the duality with knapsack families). iii) for some parameters, the direct reduction, unlike the indirect one, *does not* preserve the number of samples. One such example is when q is *superpolynomial* but the noise is *polynomially bounded*.

[3] For LPN, a sample-preserving reduction was proved in [7].

nonnegligible advantage, combining the search-to-decision reduction and success probability amplification into a single statement. By contrast, our reduction shows how to solve the LWE search problem with *nonnegligible probability*. Our results subsume previous work in the sense that the LWE search problem can be solved with high probability by first invoking our reduction, and then amplifying the success probability using standard repetition techniques. Of course, any such success probability amplification would naturally carry the cost of a higher sample complexity. We remark that a close inspection of worst-case to average-case reductions for LWE [38, 35] shows that these reductions directly support the conjecture that LWE is a *strong* one-way function, i.e., a function which is hard to invert even with just nonnegligible probability. As already discussed, worst-case to average-case reductions do not provide quantitatively interesting results, and are best used as qualitative arguments to support the conjecture that certain problems are computationally hard on average. Under the standard conjecture that search LWE is a *strong one-way function*, the results in this paper offer a fairly tight, and sample preserving proof that LWE is also a good *pseudorandom generator*, which can be efficiently used for the construction of many other lattice based cryptographic primitives. By contrast, it is not known how to take advantage of the strong one-wayness of LWE within previous search-to-decision reductions, resulting in a major degradation of the parameters. Of course, if we change the complexity assumption, and as a starting point we use the *worst-case* hardness of lattice problems or the assumption that LWE is only a *weak* one-way function, then our reduction will also necessarily incur a large blow up in sample complexity through amplification, and lead to quantitatively uninteresting results.

2 Preliminaries

We use $\mathbb{Z}, \mathbb{N}, \mathbb{C}$ for the sets of integer, natural and complex numbers respectively, and $\mathbb{T}$ for the set of complex numbers of unit magnitude. We use lower case for scalars, upper case for sets, bold lower case for vectors and bold upper case for matrices. We also use calligraphic letters for probability distributions and (possibly randomized) algorithms. For any $s \in \mathbb{N}$, the set of the first s nonnegative integers is denoted $[s] = \{0, 1, \ldots, s-1\}$.

2.1 Probability

We write $x \leftarrow \mathcal{X}$ for the operation of selecting x according to a probability distribution $\mathcal{X}$ or by running probabilistic algorithm $\mathcal{X}$. We use set comprehension notation to describe sets and probability distributions alike. E.g., $\{(x, x') \mid x \leftarrow \mathcal{X}, x' \leftarrow \mathcal{X}\}$ denotes the probability distribution obtained by drawing two samples from $\mathcal{X}$ independently at random. For any probability distribution $\mathcal{X}$ over set X and any value $x \in X$, let $\Pr\{x \leftarrow \mathcal{X}\}$ be the probability associated to x by distribution $\mathcal{X}$. The uniform distribution over a set A is denoted $\mathcal{U}(A)$, and the support of a distribution $\mathcal{X}$ is denoted $[\mathcal{X}] = \{x \in X \mid \Pr\{x \leftarrow \mathcal{X}\} > 0\}$. The *collision probability* of $\mathcal{X}$ is the probability $\mathsf{Col}(\mathcal{X}) = \Pr\{x = x' \mid x \leftarrow \mathcal{X}, x' \leftarrow \mathcal{X}\} = \sum_{x \in [\mathcal{X}]} \Pr\{x \leftarrow \mathcal{X}\}^2$ that two independent identically distributed samples from $\mathcal{X}$ take the same value. The *mode* of $\mathcal{X}$ is the probability of the most likely value, i.e., $\mathsf{mode}(\mathcal{X}) = \max_{x \in X} \Pr\{x \leftarrow \mathcal{X}\}$. It is easy to see that $\mathsf{Col}(\mathcal{X}) \leq \mathsf{mode}(\mathcal{X})$.

Whenever we compare two probability distributions, we implicitly assume that they are defined over the same set. The *statistical distance* between distributions $\mathcal{X}$ and $\mathcal{Y}$ over the set X is the quantity $\Delta(\mathcal{X}, \mathcal{Y}) = \frac{1}{2} \sum_{x \in X} |\Pr\{x \leftarrow \mathcal{X}\} - \Pr\{x \leftarrow \mathcal{Y}\}|$. The statistical distance is a metric over the set of discrete probability distributions, i.e., it is a symmetric positive function, and it satisfies the triangle inequality. It also satisfies $\Delta(f(\mathcal{X}), f(\mathcal{Y})) \leq \Delta(\mathcal{X}, \mathcal{Y})$ for any (possibly probabilistic) function f. Two distributions $\mathcal{X}, \mathcal{Y}$ are *ϵ-close* if $\Delta(\mathcal{X}, \mathcal{Y}) < \epsilon$. They are *$(t, \epsilon)$-indistinguishable* if $\Delta(\mathcal{D}(\mathcal{X}), \mathcal{D}(\mathcal{Y})) < \epsilon$ for any probabilistic predicate $\mathcal{D}\colon X \to \{0, 1\}$ (called the *distinguisher*) computable in time at most t. Otherwise, we say that $\mathcal{X}, \mathcal{Y}$ are *(t, ϵ)-distinguishable*. When $\mathcal{Y} = \mathcal{U}(X)$ is the uniform distribution, we use $\Delta_U(\mathcal{X}) = \Delta(\mathcal{X}, \mathcal{U}(X))$ as an abbreviation and say that $\mathcal{X}$ is *ϵ-random* (resp. *(t, ϵ)-pseudorandom*) if it is ϵ-close to (resp. (t, ϵ)-indistinguishable from) $\mathcal{U}(X)$.

Function families. A function family $(F, \mathcal{X})$ is a collection $F = \{f_i\colon X \to R\}_{i \in I}$ of functions indexed by $i \in I$ with common domain X and range R, together with a probability distribution $\mathcal{X}$ over the input

set $X \supseteq [\mathcal{X}]$. For simplicity, in this paper we always assume that the set of functions is endowed with the *uniform* probability distribution $\mathcal{U}(F)$, though the extension to general distributions (while not useful in this paper) is rather straightforward. Each function family $(F, \mathcal{X})$ naturally defines a probability distribution

$$\mathcal{F}(F, \mathcal{X}) = \{(f, f(x)) \mid f \leftarrow \mathcal{U}(F), x \leftarrow \mathcal{X}\} \tag{1}$$

obtained by selecting a function at random and evaluating it at a random input.

A function family $\mathcal{F} = (F, \mathcal{X})$ is called (t, ϵ)-*one-way* if for any (probabilistic) algorithm $\mathcal{I}$ running in time at most t, it holds $\Pr\{f(x) = y \mid (f, y) \leftarrow \mathcal{F}(F, \mathcal{X}), x \leftarrow \mathcal{I}(f, y)\} < \epsilon$. $\mathcal{F}$ is (t, ϵ)-*invertible* if there exists a (probabilistic) algorithm $\mathcal{I}$ running in time at most t such that $\Pr\{f(x) = y \mid (f, y) \leftarrow \mathcal{F}(F, \mathcal{X}), x \leftarrow \mathcal{I}(f, y)\} \geq \epsilon$. We then say that $\mathcal{I}$ is a (t, ϵ)-*inverter* for $\mathcal{F}$. A (t, ϵ)-*pseudorandom generator family*[4] is a function family $(F, \mathcal{X})$ such that the associated distribution $\mathcal{F}(F, \mathcal{X})$ defined in (1) is (t, ϵ)-pseudorandom.

Asymptotics. We use n as a (security) parameter that controls all other parameters. Unless otherwise stated, any other parameter (say m) will be polynomially related to n, that is $1/n^{c_1} \leq m \leq n^{c_2}$ for some constants c_1, c_2. We use standard asymptotic notation $O(\cdot), \Omega(\cdot), o(\cdot), \omega(\cdot)$, etc. We write $\mathsf{negl}(n) = n^{-\omega(1)}$ for the set of negligible functions and $\mathsf{poly}(n) = n^{O(1)}$ for the set of polynomially bounded functions. In the asymptotic computational complexity setting, one often considers probability ensembles, i.e., sequences $\mathcal{X} = (\mathcal{X}_n)_{n \in \mathbb{N}}$ of probability distributions over possibly different sets $X_n \supseteq [\mathcal{X}_n]$. Two distributions ensembles $\mathcal{X} = (\mathcal{X}_n)_{n \in \mathbb{N}}$ and $\mathcal{Y} = (\mathcal{Y}_n)_{n \in \mathbb{N}}$ are *statistically close* (denoted $\mathcal{X} \simeq \mathcal{Y}$) if $\mathcal{X}_n$ and $\mathcal{Y}_n$ are $\epsilon(n)$-close for some negligible function $\epsilon(n) = \mathsf{negl}(n)$. The ensembles $\mathcal{X}$ and $\mathcal{Y}$ are *computationally indistinguishable* (denoted $\mathcal{X} \approx \mathcal{Y}$) if $\mathcal{X}_n$ and $\mathcal{Y}_n$ are $(t(n), \epsilon(n))$-indistinguishable for any $t(n) = \mathsf{poly}(n)$ and some $\epsilon(n) = \mathsf{negl}(n)$ under a sequence $(\mathcal{D}_n \colon X_n \to \{0,1\})_{n \in \mathbb{N}}$ of distinguishers computable in uniform polynomial time. Definitions for function families are also extended in the obvious way to function family ensembles $\mathcal{F} = (\mathcal{F}_n)_n$ in the asymptotic setting by taking $\epsilon(n) = \mathsf{negl}(n)$ and $t(n) = \mathsf{poly}(n)$, and considering uniform sequences of distinguishing algorithms. In particular, a function family ensemble $\mathcal{F} = (\mathcal{F}_n)_n$ is *one-way* if $\mathcal{F}_n$ is $(t(n), \epsilon(n))$-one-way for any $t(n) = \mathsf{poly}(n)$ and some $\epsilon(n) = \mathsf{negl}(n)$. It is *pseudorandom* if the associated distribution ensemble (1) is $(t(n), \epsilon(n))$-pseudorandom, i.e., it is $(t(n), \epsilon(n))$-indistinguishable from the uniform distribution $\mathcal{U}(F_n \times R_n)$ for any $t(n) = \mathsf{poly}(n)$ and some $\epsilon(n) = \mathsf{negl}(n)$.

2.2 Groups and Knapsack Function Families.

In this work, by group we always mean *finite abelian group*. We use additive notation for groups; 0_G is the *neutral element*, $|G|$ is the *order* (size) of G and M_G is its *exponent*, i.e. the smallest positive integer e such that $e \cdot g = 0_G$ for all $g \in G$. We use the dot product notation $\mathbf{x} \cdot \mathbf{y} = \sum_i x_i \cdot y_i$ both for the inner product of two vectors $\mathbf{x}, \mathbf{y} \in R^n$ with elements in a ring R, and also to take integer linear combinations $\mathbf{x} \in \mathbb{Z}^n$ of a vector $\mathbf{y} \in G^n$ with elements in an additive group. For $\mathbf{x} = (x_1, \ldots, x_n) \in R^n$ and $a \in R$, we also define $a \cdot \mathbf{x} = \mathbf{x} \cdot a = (x_1 \cdot a, \ldots, x_n \cdot a)$.

For any group G and (positive) integer d, we use G_d to denote the quotient group G/dG where dG is the subgroup $\{d \cdot g \mid g \in G\}$, in analogy with the usual notation $\mathbb{Z}_d = \mathbb{Z}/d\mathbb{Z}$ for the group of integers modulo d. Likewise, for an element $g \in G$, we use $g \bmod dG$ (or just $g \bmod d$) for the image of g under the natural homomorphism from G to G_d. For any integer vector $\mathbf{w} = (w_1, \ldots, w_r) \in \mathbb{Z}^r$, we write $\gcd_G(\mathbf{w}) = \gcd(w_1, \ldots, w_r, M_G)$ for the greatest common divisor of the elements of $\mathbf{w}$ and the group exponent. We recall that any finite abelian group G is isomorphic to $\mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$ where $k_i | k_{i+1}$ for all i, and $k_\ell = M_G$.

[4] Notice that the functions in a pseudorandom generator family are *not* pseudorandom functions, as they do not accept any input beside the (randomly generated) seed $x \leftarrow \mathcal{X}$. Each function $f \in F$ works like a pseudorandom generator that on input a random seed $x \leftarrow \mathcal{X}$, produces an output $f(x)$ which is indistinguishable from a random element of the range R. Throughout the paper, by pseudorandom family, we will always mean a pseudorandom generator family. We also remark that in this paper the term "pseudorandom generator" is used in a loose sense, as we do not require f to "stretch" the seed x into a longer string or generate any pseudo-entropy. The function f may even compress the seed into a shorter string, and produce a distribution $f(x)$ which is *statistically* close to uniform over the range of f.

If $G \simeq \mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$ then $G_d \simeq \mathbb{Z}_{d_1} \times \mathbb{Z}_{d_2} \times \cdots \times \mathbb{Z}_{d_\ell}$ where $d_i = \gcd(d, k_i)$ for $i = 1, \ldots, \ell$. In particular

$$|G_d| = \Pi_{i=1}^{\ell} d_i \qquad \text{and} \qquad |dG| = \frac{|G|}{|G_d|}. \tag{2}$$

Lemma 2.1. *For any group G, integer vector $\mathbf{w} \in \mathbb{Z}^r$, and $d = \gcd_G(\mathbf{w})$, we have $\{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \leftarrow \mathcal{U}(G^r)\} = \mathcal{U}(d \cdot G)$. In particular, $\Pr\{\mathbf{w} \cdot \mathbf{g} = 0_G \mid \mathbf{g} \leftarrow \mathcal{U}(G^r)\} = 1/|d \cdot G| = \prod_i \gcd(d, k_i)/k_i$.*

Proof. Fix $\mathbf{w} \in \mathbb{Z}^r$, and let $d = \gcd_G(\mathbf{w})$. We want to analyze the probability distribution $\mathcal{W} = \{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \leftarrow \mathcal{U}(G^r)\}$. The function $\phi\colon \mathbf{g} \mapsto \mathbf{w} \cdot \mathbf{g}$ maps G^r to $[\mathcal{W}] = \{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \in G^r\} = d \cdot G$. Let $G_0 = \{\mathbf{g} \mid \mathbf{w} \cdot \mathbf{g} = 0\}$ be the kernel of this function. Then, ϕ partitions G^r into equivalence classes of the form $\mathbf{g} + G_0$. All the equivalence classes have the same size $|\mathbf{g} + G_0| = |G_0|$, and therefore ϕ maps the uniform distribution over G^r to the uniform distribution over $\phi(G^r) = d \cdot G$. This proves that $\mathcal{W} = \mathcal{U}(d \cdot G)$. The bound on $\Pr\{\mathbf{w} \cdot \mathbf{g} = 0_G\}$ follows from (2). □

Knapsack Families. For any group G and input distribution $\mathcal{X}$ over $\mathbb{Z}^m$, the knapsack family $\mathcal{K}(G, \mathcal{X})$ is the function family with input distribution $\mathcal{X}$ and set of functions $f_{\mathbf{g}}\colon [\mathcal{X}] \to G$ indexed by $\mathbf{g} \in G^m$ and defined as $f_{\mathbf{g}}(\mathbf{x}) = \mathbf{g} \cdot \mathbf{x} \in G$. Typically, the input distribution $\mathcal{X} = \mathcal{S}^m$ is given by m independent identically distributed samples $(x_1, \ldots, x_m)$, chosen from some probability distribution $\mathcal{S}$ over a finite (and polynomially sized) subset of the integers $[\mathcal{S}] \subset \mathbb{Z}$. We will often use $\mathbf{g}$ instead of $f_{\mathbf{g}}$ to describe a member function drawn from $\mathcal{K}(G, \mathcal{X})$. When $G, \mathcal{X}$ are clear from the context we will simply write $\mathcal{K}$. We often consider folded knapsack families $\mathcal{K}(G_d, \mathcal{X})$ over quotient groups G_d. For brevity, when G and $\mathcal{X}$ are clear from the context, we will write $\mathcal{K}_d$ instead of $\mathcal{K}(G_d, \mathcal{X})$. The following lemma shows that the distribution $\mathcal{F}(\mathcal{K}_d)$ associated to a folded knapsack function family is closely related to the distribution

$$\mathcal{F}_d(\mathcal{K}) = \{(\mathbf{g}, g + h) \mid (\mathbf{g}, g) \leftarrow \mathcal{F}(\mathcal{K}), h \leftarrow \mathcal{U}(d \cdot G)\}. \tag{3}$$

Lemma 2.2. *For any knapsack function family $\mathcal{K}$ and integer d, $\Delta_U(\mathcal{F}_d(\mathcal{K})) = \Delta_U(\mathcal{F}(\mathcal{K}_d))$. Moreover, $\mathcal{F}_d(\mathcal{K})$ is pseudorandom if and only if $\mathcal{F}(\mathcal{K}_d)$ is pseudorandom.*

Proof. The lemma follows from the existence of two efficiently computable (randomized) transformations m, m' that appropriately map distributions over $G^m \times G$ to distributions over $G_d^m \times G_d$ and vice versa.

- Let $m : G^m \times G \to G_d^m \times G_d$ be the function $m(\mathbf{g}, g) = (\mathbf{g} \bmod d, g \bmod d)$. It is straightforward to verify that m maps $\mathcal{U}(G^m \times G)$ to $\mathcal{U}(G_d^m \times G_d)$ and $\mathcal{F}_d(\mathcal{K})$ to $\mathcal{F}(\mathcal{K}_d)$.
- In the other direction, let $m'\colon G_d^m \times G_d \to G^m \times G$ be the randomized transformation that on input $(\mathbf{h}, h)$ produces an output distributed according to $\{(\mathbf{h} + d \cdot \mathbf{g}, h + d \cdot g) \mid (\mathbf{g}, g) \leftarrow \mathcal{U}(G^{m+1})\}$. Again, it is easy to see that m' maps $\mathcal{U}(G_d^m \times G_d)$ to $\mathcal{U}(G^m \times G)$ and $\mathcal{F}(\mathcal{K}_d)$ to $\mathcal{F}_d(\mathcal{K})$.

It follows that $\Delta_U(\mathcal{F}(\mathcal{K}_d)) = \Delta(m(\mathcal{F}_d(\mathcal{K})), m(\mathcal{U}(G^m \times G))) \le \Delta_U(\mathcal{F}_d(\mathcal{K}))$ and similarly $\Delta_U(\mathcal{F}_d(\mathcal{K})) = \Delta(m'(\mathcal{F}_d(\mathcal{K})), m'(\mathcal{U}(G^m \times G)) \le \Delta(\mathcal{F}(\mathcal{K}_d))$. This proves $\Delta_U(\mathcal{F}_d(\mathcal{K})) = \Delta_U(\mathcal{F}(\mathcal{K}_d))$. Since the transformations m and m' are efficiently computable, they can also be used to turn any efficient distinguisher for $\mathcal{F}_d(\mathcal{K})$ into an efficient distinguisher for $\mathcal{F}(\mathcal{K}_d)$, and vice versa. □

We will need the following variant of the Leftover Hash Lemma [21], generalized to arbitrary abelian groups. The original Leftover Hash Lemma [21], applies to any universal (or ϵ-universal) hash function family over arbitrary sets. Our version of the lemma is specific to knapsack functions, but relaxes the universality requirement.

Lemma 2.3 (Leftover Hash Lemma, generalized)**.** *For any knapsack function family $\mathcal{K} = \mathcal{K}(H, \mathcal{X})$ over a finite abelian group H,*

$$\Delta_U(\mathcal{F}(\mathcal{K})) \le \frac{1}{2}\sqrt{\sum_{1<d \mid M_H} |H_d| \cdot \mathsf{Col}(\mathcal{X}_d)} \tag{4}$$

where $\mathcal{X}_d = \mathcal{X} \bmod d = \{x \bmod d \mid x \leftarrow \mathcal{X}\}$, and d ranges over all divisors of the group exponent M_H strictly greater than 1 (M_H included).

Proof. Let $\mathcal{Z}$ be any distribution over a set Z. The following standard computation provides an upper bound on the statistical distance between $\mathcal{Z}$ and $\mathcal{U}(Z)$ in terms of the collision probability $\mathsf{Col}(\mathcal{Z})$.

$$\begin{aligned}\Delta_U(\mathcal{Z}) &= \frac{1}{2}\sum_{z\in Z}\left|\Pr\{z\leftarrow\mathcal{Z}\}-\frac{1}{|Z|}\right| \le \frac{1}{2}\sqrt{|Z|}\sqrt{\sum_{z\in Z}\left(\Pr\{z\leftarrow\mathcal{Z}\}-\frac{1}{|Z|}\right)^2}\\ &= \frac{1}{2}\sqrt{|Z|}\sqrt{\sum_{z\in Z}\Pr\{z\leftarrow\mathcal{Z}\}^2-\frac{2}{|Z|}+\frac{1}{|Z|}}\\ &\le \frac{1}{2}\sqrt{|Z|\cdot\mathsf{Col}(\mathcal{Z})-1}. \end{aligned} \tag{5}$$

We bound $\mathsf{Col}(\mathcal{F}(\mathcal{K}))$ as follows, where all probabilities are computed over the random choice of $\mathbf{h},\mathbf{h}'\leftarrow\mathcal{K}$ and $\mathbf{x},\mathbf{y}\leftarrow\mathcal{X}$:

$$\begin{aligned}\mathsf{Col}(\mathcal{F}(\mathcal{K})) &= \Pr\{(\mathbf{h}=\mathbf{h}')\wedge(\mathbf{h}\cdot\mathbf{x}=\mathbf{h}'\cdot\mathbf{y})\}\\ &= \Pr\{(\mathbf{h}=\mathbf{h}')\wedge(\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0)\}\\ &= \frac{1}{|H|^m}\cdot\Pr\{\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0\}. \end{aligned} \tag{6}$$

It remains to compute $\Pr\{\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0\}$. For that, we condition on the value of $d=\gcd_H(\mathbf{x}-\mathbf{y})$ and use Lemma 2.1. Notice that since $\gcd_H(\mathbf{x}-\mathbf{y})$ divides M_H, we can restrict d to the divisors of M_H.

$$\begin{aligned}\Pr\{\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0\} &= \sum_{d|M_H}\Pr\{\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0 \mid \gcd_H(\mathbf{x}-\mathbf{y})=d\}\cdot\Pr\{\gcd_H(\mathbf{x}-\mathbf{y})=d\}\\ &\le \sum_{d|M_H}\frac{1}{|dH|}\cdot\mathsf{Col}(\mathcal{X}_d)\\ &= \frac{1}{|H|}+\sum_{1<d|M_H}\frac{1}{|dH|}\cdot\mathsf{Col}(\mathcal{X}_d). \end{aligned} \tag{7}$$

where we used that $\Pr\{\gcd_H(\mathbf{x}-\mathbf{y})=d\}\le\Pr\{d\mid\mathbf{x}-\mathbf{y}\}=\Pr\{\mathbf{x} \bmod d=\mathbf{y} \bmod d\}=\mathsf{Col}(\mathcal{X}_d)$ in the inequality above. Combining (5), (6) and (7), and using $|H_d|\cdot|dH|=|H|$, yields the bound in the lemma. □

2.3 Lattices and Gaussian Distributions

Gaussian-like distributions play a central role in the Learning With Errors (LWE) problem. For each sample $(\mathbf{a},b=\mathbf{a}\cdot\mathbf{s}+e)$, the distribution χ from which e is drawn, is usually a Gaussian-like distribution over the integers. Several (worst-case) lattice approximation problems can be reduced (under quantum or classic polynomial time reductions [38, 35]) to LWE with Gaussian error distribution. Moreover, Gaussian noise is "LWE-complete" [39, 17] in the sense that LWE with non-Gaussian error distribution can be reduced to LWE where the error is distributed according to a wider Gaussian. Below, we focus on the *discrete* Gaussian distribution, i.e., the conditional distribution obtained restricting a normal real random variable to take integer values. We provide bounds on the collision probability of the discrete Gaussian distribution, and use the bounds to establish search-to-decision reductions for LWE. Similar results hold also for the *discretized* Gaussian distribution, i.e., the distribution obtained by rounding the output of a real Gaussian random variable to the closest integer. Statements and proofs for discretized Gaussians are virtually identical and hence omitted.

Discrete Gaussian. The *Gaussian function* $\rho_{r,\mathbf{c}}\colon\mathbb{R}^m\to\mathbb{R}$ with center $\mathbf{c}$ and width r is defined as

$$\rho_{r,\mathbf{c}}(\mathbf{x})=e^{-\frac{\pi\|\mathbf{x}-\mathbf{c}\|^2}{r^2}}.$$

The *discrete Gaussian* with parameters $r, \mathbf{c}$ over a countable set $S \subset \mathbb{R}^m$ is the distribution $\mathcal{D}_{S,r,\mathbf{c}}$ that samples each element $\mathbf{x} \in S$ with probability

$$\Pr\{\mathbf{x} \leftarrow \mathcal{D}_{S,r,\mathbf{c}}\} = \frac{\rho_{r,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{y}\in S} \rho_{r,\mathbf{c}}(\mathbf{y})}.$$

When the center $\mathbf{c}$ is omitted from the notation $\mathcal{D}_{S,r}$ it is assumed to be the origin $\mathbf{c} = \mathbf{0}$. We will be primarily interested in discrete Gaussians over the set of integer vectors $S = \mathbb{Z}^m$. For such set, the vectors $\mathbf{x} \in \mathbb{Z}^m$ sampled by $\mathcal{D}_{\mathbb{Z}^m,r}$ have each coordinate x_i identically and independently distributed according to a 1-dimensional Gaussian, i.e.,

$$\Pr\{\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,r}\} = \prod_{i=1}^{m} \Pr\{x_i \leftarrow \mathcal{D}_{\mathbb{Z},r}\} = \prod_{i=1}^{m} \frac{\rho_r(x_i)}{\rho_r(\mathbb{Z})}. \tag{8}$$

Lattices. A (full-rank) m-dimensional *lattice* is the set Λ of integer combinations of m *linearly independent* vectors $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^m$, i.e.

$$\Lambda = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{Z} \text{ for } i = 1, \ldots, m \right\}.$$

The matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$ is called a *basis* for the lattice Λ. The *determinant* of a lattice Λ (denoted $det(\Lambda)$) is the absolute value of the matrix determinant of any basis $\mathbf{B}$ of Λ, i.e. $det(\Lambda) = |det(\mathbf{B})|$. The i-th *successive minimum* $\lambda_i(\Lambda)$ is the radius r of the smallest m-dimensional (Euclidean) ball that contains i linearly independent vectors from Λ. The *dual* of a lattice Λ is the set

$$\Lambda^* = \{\mathbf{x} \in \mathbb{R}^m : \forall \mathbf{y} \in \Lambda,\ \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}.$$

The *integer lattice* $\mathbb{Z}^m$ contains all m-dimensional vectors with integer coefficients. It is easy to check that for any m, $\mathbb{Z}^m$ is a *self-dual* lattice, that is $(\mathbb{Z}^m)^* = \mathbb{Z}^m$. Also $det(\mathbb{Z}^m) = 1$ and $\lambda_i(\mathbb{Z}^m) = 1$ for all $i = 1, \ldots, m$.

The *Poisson Summation Formula* implies that for any lattice Λ and real $r > 0$,

$$\rho_r(\Lambda) = r \cdot det(\Lambda^*) \rho_{1/r}(\Lambda^*). \tag{9}$$

For any $\epsilon \in \mathbb{R}^+$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ [32] is the smallest $r > 0$ such that $\rho_{1/r}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$. We will use the following bounds.

Proposition 2.4 ([32, Lemma 3.3 and Lemma 4.4])**.** *Let Λ be an m-dimensional lattice.*

1. *For any function $f(n) = \omega(\sqrt{\log n})$, there exists $\epsilon(n) = \mathsf{negl}(n)$ such that $\eta_\epsilon(\Lambda) \leq f(n) \cdot \lambda_m(\Lambda)$.*
2. *For any $\epsilon \in (0,1), r \geq \eta_\epsilon(\Lambda)$ and $\mathbf{c} \in \mathbb{R}^m$, we have that $\rho_{r,\mathbf{c}}(\Lambda) \in \left(\frac{1-\epsilon}{1+\epsilon}, 1\right) \cdot \rho_r(\Lambda)$.*

For our search-to-decision reduction of LWE with discrete Gaussian error distribution, we also need to consider the *folded* (1-dimensional) distribution $\mathcal{D}_{\mathbb{Z},r} \bmod d = \{x \bmod d \mid x \leftarrow \mathcal{D}_{\mathbb{Z},r}\}$. The following lemma gives an upper bound on the collision probability of this distribution.

Lemma 2.5. *For any $r > 0$ and $d \in \mathbb{Z}$, we have $\mathsf{Col}(\mathcal{D}_{\mathbb{Z},r} \bmod d) \leq \frac{1}{r} + \frac{1}{d}$. Furthermore, if $r = d \cdot \omega(\sqrt{\log n})$, then $\mathsf{Col}(\mathcal{D}_{\mathbb{Z},r} \bmod d) \leq \frac{1}{d} + \mathsf{negl}(n)$.*

Proof. We first bound the mode of $\mathcal{D}_{\mathbb{Z},r} \bmod d$.

$$\mathsf{mode}(\mathcal{D}_{\mathbb{Z},r} \bmod d) = \max_{0 \leq j \leq d-1} \Pr\{\mathcal{D}_{\mathbb{Z},r} \bmod d = j\} = \max_j \frac{\rho_r(d\mathbb{Z} + j)}{\rho_r(\mathbb{Z})} = \frac{\rho_r(d\mathbb{Z})}{\rho_r(\mathbb{Z})}.$$

Using Poisson summation formula (9), we get $\rho_r(\mathbb{Z}) = r \cdot \rho_{1/r}(\mathbb{Z}) \geq r$. For the numerator, we have

$$\rho_r(d\mathbb{Z}) = \rho_{r/d}(\mathbb{Z}) = 1 + 2\sum_{i\geq 1} \rho_{r/d}(i) \leq 1 + \int_{-\infty}^{\infty} \rho_{r/d}(x)\, dx = 1 + r/d.$$

We conclude that $\mathsf{Col}\,(\mathcal{D}_{\mathbb{Z},r} \bmod d) \leq \mathsf{mode}\,(\mathcal{D}_{\mathbb{Z},r} \bmod d) \leq 1/r + 1/d$.

When $r = d \cdot \omega(\sqrt{\log n})$, a better bound on $\mathsf{Col}\,(\mathcal{D}_{\mathbb{Z},r} \bmod d)$ is given by

$$\begin{aligned}
\Pr\{\mathcal{D}_{\mathbb{Z},r} \bmod d = 0\} - \Pr\{\mathcal{D}_{\mathbb{Z},r} \bmod d = j\} &= \frac{\rho_r(d\cdot\mathbb{Z}) - \rho_r(d\cdot\mathbb{Z}+j)}{\rho_r(\mathbb{Z})} \\
&= \frac{\rho_{r/d}(\mathbb{Z}) - \rho_{r/d,-j/d}(\mathbb{Z})}{\rho_r(\mathbb{Z})} \\
&\leq \frac{\rho_{r/d}(\mathbb{Z})}{\rho_r(\mathbb{Z})}\left(1 - \frac{1-\epsilon}{1+\epsilon}\right) \\
&\leq \mathsf{negl}(n)
\end{aligned}$$

where the inequalities follow from Proposition 2.4 and the fact that $r/d = \omega(\sqrt{\log n})$. This bound shows that $\mathcal{D}_{\mathbb{Z},r} \bmod d$ is statistically close to the uniform distribution over $\mathbb{Z}_d$ and therefore $\mathsf{Col}\,(\mathcal{D}_{\mathbb{Z},r} \bmod d) \leq \mathsf{Col}\,(\mathcal{U}(\mathbb{Z}_d)) + \mathsf{negl}(n) = \frac{1}{d} + \mathsf{negl}(n)$. □

2.4 Fourier Analysis and Learning

We have already mentioned that Fourier analysis and Gaussian distributions play an important role in basing the average-case hardness of LWE on worst-case lattice assumptions [38, 35]. In this paper, we also use Fourier analysis, but in a quite different way, closer to the use made of Fourier analysis in learning theory and in the complexity study of boolean functions. (E.g., see [25, 9, 34].) In cryptography, two noteworthy examples that make a similar use of Fourier analysis are the Kushilevitz-Mansour [25] formulation of the proof of the Goldreich-Levin [18] hard-core predicate for any one-way function, and the proofs of hard-core predicates for several number-theoretic one-way functions by Akavia, Goldwasser and Safra [4].

Below we review some basic facts from Fourier analysis focusing on the discrete Fourier transform over finite abelian groups. We restrict the presentation to what is needed and refer the interested reader to [3, 41] for more details.

Fourier Basics. Let H be a finite abelian group and $h_1, h_2 : H \to \mathbb{C}$ be functions from H to the complex numbers. The *inner product* of h_1 and h_2 is defined as

$$\langle h_1, h_2 \rangle = \mathop{\mathbb{E}}_{x \leftarrow \mathcal{U}(H)} \left[\, h_1(x)\overline{h_2(x)} \,\right] = \frac{1}{|H|}\sum_{x\in H} h_1(x)\overline{h_2(x)}$$

where $\bar{z}$ is the complex conjugate of $z \in \mathbb{C}$. The ℓ_2-norm[5] and ℓ_∞-norm of h are defined as

$$\|h\|_2 = \sqrt{\langle h, h\rangle} \quad \text{and} \quad \|h\|_\infty = \max_{x\in H} |h(x)|.$$

The set of *characters* of H (denoted $\mathrm{char}(H)$) is the set of all the *homomorphisms* from H to the complex numbers of unit magnitude $\mathbb{T}$,

$$\mathrm{char}(H) = \{\chi \colon H \to \mathbb{T} \;\big|\; \forall x, y \in H,\, \chi(x+y) = \chi(x)\cdot\chi(y)\}.$$

The set $\mathrm{char}(H)$ with point-wise addition forms a group which is isomorphic to H. If $H = \mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$ and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_\ell) \in H$, then the character $\chi_{\boldsymbol{\alpha}} \colon H \to \mathbb{T}$ associated to $\boldsymbol{\alpha}$ is defined as

$$\chi_{\boldsymbol{\alpha}}(\mathbf{x}) = \omega_{k_1}^{\alpha_1 x_1} \ldots \omega_{k_\ell}^{\alpha_\ell x_\ell}$$

[5]Notice that the definition of $\|h\|_2$ differs from the standard definition of the euclidean norm of a vector by a $\sqrt{|H|}$ normalization factor.

where $\omega_{k_i} = e^{i\frac{2\pi}{k_i}}$ is the k_i-th primitive root of unity. We will be particularly interested in functions defined over *vector groups* $H = \mathbb{Z}_k^\ell$, in which case $\chi_{\boldsymbol{\alpha}}(\mathbf{x}) = (\omega_k)^{\sum_{i=1}^{\ell} \alpha_i x_i} = \omega_k^{\mathbf{x}\cdot\boldsymbol{\alpha}}$.

FOURIER TRANSFORM. The *Fourier transform* of a function $h\colon H \to \mathbb{C}$ is the function $\widehat{h}\colon H \to \mathbb{C}$ defined as $\widehat{h}(\boldsymbol{\alpha}) = \langle h, \chi_{\boldsymbol{\alpha}} \rangle$. The Fourier transform measures the correlation of h with the characters of H. The *energy* of a Fourier coefficient $\boldsymbol{\alpha}$ is defined as its squared norm $|\widehat{h}(\boldsymbol{\alpha})|^2$, while the *total energy* of h is defined as $\sum_{\boldsymbol{\alpha}\in H} |\widehat{h}(\boldsymbol{\alpha})|^2$. Parseval's identity asserts that $\sum_{\boldsymbol{\alpha}\in H} |\widehat{h}(\boldsymbol{\alpha})|^2 = \|h\|_2^2$.

Learning Heavy Fourier Coefficients Let $\tau \in \mathbb{R}$, $\boldsymbol{\alpha} \in H$ and $h\colon H \to \mathbb{C}$ where H is a finite abelian group. Following the notation and terminology from [3], we say that $\boldsymbol{\alpha}$ is a τ-significant (or τ-heavy) Fourier coefficient of h if $|\widehat{h}(\boldsymbol{\alpha})|^2 \geq \tau$. The set of τ-significant Fourier coefficients of h is $\mathrm{Heavy}_\tau(h) = \{\boldsymbol{\alpha} \in H \mid |\widehat{h}(\boldsymbol{\alpha})|^2 \geq \tau\}$

Theorem 2.6. *(Significant Fourier Transform,[3, Theorem 3.3]) There exists a probabilistic algorithm ($\mathcal{SFT}$) that on input a threshold τ and given query access to a function $h\colon H \to \mathbb{C}$, returns all τ-heavy Fourier coefficients of h in time $\mathsf{poly}(\log|H|, 1/\tau, \|h\|_\infty)$ with probability[6] at least $2/3$.*

For functions with range $\mathbb{T}$ as considered in this work, it is immediate to verify that $\|h\|_2 = \|h\|_\infty = 1$ and therefore (by Parseval's identity) $\sum_{\boldsymbol{\alpha}\in H} |\widehat{h}(\boldsymbol{\alpha})|^2 = 1$. Among these functions, of particular interest are those whose Fourier spectrum contains coefficients with energy which is a noticeable fraction of the total energy of the function, i.e., there exists a character $\boldsymbol{\beta} \in H$ such that $|\widehat{h}(\boldsymbol{\beta})|^2 \geq \frac{1}{poly(\log|H|)}$. In this context, Theorem 2.6 says that $\mathcal{SFT}$, given query access to a function $h : H \to \mathbb{T}$, can find all its $\frac{1}{poly(\log|H|)}$-heavy Fourier coefficients in time polynomial in $\log|H|$.

3 Pseudorandomness of Knapsack Functions

In this section we establish the connection between the search and decision problems associated to families of bounded knapsack functions. The following theorem summarizes our main result.

Theorem 3.1 (Main). *Let $\mathcal{X}$ be a distribution over $[s]^m \subset \mathbb{Z}^m$ for some $s = \mathsf{poly}(n)$, $m = \mathsf{poly}(n)$, and G be a finite abelian group. If $\mathcal{K} = \mathcal{K}(G, \mathcal{X})$ is one-way and $\mathcal{K}_d = \mathcal{K}(G_d, \mathcal{X})$ is pseudorandom for all $d < s$, then $\mathcal{K}$ is pseudorandom.*

Remark 3.2. *Theorem 3.1 as well as all its implications (see Section 4) hold true even if $\mathcal{X}$ is defined over $\{a, \ldots, b\}^m$ for $a, b \in \mathbb{Z}$ (or more generally over $\{a_1, \ldots, b_1\} \times \cdots \times \{a_m, \ldots, b_m\}$) as long as the (maximum) size $s = \max_i\{b_i - a_i + 1\}$ of the intervals is polynomially bounded. Indeed, knapsack instances $(\mathbf{g}, f_{\mathbf{g}}(\mathbf{x}))$ with input $x_i \in \{a_i, a_i+1, \ldots, b_i\}$ can be immediately reduced to instances $(\mathbf{g}, f_{\mathbf{g}}(\mathbf{y})) = (\mathbf{g}, f_{\mathbf{g}}(\mathbf{x}) - f_{\mathbf{g}}(a_1, \ldots, a_m))$ where $y_i = x_i - a_i \in \{0, \ldots, b_i - a_i\}$.*

Also, all statements remain essentially unchanged for distributions $\mathcal{X}$ such that an input $\mathbf{x} \leftarrow \mathcal{X}$ belongs to $[s]^m$ except with negligible probability even if $\mathcal{X}$'s support is possibly larger than $[s]^m$. For ease of exposition, we will omit dealing with these two technicalities in the rest of the paper.

We remark that for knapsack families $\mathcal{K} = \mathcal{K}(G, \mathcal{X})$ that *stretch* their input, Theorem 3.1 is an *if and only if* statement. Indeed, if $\mathcal{K}$ is pseudorandom, then so is $\mathcal{K}_d$ for any d, because there is an efficiently computable regular transformation $(\mathbf{g}, g) \mapsto (\mathbf{g} \bmod d, g \bmod d)$ that maps $\mathcal{F}(\mathcal{K})$ to $\mathcal{F}(\mathcal{K}_d)$. Moreover, if $\mathcal{K}(G, \mathcal{X})$ stretches the input (or, more specifically, if the range $[\mathcal{F}(\mathcal{K})]$ is sparse in $G^m \times G$,) then any inverter with noticeable success probability can be used as a distinguisher for $\mathcal{F}(\mathcal{K})$ in a straightforward way.

Proving the direction as stated in Theorem 3.1 is much more involved, and makes use of the intermediate notion of (un)predictability defined below. Informally, for any $\ell \in \mathbb{N}$, an ℓ-predictor for a function family

[6] The success probability is taken over the internal randomness of the $\mathcal{SFT}$ algorithm only, and can be amplified using standard repetition techniques. However, this is not needed in our context, so for simplicity we fix the success probability to 2/3.

$(F, \mathcal{X})$ with integer inputs $[\mathcal{X}] \subset \mathbb{Z}^m$ is a weak form of inverter algorithm that on input a function $f \in F$, a target value $f(\mathbf{x})$ and a query vector $\mathbf{r} \in \mathbb{Z}_\ell^m$, attempts to recover the value of $\mathbf{x} \cdot \mathbf{r} \pmod{\ell}$, rather than producing the entire input $\mathbf{x}$. Here ℓ is an auxiliary value, unrelated to the parameters of the knapsack function family, that describes the amount of information recovered by the weak inverter. Below we define two notions to measure the quality of a prediction: *accuracy* and *bias*. The first is probably the most natural notion, and directly measures the predictor's success probability. The second is more technical, and it is needed to use Fourier analytic techniques. For the special case of prime ℓ the two notions are closely related, as shown in Lemma 3.5.

Definition 3.3. *For any $\ell \in \mathbb{N}$ and function family $(F, \mathcal{X})$ with domain $[\mathcal{X}] \subseteq \mathbb{Z}^m$, an* ℓ-predictor *for $(F, \mathcal{X})$ is a probabilistic algorithm $\mathcal{P}$ that on input $(f, y, \mathbf{r}) \in F \times R \times \mathbb{Z}_\ell^m$ outputs a value $\mathcal{P}(f, y, \mathbf{r}) \in \mathbb{Z}_\ell$ which is intended to be a guess for $\mathbf{x} \cdot \mathbf{r} \pmod{\ell}$. The error distribution of a predictor $\mathcal{P}$ is defined as*

$$\mathcal{E}_\ell(\mathcal{P}) = \{\mathbf{x} \cdot \mathbf{r} - \mathcal{P}(f, f(\mathbf{x}), \mathbf{r}) \bmod \ell \mid f \leftarrow \mathcal{U}(F), \mathbf{x} \leftarrow \mathcal{X}, \mathbf{r} \leftarrow \mathcal{U}(\mathbb{Z}_\ell^m)\}.$$

An ℓ-predictor $\mathcal{P}$ is (t, ϵ)-accurate *if it runs in time t and $\Pr\{0 \leftarrow \mathcal{E}_\ell(\mathcal{P})\} \geq \frac{1}{\ell} + \epsilon$, i.e., $\mathcal{P}$ outputs the correct inner product $\mathbf{x} \cdot \mathbf{r} \pmod{\ell}$ with a probability which is better (by ϵ) than a random guess. The* bias *of an ℓ-predictor $\mathcal{P}$ is the quantity $\left|\mathbb{E}\left[\, \omega_\ell^{-k} \mid k \leftarrow \mathcal{E}_\ell(\mathcal{P}) \,\right]\right|$. If $\mathcal{P}$ runs in time t and has bias* at least ϵ, *we say that $\mathcal{P}$ is (t, ϵ)-biased. A function family $(F, \mathcal{X})$ that admits a (t, ϵ)-biased ℓ-predictor is called* (t, ϵ, ℓ)-biased.

The proof of Theorem 3.1 proceeds in two steps. In the first step (Lemma 3.4) we show that any predictor for $\mathcal{K}$ can be efficiently transformed into an inverter for $\mathcal{K}$. This step uses Fourier analysis and holds true for *any* (not necessarily knapsack) function family with domain $[\mathcal{X}] \subseteq \mathbb{Z}^m$. In the second step (Proposition 3.9), we prove that if there exists a distinguisher for $\mathcal{K}$, but no distinguisher for $\mathcal{K}_d$ for small d, then there exists a predictor for $\mathcal{K}$. This step is specific to knapsack families and depends on both the underlying group G and the distribution $\mathcal{X}$. The two steps combined yield Theorem 3.1. Sections 3.1 and 3.2 are devoted to each step of the reduction.

3.1 From One-wayness to Unpredictability

Proving that predictability implies invertibility is not specific to knapsack families. Rather, it holds for any function family $(F, \mathcal{X})$ with $[\mathcal{X}] \subseteq \mathbb{Z}^m$. The proof uses the $\mathcal{SFT}$ algorithm from Theorem 2.6 to learn the heavy Fourier coefficients of a function. We remark that the learning takes place over the group $H = \mathbb{Z}_\ell^m$, which is unrelated to the group G of our knapsack function family. Lemma 3.4 provides sufficient conditions under which predictability implies invertibility.

Lemma 3.4. *Let $(F, \mathcal{X})$ be a function family with $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$. If $(F, \mathcal{X})$ is (t, ϵ, ℓ)-biased for some $\ell \geq s$, then $(F, \mathcal{X})$ is $(\mathsf{poly}(n, \log \ell, 1/\epsilon) \cdot t, \frac{\epsilon}{3})$-invertible.*

Proof. Let $\mathcal{P}$ be a (t, ϵ)-biased ℓ-predictor for $\mathcal{F}$. We use $\mathcal{P}$ to define an inverter $\mathcal{I}$ that on input $(f, f(\mathbf{x}))$ tries to recover $\mathbf{x}$ using the $\mathcal{SFT}$ algorithm from Theorem 2.6. In order to run $\mathcal{SFT}$, the inverter $\mathcal{I}$ needs to provide answers to the queries $\mathbf{r} \in \mathbb{Z}_\ell^m$ made by $\mathcal{SFT}$. The queries are answered invoking $\mathcal{P}$ on an appropriate input (to be defined). The goal is to present $\mathcal{SFT}$ with an oracle/function $h : \mathbb{Z}_\ell^m \to \mathbb{C}$ which is *highly correlated* with the character $\chi_{\mathbf{x}}$, so that $\mathcal{SFT}$ will include $\mathbf{x}$ in the list of heavy Fourier coefficients. Details follow.

The inverter $\mathcal{I}$ takes as input a function $f \leftarrow \mathcal{U}(F)$ and a value $y = f(\mathbf{x})$; it then picks a random string *coins* and runs algorithm $\mathcal{SFT}$ (from Theorem 2.6) with $\tau = \frac{\epsilon^2}{4}$. For every query $\mathbf{r} \in \mathbb{Z}_\ell^m$ issued by $\mathcal{SFT}$, $\mathcal{I}$ runs $\mathcal{P}$ on input $(f, f(\mathbf{x}), \mathbf{r}; \mathit{coins})$ and returns $\omega_\ell^{\mathcal{P}(f, f(\mathbf{x}), \mathbf{r}; \mathit{coins})} \in \mathbb{T}$ to $\mathcal{SFT}$, where $\omega_\ell = e^{2\pi i/\ell}$. Notice that the same random string *coins* is used for all queries, so that the queries of $\mathcal{SFT}$ are answered according to a deterministic function

$$h_{f, f(\mathbf{x}), \mathit{coins}}(\mathbf{r}) = \omega_\ell^{\mathcal{P}(f, f(\mathbf{x}), \mathbf{r}; \mathit{coins})}$$

from $\mathbb{Z}_\ell^m$ to $\mathbb{C}$ parametrized by $f, f(\mathbf{x})$ and *coins*. Let $L = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|L|}\} \subseteq \mathbb{Z}_\ell^m$ be the (candidate) $\frac{\epsilon^2}{4}$-heavy Fourier coefficients returned by $\mathcal{SFT}$. If $f(\mathbf{x}_i) = y$ for some $\mathbf{x}_i \in L$, then $\mathcal{I}$ outputs $\mathbf{x}_i$. (If more than one $\mathbf{x}_i \in L$ satisfies $f(\mathbf{x}_i) = y$, then $\mathcal{I}$ selects one of them arbitrarily.) Otherwise, it fails.

We now analyze the success probability of $\mathcal{I}$. Clearly $\Pr\{f(\mathcal{I}(f,f(\mathbf{x}))) = f(\mathbf{x})\} \geq \Pr\{\mathbf{x} \in L\}$. In order to bound $\Pr\{\mathbf{x} \in L\}$, we consider the Fourier transform of the function h used to answer the queries of $\mathcal{SFT}$, and compute the Fourier coefficient corresponding to $\mathbf{x}$:

$$\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x}) = \mathop{\mathbb{E}}_{\mathbf{r}\leftarrow\mathcal{U}(\mathbb{Z}_\ell^m)} \left[\omega_\ell^{\mathcal{P}(f,f(\mathbf{x}),\mathbf{r};coins)} \overline{\chi_{\mathbf{x}}(\mathbf{r})} \right] = \mathop{\mathbb{E}}_{\mathbf{r}\leftarrow\mathcal{U}(\mathbb{Z}_\ell^m)} \left[\omega_\ell^{[\mathcal{P}(f,f(\mathbf{x}),\mathbf{r};coins)-\mathbf{x}\cdot\mathbf{r}]} \right].$$

Averaging over $f \leftarrow \mathcal{U}(F), f(\mathbf{x})$ and *coins* we get

$$\mathop{\mathbb{E}}_{f,f(\mathbf{x}),coins} \left[\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x}) \right] = \mathop{\mathbb{E}}_{f,f(\mathbf{x}),coins,\mathbf{r}} \left[\omega_\ell^{[\mathcal{P}(f,f(\mathbf{x}),\mathbf{r};coins)-\mathbf{x}\cdot\mathbf{r}]} \right] = \mathbb{E}\left[\omega_\ell^{-k} \mid k \leftarrow \mathcal{E}_\ell(\mathcal{P}) \right].$$

Notice that this is the (complex) bias of the predictor. So, by Jensen's inequality ($|\mathbb{E}[\mathcal{Z}]| \leq \mathbb{E}[|\mathcal{Z}|]$),

$$\mathop{\mathbb{E}}_{f,f(\mathbf{x}),coins} \left[\left|\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x})\right| \right] \geq \left| \mathop{\mathbb{E}}_{f,f(\mathbf{x}),coins} \left[\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x}) \right] \right| = \left|\mathbb{E}\left[\omega_\ell^{-k} \mid k \leftarrow \mathcal{E}_\ell(\mathcal{P}) \right]\right| = \epsilon.$$

This proves that the expected magnitude of the Fourier coefficient $\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x})$ is at least ϵ. By Markov's inequality, $\Pr_{f,f(\mathbf{x}),coins}\{|\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x})| \geq \frac{\epsilon}{2}\} \geq \frac{\epsilon}{2}$. So, with probability at least $\epsilon/2$ (over $f \leftarrow \mathcal{U}(F), \mathbf{x} \leftarrow \mathcal{X}$ and *coins*) $\mathbf{x}$ is a $\frac{\epsilon^2}{4}$-heavy Fourier coefficient of $h_{f,f(\mathbf{x}),coins}$, and it will be included in L with probability at least $\Pr\{f(\mathcal{I}(f,f(\mathbf{x}))) = f(\mathbf{x})\} \geq (\epsilon/2)\cdot(2/3) = \epsilon/3$. Finally, the running time of the inverter is bounded by the running time of $\mathcal{SFT}$ times the running time of the predictor. Since $\|\widehat{h}_{f,f(\mathbf{x}),coins}\|_\infty = 1$, the running time of $\mathcal{SFT}$ is bounded by $\mathsf{poly}(n, \log\ell, 1/\epsilon)$ (see Theorem 2.6) and hence the overall time of the inverter is $\mathsf{poly}(n, \log\ell, 1/\epsilon)\cdot t$. □

The previous lemma uses the technical notion of bias to quantify the quality of a prediction algorithm. The following lemma shows that for the special case of a prime ℓ, it is enough to bound the success probability of the predictor.

Lemma 3.5. *Let $(F,\mathcal{X})$ be a function family with $[\mathcal{X}] \subset \mathbb{Z}^m$. For any prime p, if $(F,\mathcal{X})$ admits a (t,ϵ)-accurate p-predictor, then it also admits a (t,ϵ)-biased p-predictor.*

Proof. Let $\mathcal{P}'$ be the predictor that takes as input $f, f(\mathbf{x})$ and $\mathbf{r} \in \mathbb{Z}_p^m$ and tries to predict $\mathbf{x}\cdot\mathbf{r} \pmod p$ as follows: pick $y \leftarrow \mathcal{U}(\mathbb{Z}_p^*)$, run $z \leftarrow \mathcal{P}(f, f(\mathbf{x}), y\mathbf{r})$, and return z/y. For any k, we have

$$\begin{aligned}
\Pr\{k \leftarrow \mathcal{E}_p(\mathcal{P}')\} &= \Pr\{\mathcal{P}'(f,f(\mathbf{x}),\mathbf{r}) = \mathbf{x}\cdot\mathbf{r} - k \pmod p\} \\
&= \Pr\{y^{-1}\mathcal{P}(f,f(\mathbf{x}),y\mathbf{r}) = \mathbf{x}\cdot\mathbf{r} - k \pmod p\} \\
&= \Pr\{\mathcal{P}(f,f(\mathbf{x}),\mathbf{t}) = \mathbf{x}\cdot\mathbf{t} - yk \pmod p\}
\end{aligned}$$

where $\mathbf{t} = y\mathbf{r}$ has the same distribution as $\mathbf{r}$. Using the accuracy bound, for $k = 0$ we immediately get

$$\Pr\{0 \leftarrow \mathcal{E}_p(\mathcal{P}')\} = \Pr\{\mathcal{P}(f,f(\mathbf{x}),\mathbf{t}) = \mathbf{x}\cdot\mathbf{t} \pmod p\} = \frac{1}{p} + \epsilon.$$

For $k \neq 0$, since y is distributed uniformly at random over $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$, we have

$$\begin{aligned}
\Pr\{k \leftarrow \mathcal{E}_p(\mathcal{P}')\} &= \Pr\{y = (\mathbf{x}\cdot\mathbf{t} - \mathcal{P}(f,f(\mathbf{x}),\mathbf{t}))/k \pmod p\} \\
&= \frac{1}{p-1}\cdot(1 - \Pr\{\mathcal{P}(f,f(\mathbf{x}),\mathbf{t}) = \mathbf{x}\cdot\mathbf{t}\}) \\
&= \frac{1}{p} - \frac{\epsilon}{p-1}.
\end{aligned}$$

Using these expressions and the identity $\sum_{k=0}^{p-1}\omega_p^{-k} = 0$, the bias of $\mathcal{P}'$ is easily computed as

$$\left|\mathbb{E}\left[\omega_p^{-k} \mid k \leftarrow \mathcal{E}_p(\mathcal{P}') \right]\right| = \left| \sum_{k=0}^{p-1} \Pr\{k \leftarrow \mathcal{E}_p(\mathcal{P}')\}\cdot\omega_p^{-k} \right| = \left| \frac{\epsilon p}{p-1} \right| \geq \epsilon.$$

Finally $\mathcal{P}'$ runs in essentially the same time as $\mathcal{P}$. □

Combining the previous two lemmas, we recover as a special case the results of [18, 19] for learning linear functions over a *field* given query access to a noisy version of the function.

Corollary 3.6. *Let $(F, \mathcal{X})$ be a function family with $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$, and p a prime such that $p \geq s$. If there exists a (t, ϵ)-accurate p-predictor for $(F, \mathcal{X})$, then $(F, \mathcal{X})$ is $(\mathsf{poly}(n, \log p, 1/\epsilon) \cdot t, \frac{\epsilon}{3})$-invertible.*

Proof. Easily follows from Lemma 3.4 and Lemma 3.5. □

3.2 From Unpredictability to Pseudorandomness

In this section we prove that, for *knapsack* function families, unpredictability implies pseudorandomness. In other words, we show that, under certain conditions, a distinguisher $\mathcal{D}$ for $\mathcal{K} = \mathcal{K}(G, \mathcal{X})$ with noticeable distinguishing advantage can be turned into a predictor for $\mathcal{K}$ with *noticeable bias*. At a high level, the predictor works as follows: on input a modulus ℓ, function $\mathbf{g} \in G^m$, $y = \mathbf{g} \cdot \mathbf{x} \in G$ and $\mathbf{r} \in \mathbb{Z}_\ell^m$, it first makes a guess for the inner product $\mathbf{x} \cdot \mathbf{r} \bmod \ell$; it then uses that guess to modify the knapsack instance $(\mathbf{g}, y)$, and finally invokes the distinguisher $\mathcal{D}$ on the modified instance $(\mathbf{g}', y')$. The output of $\mathcal{D}$ is used to determine whether the initial guess was correct or not. The same technique was used by Impagliazzo and Naor in [22]. However, in the restricted subset-sum setting considered in [22], the reduction is rather straightforward: if the guess for $\mathbf{x} \cdot \mathbf{r}$ is correct, then the modified knapsack instance $(\mathbf{g}', y')$ is distributed according to $\mathcal{F}(\mathcal{K})$, whereas if the guess is wrong, the distribution of $(\mathbf{g}', y')$ is (statistically close to) uniform. But these are exactly the two distributions that $\mathcal{D}$ can tell apart and therefore a noticeable distinguishing advantage translates directly into an accurate/biased predictor.

When considering general abelian groups and distributions $\mathcal{X}$ with $[\mathcal{X}] \not\subseteq \{0, 1\}^m$, several technical difficulties arise. Unlike [22], if the guess for $\mathbf{x} \cdot \mathbf{r}$ is wrong, then the distribution of $(\mathbf{g}', y')$ can be statistically far from uniform. In fact, $(\mathbf{g}', y')$ can be distributed according to $\mathcal{F}_d(\mathcal{K})$ for any divisor d of the group exponent M_G. Notice that for $d = 1$ and $d = M_G$ we get the two "extreme" distributions $\mathcal{F}_1(\mathcal{K}) = \mathcal{U}(G^m \times G)$ and $\mathcal{F}_{M_G}(\mathcal{K}) = \mathcal{F}(\mathcal{K})$. However, other $\mathcal{F}_d(\mathcal{K})$ (with $1 < d < M_G$) can also arise. Depending on the order and structure of the underlying group, and the output distribution of the distinguisher $\mathcal{D}$ on the various auxiliary distributions $\mathcal{F}_d(\mathcal{K})$, the technical details of the reduction differ significantly. As a warm-up, we first present a weak form of our main Theorem.

Proposition 3.7. *Let $\mathcal{K} = \mathcal{K}(G, \mathcal{X})$ be a knapsack family with $[\mathcal{X}] \subseteq [s]^m$ and $s = \mathsf{poly}(n)$. If $\mathcal{K}$ is (t, δ)-distinguishable from uniform for some noticeable δ, but $\mathcal{K}_d = \mathcal{K}(G_d, \mathcal{X})$ is pseudorandom for all $d < 2ms^2$, then for any prime p with $s \leq p < 2s = \mathsf{poly}(n)$, $\mathcal{K}$ is $(O(t + m), \epsilon, p)$-biased for some noticeable[7] ϵ.*

Proof. Let $\mathcal{D}$ be a (t, δ)-distinguisher, and define $\beta_d = \Pr\{\mathcal{D}(\mathcal{F}_d(\mathcal{K})) = 1\}$. Notice that $\beta_{M_G} = \Pr\{\mathcal{D}(\mathcal{F}(\mathcal{K})) = 1\}$ and $\beta_1 = \Pr\{\mathcal{D}(\mathcal{U}(G^m \times G)) = 1\}$. By assumption, $\beta_{M_G} - \beta_1 = \delta$ (because $\mathcal{D}$ is a (t, δ)-distinguisher for $\mathcal{F}(\mathcal{K})$), while $\beta_d - \beta_1 = \mathsf{negl}(n)$ for all $d < 2ms^2$ (because $\mathcal{K}_d$ is pseudorandom for all $d < 2ms^2$).

Let p be any prime between s and $2s$. The predictor $\mathcal{P}$ is shown as Algorithm 1. Intuitively, the predictor tries to guess the inner product $\mathbf{x} \cdot \mathbf{r}$ *over the integers*. If the guess c is correct, the predictor invokes the distinguisher on input $\mathcal{F}_{M_G}(\mathcal{K}) = \mathcal{F}(\mathcal{K})$, otherwise it invokes $\mathcal{D}$ on $\mathcal{F}_d(\mathcal{K})$ for some $d < m(s-1)p < 2ms^2$. But for all such d, $\mathcal{F}(\mathcal{K}_d)$ and therefore $\mathcal{F}_d(\mathcal{K})$ (by Lemma 2.2) is pseudorandom, so $\mathcal{D}$ will behave as if it had been invoked on the uniform distribution $\mathcal{F}_1(\mathcal{K})$. So, the distinguisher $\mathcal{D}$ will determine (with advantage $\delta - \mathsf{negl}(n)$) if the guess c was correct, and if not, the predictor $\mathcal{P}$ will output a guess other than c.

We show that $\mathcal{P}$ is an ϵ-accurate p-predictor for some nonnegligible ϵ. The Proposition then follows directly from Lemma 3.5. Let $c' = \mathbf{x} \cdot \mathbf{r} = A \cdot p + v$ $(0 \leq v < p)$ be the inner product $\mathbf{x} \cdot \mathbf{r}$ *over the integers*. $\mathcal{P}$ is trying to predict $v = c' \pmod p$. The input to the distinguisher $\mathcal{D}$ (line 4, Algorithm 1) is $(\bar{\mathbf{g}}, R)$ where

$$R = y - cg = \mathbf{g} \cdot \mathbf{x} - cg = \mathbf{g} \cdot \mathbf{x} - c'g + c'g - cg = \mathbf{g} \cdot \mathbf{x} - (\mathbf{r} \cdot \mathbf{x})g + (c' - c)g = \bar{\mathbf{g}} \cdot \mathbf{x} + (c' - c)g$$

By Lemma 2.1, the input to $\mathcal{D}$ is $(\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + d \cdot g) = \mathcal{F}_d(\mathcal{K})$ for $d = \gcd_G(c' - c)$. So, the probability that $\mathcal{D}$ outputs 1 is by definition $\beta_{\gcd_G(c'-c)}$.

[7] Here we do not seek to optimize ϵ as a function of δ, but we mention that the predictor in the proof has bias at least $\epsilon \geq \delta/(2ms^2)$.

input : $(\mathbf{g}, y, \mathbf{r})$ // $y = \mathbf{g} \cdot \mathbf{x},\ \mathbf{r} \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$
output: $guess \in \mathbb{Z}_p$

Pick $c \leftarrow \mathcal{U}(\mathbb{Z}_{m(s-1)p})$;
Pick $g \leftarrow \mathcal{U}(G)$;
$\bar{\mathbf{g}} \leftarrow \mathbf{g} - \mathbf{r} \cdot g$ // $\mathbf{r} \cdot g = (r_1 \cdot g, \ldots, r_m \cdot g)$;
Run $\mathcal{D}$ on input $(\bar{\mathbf{g}}, y - c \cdot g)$;
if $\mathcal{D}$ *outputs 1* **then**
    $guess \leftarrow c \bmod p$;
else
    $guess \leftarrow \mathcal{U}(\mathbb{Z}_p \setminus (c \bmod p))$;
end
return $guess$

Algorithm 1: Predictor for weak reduction (Proposition 3.7)

For any d, let C_d be the event "$\gcd_G(c' - c) = d$". Clearly, $\sum_{d|M_G} \Pr\{C_d\} = 1$. Notice that since $c, c' \in [m(s-1)p]$, we only need to consider either $d = M_G$ (in which case $c' = c$), or small values $d < m(s-1)p < 2ms^2$. (For all other values of d, we have $\Pr\{C_d\} = 0$.) The probability that $\mathcal{P}$ guesses correctly the inner product $\mathbf{x} \cdot \mathbf{r} \pmod p$ is given by

$$\Pr\{guess = v\} = \sum_{d|M_G} \Pr\{guess = v \mid C_d\} \Pr\{C_d\}. \tag{10}$$

Conditioning on the output of $\mathcal{D}$, we get that for every d,

$$\Pr\{guess = v \mid C_d\} = a_d \cdot \beta_d + \frac{1 - a_d}{p-1}(1 - \beta_d) \tag{11}$$

where $a_d = \Pr\{c = c' \pmod p \mid C_d\}$. It immediately follows from the definition that $\sum_{d|M_G} a_d \Pr\{C_d\} = \Pr\{c = c' \pmod p\} = \frac{1}{p}$. Notice that for $d = M_G$ we have $a_{M_G} = 1$, $\Pr\{C_{M_G}\} = 1/(m(s-1)p)$ and $\beta_{M_G} = \beta_1 + \delta$. For all other d (with $\Pr\{C_d\} \neq 0$,) we have $\beta_d = \beta_1 + \mathsf{negl}(n)$. Plugging (11) in (10) and simplifying, we obtain

$$\begin{aligned}
\Pr\{guess = v\} &= \sum_{d|M_G} \Pr\{C_d\} \left(a_d \beta_d + \frac{1 - a_d}{p-1}(1 - \beta_d) \right) \\
&= \Pr\{C_{M_G}\} \left(a_{M_G} \beta_{M_G} + \frac{1 - a_{M_G}}{p-1}(1 - \beta_{M_G}) \right) \\
&\quad + \sum_{d|M_G, d<M_G} \Pr\{C_d\} \left(a_d(\beta_1 + \mathsf{negl}(n)) + \frac{1 - a_d}{p-1}(1 - \beta_1 - \mathsf{negl}(n)) \right) \\
&\geq \frac{1}{p} + \frac{\delta}{m(s-1)p} - \mathsf{negl}(n).
\end{aligned}$$

This proves that the p-predictor is ϵ-accurate for $\epsilon \geq \delta/(m(s-1)p) - \mathsf{negl}(n) \geq \delta/(2ms^2)$. Since p is a prime, by Lemma 3.5, there is also an ϵ-biased predictor with essentially the same running time $O(m+t)$ as $\mathcal{P}$. □

Proposition 3.7 already gives search-to-decision reductions for some interesting families $\mathcal{K}$, but it requires (as an assumption) the pseudorandomness of $\mathcal{K}_d = \mathcal{K}(G_d, \mathcal{X})$ for a larger range of values of d than as specified in Theorem 3.1. The following lemma plays a crucial role in the proof of Proposition 3.9, which extends Proposition 3.7 to hold under the assumptions in the theorem.

Lemma 3.8. *For any $d \in \mathbb{N}$, $\sum_{r|d} r^2 \leq (\pi^2/6) \cdot d^2$.*

Proof. Let $d = r_1 > r_2 > \ldots > r_k = 1$ be all the divisors of d. Clearly it must be $r_i \leq \frac{d}{i}$. It follows that

$$\sum_{i=1}^{k} r_i^2 \leq \sum_{i=1}^{k} \left(\frac{d}{i}\right)^2 \leq d^2 \sum_{i=1}^{\infty} \frac{1}{i^2} = d^2 \cdot \frac{\pi^2}{6}.$$

□

Proposition 3.9. *Let $\mathcal{K} = \mathcal{K}(G, \mathcal{X})$ be a knapsack function family with $[\mathcal{X}] \subseteq [s]^m$ and $s = \mathsf{poly}(n)$. If $\mathcal{K}$ is (t, δ)-distinguishable from uniform for some noticeable δ, but $\mathcal{F}_d(\mathcal{K})$ is pseudorandom for all $d < s$, then $\mathcal{K}$ is $(O(t+m), \epsilon, d^*)$-biased for some noticeable ϵ and polynomially bounded $d^* \geq s$.*

Proof. In order to relax the condition for pseudorandomness of $\mathcal{K}_d$ from any $d \leq 2ms^2$ to any $d < s$, we need to overcome two major technical difficulties. First, guessing the inner product $\mathbf{x} \cdot \mathbf{r}$ over the integers, as done in [22] and Proposition 3.7, is unlikely to work for the following reason: A correct guess for $\mathbf{x} \cdot \mathbf{r}$ produces indeed the distribution $\mathcal{F}(\mathcal{K}) = \mathcal{F}_{M_G}(\mathcal{K})$ as input to the distinguisher as desired; however, a wrong guess produces $\mathcal{F}_d(\mathcal{K})$ for some d smaller than $2ms^2$ *but possibly larger than* s. In that case, we have no guarantee that the distinguishing advantage between $\mathcal{F}_{M_G}(\mathcal{K})$ and $\mathcal{F}_d(\mathcal{K})$ (and therefore the predicting advantage of $\mathcal{P}$) is noticeable. We overcome this difficulty by having $\mathcal{P}$ guess $\mathbf{x} \cdot \mathbf{r} \pmod d$ (instead of over the integers) for some divisor[8] d of M_G (with $s \leq d < 2ms^2$).

For such a divisor d, our predictor runs the distinguisher with input $\mathcal{F}_d(\mathcal{K})$ whenever the guess for $\mathbf{x} \cdot \mathbf{r} \pmod d$ is correct or with $\mathcal{F}_{d'}(\mathcal{K})$ for some $d' \mid d$ $(d' < d)$ when the guess for $\mathbf{x} \cdot \mathbf{r} \pmod d$ is wrong. The second challenge is to actually prove the existence of an appropriate d for which the distinguishing gap of $\mathcal{D}$ between $\mathcal{F}_d(\mathcal{K})$ and $\mathcal{F}_{d'}(\mathcal{K})$ is sufficiently large $\forall d' \mid d$. Notice here that d might be *composite* and hence a predictor that guesses $\mathbf{x} \cdot \mathbf{r} \pmod d$ with probability larger than $1/d + 1/\mathsf{poly}(n)$ *does not* necessarily imply an inverter with noticeable success probability (recall that the results of [19] hold over *fields*). Here is where the power of Lemma 3.4 and Fourier analysis come into the play. What we actually show, is that there exists a (possibly composite) d^* and an associated d^*-predictor that has bias ϵ for some noticeable ϵ. Details follow.

We adopt the notation from proof of Proposition 3.7. Namely for a (t, δ)-distinguisher $\mathcal{D}$ we define $\beta_d = \Pr\{\mathcal{D}(\mathcal{F}_d(\mathcal{K})) = 1\}$. In addition, for brevity, we often write $a \equiv_c b$ instead of $a \equiv b \pmod c$ and define $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

By assumption, $\beta_{M_G} - \beta_1 = \delta$ while $\beta_d - \beta_1 = \mathsf{negl}(n)$ for all $d < s$. We can further assume that there exists $\tilde{d}$ with $s \leq \tilde{d} \leq 2ms^2 = \mathsf{poly}(n)$ such that $\beta_{\tilde{d}} - \beta_1 = \tilde{\delta}$ for some *noticeable* $\tilde{\delta}$ (otherwise the proof follows directly from Proposition 3.7). Let d^* be the *smallest* divisor of $\tilde{d}$ such that[9] $|\beta_{d^*} - \beta_1| \geq \frac{d^{*3}\tilde{\delta}}{\tilde{d}^3}$. Notice that d^* has the following two useful properties: (a) $d^* \geq s$. This is true because $|\beta_{d^*} - \beta_1| \geq \frac{d^{*3}\tilde{\delta}}{\tilde{d}^3} = 1/\mathsf{poly}(n)$ whereas by assumption $|\beta_d - \beta_1| = \mathsf{negl}(n)$ for all $d < s$. (b) $|\beta_{d'} - \beta_1| < \frac{d'^3\tilde{\delta}}{\tilde{d}^3}$ for all $d' \mid d^*$ by definition of d^*. We will use these properties to construct a d^*-predictor $\mathcal{P}$ for $\mathcal{K}$. $\mathcal{P}$ is shown as Algorithm 2. In the remaining of the proof we analyze the error distribution $\mathcal{E}_{d^*}(\mathcal{P})$ of $\mathcal{P}$ and prove that $\mathcal{P}$ has noticeable bias.

Let $c' = \mathbf{r} \cdot \mathbf{x} = A \cdot d^* + v$ $(0 \leq v < d^*)$ be the inner product of $\mathbf{x} \cdot \mathbf{r}$ *over the integers*. The input to the distinguisher $\mathcal{D}$ (line 4, Algorithm 2) is $(\bar{\mathbf{g}}, R)$ where

$$\begin{aligned} R &= y - cg_1 + d^* g_2 = \mathbf{g} \cdot \mathbf{x} - cg_1 + d^* g_2 \\ &= \mathbf{g} \cdot \mathbf{x} - c'g_1 + (c' - c)g_1 + d^* g_2 = \mathbf{g} \cdot \mathbf{x} - (\mathbf{r} \cdot \mathbf{x})g_1 + (c' - c)g_1 + d^* g_2 = \\ &= \bar{\mathbf{g}} \cdot \mathbf{x} + (Ad^* + v - c)g_1 + d^* g_2 \end{aligned}$$

Notice that c is the *initial* attempt (line 1) of $\mathcal{P}$ to guess $\mathbf{x} \cdot \mathbf{r} \pmod{d^*}$ while v is the *actual* value of $\mathbf{x} \cdot \mathbf{r} \pmod{d^*}$. If $c = v$ then, by Lemma 2.1, $\mathcal{D}$ is invoked on $(\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + d^* g) = \mathcal{F}_{d^*}(\mathcal{K})$.
If $c \neq v$ then $\mathcal{D}$ gets $(\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + \gcd(Ad^* + v - c, d^*) \cdot \mathcal{U}(G)) = (\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + d' \cdot \mathcal{U}(G)) = \mathcal{F}_{d'}(\mathcal{K})$ for some $d' \mid d^*$ with $d' < d^*$ (notice that if $c \neq v$, then $d^* \nmid Ad^* + v - c$). More specifically, $d' = \gcd(v - c, d^*)$.

[8] If no divisor $d \in [s, 2ms^2]$ exists, we can apply Proposition 3.7 directly.

[9] Such a d^* always exists. Indeed $\tilde{d}$ itself satisfies this condition and is a divisor of itself.

input : $(\mathbf{g}, y, \mathbf{r})$ // $y = \mathbf{g} \cdot \mathbf{x},\ \mathbf{r} \leftarrow \mathcal{U}(\mathbb{Z}_{d^*}^m)$
output: $guess \in \mathbb{Z}_{d^*}$

Pick $c \leftarrow \mathcal{U}(\mathbb{Z}_{d^*})$;
Pick $g_1 \leftarrow \mathcal{U}(G), g_2 \leftarrow \mathcal{U}(G)$;
$\bar{\mathbf{g}} \leftarrow \mathbf{g} - \mathbf{r} \cdot g_1$ // $\mathbf{r} \cdot g_1 = (r_1 \cdot g_1, \ldots, r_m \cdot g_1)$;
Run $\mathcal{D}$ on input $(\bar{\mathbf{g}}, y - c \cdot g_1 + d^* \cdot g_2)$;
if $\mathcal{D}$ *returns 1* **then**
    $guess \leftarrow c$;
else
    $guess \leftarrow \mathcal{U}(\mathbb{Z}_{d^*} \setminus c)$;
end
return *guess*

Algorithm 2: Predictor for strong reduction (Proposition 3.9)

For all $j \in [d^*]$ let C_j be the event $c \equiv_{d^*} v - j$, i.e. the initial guess c differs from actual v by $j \pmod{d^*}$. Notice that $\Pr\{C_j\} = 1/d^*$ for all $j \in [d^*]$. The error distribution of $\mathcal{P}$ is given by the probabilities $\Pr\{guess \equiv_{d^*} v - k\}$, $k \in [d^*]$. Conditioning on the events C_j, we get

$$\begin{aligned} Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} &= \Pr\{guess \equiv_{d^*} v - k\} \\ &= \sum_{j=0}^{d^*-1} \Pr\{guess \equiv_{d^*} v - k \,\big|\, C_j\} \Pr\{C_j\} \\ &= \frac{1}{d^*} \sum_{j=0}^{d^*-1} \Pr\{guess \equiv_{d^*} v - k \,\big|\, C_j\} \end{aligned} \tag{12}$$

If we further condition on whether $\mathcal{D}$ outputs 1 or 0, it is not hard to see that (recall $\delta_{kj} = 1$ iff $k = j$)

$$\Pr\{guess \equiv_{d^*} v - k \,\big|\, C_j\} = \delta_{kj} \cdot \beta_{\gcd(j,d^*)} + \frac{1 - \delta_{kj}}{d^* - 1}(1 - \beta_{\gcd(j,d^*)})$$

Replacing in (12) gives

$$Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} = \frac{1}{d^*} + \frac{1}{d^*}\beta_{\gcd(k,d^*)} - \frac{1}{d^*(d^* - 1)} \sum_{j \neq k} \beta_{\gcd(j,d^*)}.$$

Notice that

$$Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} - Pr\{1 \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} = \frac{1}{d^* - 1}(\beta_{\gcd(k,d^*)} - \beta_1).$$

Using this and the fact that $\sum_{k=0}^{d^*-1} \omega_{d^*}^{-k} = 0$ we get that

$$\begin{aligned} \left| \sum_{k=0}^{d^*-1} Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} \cdot \omega_{d^*}^{-k} \right| &= \frac{1}{d^* - 1} \left| \sum_{k=0}^{d^*-1} (\beta_{\gcd(k,d^*)} - \beta_1) \omega_{d^*}^{-k} \right| \\ &\geq \frac{1}{d^* - 1} \left[|\beta_{d^*} - \beta_1| - \sum_{k=1}^{d^*-1} \left| \beta_{\gcd(k,d^*)} - \beta_1 \right| \right] \end{aligned} \tag{13}$$

Next we bound $\sum_{k=1}^{d^*-1} \left| \beta_{\gcd(k,d^*)} - \beta_1 \right|$ from above. Define $\Phi(d^*, r) = \{1 \leq i < d^* : \gcd(i, d^*) = r\}$ and let[10]

[10]This is a generalization of Euler's totient function.

$\phi(d^*, r) = |\Phi(d^*, r)|$. Clearly for any divisor d' of d^*, we have $\phi(d^*, d') \leq \frac{d^*}{d'}$. So

$$\sum_{k=1}^{d^*-1} \left|\beta_{\gcd(k,d^*)} - \beta_1\right| \leq \sum_{\substack{d' \mid d^* \\ d'<d^*}} \phi(d^*, d') \left|\beta_{d'} - \beta_1\right| \leq \sum_{\substack{d' \mid d^* \\ d'<d^*}} \frac{d^* d'^3 \tilde{\delta}}{d' \tilde{d}^3} = \frac{d^* \tilde{\delta}}{\tilde{d}^3} \sum_{\substack{d' \mid d^* \\ d'<d^*}} d'^2$$

where in the last inequality we used the fact that for all proper divisors d' of d^*, $|\beta_{d'} - \beta_1| < \frac{d'^3 \tilde{\delta}}{\tilde{d}^3}$. Replacing back in (13) we finally get

$$\begin{aligned}
\left|\sum_{k=0}^{d^*-1} Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} \cdot \omega_{d^*}^{-k}\right| &\geq \frac{1}{d^* - 1}\left[\frac{d^{*3}\tilde{\delta}}{\tilde{d}^3} - \frac{d^*\tilde{\delta}}{\tilde{d}^3} \sum_{\substack{d' \mid d^* \\ d'<d^*}} d'^2\right] \\
&= \frac{d^*\tilde{\delta}}{\tilde{d}^3(d^* - 1)}\left[d^{*2} - \sum_{\substack{d' \mid d^* \\ d'<d^*}} d'^2\right] \\
&\geq \frac{d^{*3}\tilde{\delta}}{\tilde{d}^3(d^* - 1)}\left(2 - \frac{\pi^2}{6}\right) \\
&\geq \frac{1}{\mathsf{poly}(n)}
\end{aligned}$$

where in the penultimate inequality we used Lemma 3.8. □

4 Implications and applications

Theorem 3.1 provides explicit criteria for checking if the output of a knapsack function family is pseudorandom. Given a group G and some input distribution $\mathcal{X}$, one needs only to check that the folded knapsack families $\mathcal{K}_d = \mathcal{K}(G_d, \mathcal{X})$ are pseudorandom. As it turns out, for many choices of $(G, \mathcal{X})$, the folded knapsack functions $\mathcal{K}_d$ *compress* their input and map the input distribution $\mathcal{X}$ to a distribution which is *statistically close* to uniform over G_d. More specifically, $\Delta_U(\mathcal{F}(\mathcal{K}_d)) = \mathsf{negl}(n)$, and $\mathcal{K}_d$ is pseudorandom in a strong statistical sense. Below, we provide some representative examples, focusing on those that are most interesting in applications. But before we do that, it is instructive to digress a little and explore a case where one-wayness does not imply pseudorandomness[11]. Intuitively, in any such counterexample there should exists a divisor d of $|G|$ such that $\mathcal{F}(\mathcal{K}_d)$ can be easily distinguished from the uniform distribution. Lemma 4.1 formalizes this intuition.

Lemma 4.1. *If there exists a group G such that $\mathcal{K}(G, \mathcal{X})$ is one-way, then there exist group G' and distribution $\mathcal{X}'$ such that $\mathcal{K}(G', \mathcal{X}')$ is one-way, but not pseudorandom.*

Proof. Let p be a small prime such that $\gcd(p, M_G) = \gcd(p, |G|) = 1$. Notice that $M_G \leq |G|$ has less than $\log_2 |G|$ distinct prime factors. So, we can always choose p among the first $\log_2 |G|$ primes, and $p = O(\log |G| \log\log |G|)$. (If $|G|$ or M_G is known, such p can be computed by generating the sequence of all primes, and checking each one of them for coprimality. When only an upper bound $M_G \leq B$ is known, and coprimality cannot be efficiently checked, one can find p probabilistically by picking a prime uniformly at random among the first $O(\log B)$ primes.)

Let $G' \simeq G \times \mathbb{Z}_p$, and $\mathcal{X}' = p\mathcal{X} = \{px \mid x \leftarrow \mathcal{X}\}$. First notice that $\mathcal{K}(G', \mathcal{X}')$ is not pseudorandom. In fact, on input a function $\mathbf{g}' = (g'_1, \ldots, g'_m)$ and element $(r'_1, r'_2) \in G'$, a distinguisher simply outputs 1 if and only if $r'_2 = 0$. It is easy to check that distinguisher has advantage $1 - 1/p \geq 1/2$.

[11] Strictly speaking, what we prove is: there exist group G and distribution $\mathcal{X}$ for which $\mathcal{K}(G, \mathcal{X})$ is widely believed to be one-way whereas it is provably not pseudorandom.

Assume now that there exists a (t', ϵ)-inverter $\mathcal{I}'$ for $\mathcal{K}(G', \mathcal{X}')$. Consider the following inverter $\mathcal{I}$ against $\mathcal{K}(G, \mathcal{X})$. On input $\mathbf{g} \leftarrow \mathcal{U}(G^m)$ and target $y = \mathbf{g} \cdot \mathbf{x} \in G$, $\mathcal{I}$ simply picks $\mathbf{b} = (b_1, \ldots, b_m) \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$ and invokes $\mathcal{I}'$ on input $\mathbf{g}' = (g'_1, \ldots, g'_m)$ and y' where $g'_i = (g_i, b_i)$ and $y' = (py, 0)$. Notice that $(\mathbf{g}', y')$ is distributed according to $\mathcal{F}(\mathcal{K}(G', \mathcal{X}'))$, exactly as required by $\mathcal{I}'$. So, the inverter $\mathcal{I}'$ will output a preimage $\mathbf{x}' = (x'_1, \ldots, x'_m)$ of y' with probability ϵ. Notice also that if $(x'_1, \ldots, x'_m)$ is in the support of the input distribution $[\mathcal{X}']$, then p divides $(x'_1, \ldots, x'_m)$. If this is the case, $\mathcal{I}$ outputs $\mathbf{x} = (x_1, \ldots, x_m)$, where $x_i = x'_i/p$ for $i = 1, \ldots, m$. Since $\gcd(p, M_G) = 1$, multiplication by p is an invertible function from G to G. In particular, if the inverter $\mathcal{I}'$ is successful, then $\mathbf{g}'\mathcal{I}'(\mathbf{g}', y') = y'$ implies $\mathbf{g}\mathcal{I}(\mathbf{g}, y) = y$, and the $\mathcal{I}$ is successful too. This proves that $\mathcal{I}$ is a (t, ϵ)-inverter for $t \approx t'$. □

4.1 Specific Groups and Input Distributions

We start with groups G whose order does not contain any factors that are within the maximum value the input can take. In this case one-wayness implies pseudorandomness for *any* input distribution.

Lemma 4.2. *Let p be the smallest prime factor of $|G|$ and $\mathcal{X}$ be such that $[\mathcal{X}] \subseteq [s]^m$ where $s = \mathsf{poly}(n)$ such that $s \leq p$. If $\mathcal{K}(G, \mathcal{X})$ is one-way, then it is also pseudorandom.*

Proof. Consider $\mathcal{K}_d$ for any $d < p$. Since $\gcd(d, |G|) = 1$ for all $d < p$, we have $dG = G$. It follows that the range of $\mathcal{K}_d$ is $G_d = G/dG = \{0\}$ and $\mathcal{K}_d$ is trivially pseudorandom for every $d < p$. The lemma then follows directly from Theorem 3.1. □

Lemma 4.2 is already very powerful. For instance, in the standard subset sum problem we have $[\mathcal{X}] = \{0, 1\}^m \subseteq [p]^m$ for any prime p. In this setting, Lemma 4.2 significantly extends the results from [22] and [14]. More specifically, it asserts that *any* knapsack family $\mathcal{K}(G, \mathcal{X})$ with $[\mathcal{X}] \subseteq \{0, 1\}^m$ is pseudorandom provided it is one-way, for *any* abelian group G and binary (not necessarily uniform) input distribution $\mathcal{X}$. Other settings Lemma 4.2 is directly applicable to include groups with prime order, vector groups of the form $\mathbb{Z}_p^k$ for prime p and more generally groups of the form $\mathbb{Z}_{p^e}^k$ where p is a prime such that $p \geq s = \mathsf{poly}(n)$ where $[\mathcal{X}] \subseteq [s]^m$.

For groups with small prime factors (smaller than s, where $[\mathcal{X}] \subseteq [s]^m$), the connection between one-wayness and pseudorandomness is more subtle. In that case, Lemma 4.1 tells us that in order to prove *search-to-decision* reductions, the group and input distribution need to be *restricted* somehow. Still, our main theorem can be used to prove search-to-decision reductions for a wide range of groups and input distributions. In the rest of this section we give a few representative examples, focusing on *vector groups* $G = \mathbb{Z}_q^k$ both for simplifying the exposition and because these groups are most interesting from a cryptographic viewpoint (see Section 4.2).

For a vector group $G = \mathbb{Z}_q^k$ consider the folded knapsack function $\mathcal{K}_d = \mathcal{K}(G_d, \mathcal{X})$. First notice that $M_G = q$ and $dG = d\mathbb{Z}_q^k = \gcd(d, q) \cdot \mathbb{Z}_q^k \simeq Z_{q/\gcd(d,q)}$. By Theorem 3.1, proving pseudorandomness of $\mathcal{K}(G, \mathcal{X})$ reduces to proving that the folded families $\mathcal{K}_d$ are pseudorandom for all $d < s$ with $d \mid q$. In fact, below we prove that in many interesting settings the function families $\mathcal{K}_d$ are *statistically* random. Lemma 4.3 provides sufficient conditions for pseudorandomness expressed in terms of the statistical properties of $\mathcal{X}$ and the factorization of q: for every "small" divisor d of q, the d-folded distribution $\mathcal{X}_d = \{\mathbf{x} \bmod d \mid \mathbf{x} \leftarrow \mathcal{X}\}$ should have collision probability much smaller than the inverse of the order of the quotient group $|G_d| = |\mathbb{Z}_d^k| = d^k$.

Lemma 4.3. *If $\mathcal{K} = \mathcal{K}(\mathbb{Z}_q^k, \mathcal{X})$ is one-way, $[\mathcal{X}] \subseteq [s]^m$ for some $s = \mathsf{poly}(n)$ and $\mathsf{Col}(\mathcal{X}_d) = \mathsf{negl}(n)/d^k$ for all $d \mid q$ with $d < s$, then $\mathcal{K}$ is also pseudorandom.*

Proof. For any d such that $d \mid q$, $G_d = \mathbb{Z}_q^k/d\mathbb{Z}_q^k = \mathbb{Z}_d^k$. Given Theorem 3.1, it suffices to prove that $\Delta_U(\mathcal{F}(\mathcal{K}(\mathbb{Z}_d^k, \mathcal{X})) = \mathsf{negl}(n)$ for all divisors $d \mid q$ with $d < s$. For that, we can directly apply Lemma 2.3 with $H = G_d = \mathbb{Z}_d^k$, and $H_{\tilde{d}} = \mathbb{Z}_{\tilde{d}}^k$ to get

$$\Delta_U(\mathcal{F}(\mathcal{K}(H, \mathcal{X}))) \leq \frac{1}{2} \sqrt{\sum_{1 < \tilde{d} \mid d} \tilde{d}^k \cdot \mathsf{Col}\left(\mathcal{X}_{\tilde{d}}\right)} = \mathsf{negl}(n)$$

where we used the hypothesis that $\mathsf{Col}\left(\mathcal{X}_{\tilde{d}}\right) = \mathsf{negl}(n)/\tilde{d}^k$ for all $\tilde{d} \mid q$ with $1 < \tilde{d} < s$ and the fact that d has at most $d < s = \mathsf{poly}(n)$ divisors $\tilde{d}$. □

Below we give 2 natural families of distributions which have small collision probability when folded, and thereby result in pseudorandom knapsack families.

Uniformly Folded Distributions. For a given group G we say that a distribution $\mathcal{X}$ with $[\mathcal{X}] \subseteq [s]^m$ is *uniformly folded* with respect to G, if $\mathcal{X}_d \simeq \mathcal{U}(\mathbb{Z}_d^m)$ is (statistically close to) the uniform distribution for all $d < s$ such that $d \mid M_G$.

Lemma 4.4. *If $\mathcal{K} = \mathcal{K}(\mathbb{Z}_q^k, \mathcal{X})$ is one-way, $k \leq m - \omega(\log n)$, and $\mathcal{X}$ is uniformly folded with respect to $\mathbb{Z}_q^k$, (with $[\mathcal{X}] \subseteq [s]^m$ for some $s = \mathsf{poly}(n)$), then $\mathcal{K}$ is also pseudorandom.*

Proof. Directly follows from Lemma 4.3 and from the fact that if $\mathcal{X}_d = \mathcal{U}(\mathbb{Z}_d^m)$, then $\mathsf{Col}\left(\mathcal{X}_d\right) = 1/d^m$. □

Two examples of uniformly folded distributions are $\mathcal{X} = \mathcal{U}(\mathbb{Z}_q^m)$ (with respect to group $G = Z_q^k$ for any q and k) and $\mathcal{X} = \mathcal{U}(\mathbb{Z}_{p^i}^m)$ (with respect to group $G = Z_{p^e}^k$ for prime p and $i \leq e$). As an immediate corollary to Lemma 4.4, we obtain the following.

Corollary 4.5. *Assume $\mathcal{K} = \mathcal{K}(Z_{p^e}^k, \mathcal{U}(\mathbb{Z}_{p^i}^m))$ is a one-way function family for some prime p, and integer parameters $i \leq e$, and m such that $p^i = \mathsf{poly}(n)$. Then $\mathcal{K}$ is pseudorandom.*

Gaussian. Gaussian-like distributions are typically used for sampling the error in LWE-based cryptographic constructions. The following lemma establishes the search-to-decision reduction for knapsack families defined over $\mathbb{Z}_q^k$ with Gaussian-like input distribution. We state the result for *discrete Gaussians* (defined in Section 2.3). Qualitatively similar results hold for *discretized* (rounded) Gaussians as well.

Lemma 4.6. *Let $k \leq m - \omega(\log n)$ and r be the Gaussian width satisfying*[12] *$\omega(\log n) \leq r \leq \mathsf{poly}(n)$. If $\mathcal{K}(\mathbb{Z}_q^k, \mathcal{D}_{\mathbb{Z}^m, r})$ is one-way then it is also pseudorandom provided that* any *of the following two conditions holds:*

(a) q is prime, or

(b) q has no divisors within the interval $\left[\frac{r}{\beta(n)}, r \cdot \beta(n)\right]$ for some function $\beta(n) = \omega(\sqrt{\log n})$.

Proof. (a) When q is a prime, the proof follows directly from Lemma 4.2.
(b) By a standard tail inequality,

$$\Pr_{\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m, r}} \{\exists i \text{ such that } |x_i| > \lfloor r \cdot \beta(n)/2 \rfloor - 1\} = \mathsf{negl}(n).$$

This means that effectively $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m, r}$ takes values in $\mathcal{S}^m$ where $\mathcal{S} = \{-\lfloor r \cdot \beta(n)/2 \rfloor + 1, \ldots, \lfloor r \cdot \beta(n)/2 \rfloor - 1\}$ is an interval of size $s < r \cdot \beta(n) = \mathsf{poly}(n)$. Consider now *any* divisor d of q such that $d < s < r \cdot \beta(n)$. By hypothesis, all divisors of q lie outside the interval $\left[\frac{r}{\beta(n)}, r \cdot \beta(n)\right]$, which implies that $d < r/\beta(n)$. Consider now the d-folded distribution $\mathcal{D}_{\mathbb{Z}^m, r} \bmod d$. For all $i \in [m]$, $x_i \leftarrow \mathcal{D}_{\mathbb{Z}, r} \bmod d$ where $r > d \cdot \beta(n)$. Lemma 2.5 then asserts that $\mathsf{Col}\left(\mathcal{D}_{\mathbb{Z}, r} \bmod d\right) \leq 1/d + \mathsf{negl}(n)$ or equivalently $\mathsf{Col}\left(\mathcal{D}_{\mathbb{Z}^m, r} \bmod d\right) \leq \mathsf{poly}(n)/d^m \leq \mathsf{negl}(n)/d^k$, where we used the fact that $k \leq m - \omega(\log n)$. We can then apply Lemma 4.3 (see also Remark 3.2) to conclude the proof. □

[12] In typical instantiations, $r = \Omega(n^\theta)$ for some constant $\theta > 0$.

4.2 Applications to LWE

In this section, we show how our results for knapsack functions imply similar search-to-decision reductions for the Learning With Errors (LWE) problem with the interesting feature of being *sample-preserving*. Following common notational conventions from the existing LWE literature, we use n for the length of the secret vector $\mathbf{s}$, m for the number of samples, q for the modulus and χ for the error distribution. Let n, m, q be positive integers and χ a distribution over $\mathbb{Z}_q$. For any $q, n \in \mathbb{Z}$, $\mathbf{s} \in \mathbb{Z}_q^n$, and χ, define the distribution

$$\mathcal{A}_{\mathbf{s},\chi} = \{(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e) \mid \mathbf{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n), e \leftarrow \chi\}.$$

We recall that the LWE problem with parameters n, m, q and χ is the problem of recovering $\mathbf{s}$ given m samples from distribution $\mathcal{A}_{\mathbf{s},\chi}$. In the decisional version of LWE (DLWE), one is given m samples drawn independently at random either from $\mathcal{A}_{\mathbf{s},\chi}$ (for some secret $\mathbf{s}$) or from $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. The goal is to tell the two distributions apart with noticeable probability.

We are interested in reductions from LWE to DLWE that *preserve* all the parameters n, m, q, χ, including the *number of samples* m. Sample-preserving reductions are more naturally described using matrix notation for the LWE problem. Given a collection of m LWE samples $(\mathbf{a}_i, b_i) \leftarrow \mathcal{A}_{\mathbf{s},\chi}$, we can combine them in a matrix $\mathbf{A}$ having the vectors $\mathbf{a}_i$ as rows, and a column vector $\mathbf{b}$ with entries equal to b_i. That is, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{e} \leftarrow \chi^m$. With this notation, we want to prove that any algorithm that distinguishes $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ from $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$ can be used to recover the secret $\mathbf{s}$. Notice that once the secret $\mathbf{s}$ has been recovered, one can also recover the error vector $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{s}$. So, we can equivalently define LWE as the problem of recovering both $\mathbf{s}$ and $\mathbf{e}$ from $\mathbf{A}$ and $\mathbf{A}\mathbf{s} + \mathbf{e}$. This is exactly the problem of inverting the following function family.

Definition 4.7. *Let n, m, q be positive integers and χ a probability distribution over $\mathbb{Z}_q$. Let* $\mathrm{LWE}(n, m, q, \chi)$ *be the function family $(F, \mathcal{X})$ where $\mathcal{X} = \{(\mathbf{s}, \mathbf{e}) \mid \mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n), \mathbf{e} \leftarrow \chi^m\}$, and F is the set of functions $f_{\mathbf{A}}$ indexed by $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and defined as $f_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$.*

Similarly, the decision version of LWE is precisely the problem of distinguishing $\mathcal{F}(\mathrm{LWE}(n, m, q, \chi))$ from the uniform distribution $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$. However, $\mathrm{LWE}(n, m, q, \chi)$ is not a knapsack function family. In order to apply the results from Section 3, we exploit the duality between the LWE problem and an associated knapsack function family described in the following lemmas. Since duality between LWE and knapsack functions has been noticed before in the literature [42, 30, 26], here we only sketch the proofs of the lemmas.

Lemma 4.8. *For any*[13] *$n, m \geq n + \omega(\log n), q$ and χ, there is a polynomial time reduction from the problem of inverting* $\mathrm{LWE}(n, m, q, \chi)$ *with probability ϵ, to the problem of inverting $\mathcal{K}(\mathbb{Z}_q^{m-n}, \chi^m)$ with probability $\epsilon' = \epsilon + \mathsf{negl}(n)$.*

Proof Sketch. The transformation from the LWE problem into an equivalent knapsack problem requires that the matrix $\mathbf{A}$ be nonsingular, i.e., the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^n$. When $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$, this is true except with probability at most $1/p^{m-n-1}$, where p is the smallest prime factor of q. So, for $m \geq n + \omega(\log n)$, $\Pr\{\mathbf{A} \text{ is singular}\} = \mathsf{negl}(n)$. We can therefore assume $\mathbf{A}$ has been chosen at random, but conditioned on being nonsingular.

Consider now the set of all vectors $\mathbf{g} \in \mathbb{Z}_q^m$ such that $\mathbf{g}\mathbf{A} = 0 \pmod q$. Under the assumption that $\mathbf{A}$ is nonsingular, this set is generated by the rows of a matrix $\mathbf{G} \in \mathbb{Z}_q^{(m-n) \times m}$ that can be efficiently computed from $\mathbf{A}$ using linear algebra. We can further randomize $\mathbf{G}$ by left-multiplying it by a random unimodular matrix $\mathbf{U} \in \mathbb{Z}_q^{(m-n) \times (m-n)}$. Finally, if $\mathbf{A}$ is chosen at random among all nonsingular matrices, then this randomized $\mathbf{G}$ is also distributed uniformly at random among all matrices whose *columns* generate $\mathbb{Z}_q^{m-n}$. As before, the distribution of $\mathbf{G}$ is within negligible statistical distance from $\mathcal{U}(\mathbb{Z}_q^{(m-n) \times m})$, so we can treat the columns of $\mathbf{G}$ as random elements from the vector group $G = \mathbb{Z}_q^{m-n}$. Finally, we set $\mathbf{c} = \mathbf{G}\mathbf{b} = \mathbf{G}\mathbf{A}\mathbf{s} + \mathbf{G}\mathbf{e} = \mathbf{G}\mathbf{e}$, so the distribution $(\mathbf{G}, \mathbf{c})$ is *statistically close* to a *random instance* of the knapsack problem with group $G = \mathbb{Z}_q^{m-n}$ and input distributed according to the error distribution χ^m. □

[13] The requirement $m \geq n + \omega(\log n)$ is a standard assumption in the context of LWE, where typically $m \geq n + \Omega(n)$.

Lemma 4.9. *For any $n, m \geq n + \omega(\log n), q$ and χ, there is a polynomial time reduction from the problem of distinguishing $\mathcal{F}(\mathcal{K}(\mathbb{Z}_q^{m-n}, \chi^m))$ from uniform with advantage ϵ to the problem of distinguishing $\mathcal{F}(\mathrm{LWE}(n, m, q, \chi))$ from uniform with advantage $\epsilon' = \epsilon + \mathsf{negl}(n)$.*

Proof Sketch. The distinguisher for the knapsack function is obtained similarly, transforming the knapsack instance into a corresponding LWE one. This transformation essentially reverses the steps taken to transform LWE into knapsack. We start from a pair $(\mathbf{G}, \mathbf{c})$. As before, we can assume without loss of generality (up to negligible statistical error) that the columns of $\mathbf{G}$ generate $\mathbb{Z}_q^{m-n}$. Next, by linear algebra, we compute a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ whose columns generate the set of vectors $\mathbf{a} \in \mathbb{Z}_q^m$ such that $\mathbf{Ga} = \mathbf{0} \pmod q$. As before, we can randomize $\mathbf{A}$ by right-multiplying it by a random unimodular matrix $\mathbf{U} \in \mathbb{Z}_q^{n \times n}$ to obtain $\mathbf{A}'$. We also map $\mathbf{c}$ to $\mathbf{A}'\mathbf{s}' + \mathbf{r}$ where $\mathbf{s}' \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $\mathbf{r}$ is an arbitrary solution to the equation $\mathbf{Gr} = \mathbf{c}$. It can be checked that this transformation maps the knapsack distribution $(\mathbf{G}, \mathbf{c} = \mathbf{Ge})$ to the LWE distribution $(\mathbf{A}', \mathbf{A}'\mathbf{s} + \mathbf{e})$ (with uniformly random $\mathbf{s}$), when $\mathbf{G}$ and $\mathbf{A}'$ are chosen at random subject to the constraint that they are nonsingular. The transformation also maps the uniform distribution to a (statistically close to) uniform distribution. So, by feeding $(\mathbf{A}', \mathbf{A}'\mathbf{s} + \mathbf{e})$ into an LWE distinguisher, we get a distinguisher for the knapsack function with essentially the same (up to negligible terms, due to nonsingular matrices) distinguishing advantage of the LWE distinguisher. □

We remark that reductions exist also in the directions opposite to those described in Lemma 4.8 and Lemma 4.9, but this is all we need here.

Sample-preserving reductions for LWE, i.e., reductions from the problem of inverting $\mathrm{LWE}(n, m, q, \chi)$ to the problem of distinguishing $\mathcal{F}(\mathrm{LWE}(n, m, q, \chi))$ from uniform, are immediately obtained combining the reductions from Lemma 4.8 and Lemma 4.9 with the results from Section 3 on $\mathcal{K}(\mathbb{Z}_q^{m-n}, \chi^m)$. Similarly to the knapsack case, the reductions *do not hold unconditionally*; rather they hold for specific, yet very broad, moduli q and error distributions χ. Below we provide some examples of such moduli q and distributions χ. Throughout, it is assumed that $m \geq n + \omega(\log n)$.

Proposition 4.10. *Assume there exists an efficient algorithm $\mathcal{D}$ that distinguishes between $\mathcal{F}(\mathrm{LWE}(n, m, q, \chi))$ and $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$ with noticeable advantage. Then there exists an efficient algorithm $\mathcal{I}$ that inverts $\mathrm{LWE}(n, m, q, \chi)$ with noticeable success probability in any of the following cases:*

(i) Binary modulus $q = 2$ and any *error distribution χ over $\{0, 1\}$.*

(ii) Prime modulus $q = \mathsf{poly}(n)$ and any *error distribution χ over $\mathbb{Z}_q$.*

(iii) Prime power modulus $q = p^e$ for prime $p = \mathsf{poly}(n)$, and χ such that $[\chi] \subseteq \{-(p-1)/2, \ldots, (p-1)/2\}$.

(iv) Prime power modulus $q = p^e$ where χ is the uniform distribution over $\mathbb{Z}_{p^i}$ for some $i < e$ such that $p^i = \mathsf{poly}(n)$.

Proof. The proof for all cases follows easily by combining Lemmas 4.8 and 4.9 with the results for bounded knapsack families from Section 4.1. In particular cases (i), (ii) and (iii) are direct applications of Lemma 4.2 whereas case (iv) is immediate from Corollary 4.5. □

Remark 4.11. *Case (i) from Proposition 4.10 provides a sample-preserving search-to-decision reduction for the Learning Parity with Noise (LPN) problem. Such a reduction was already given in [7]. In contrast, other reductions appearing in the literature [10, 23] do not preserve the number of samples. Using $q = \mathsf{poly}(n)$ as in case (ii) and Gaussian error distribution χ over $\mathbb{Z}_q$ is typical in LWE-based cryptographic applications. In fact, these parameters were used in the first LWE-based semantically secure scheme by Regev [38] who also presented a (non sample-preserving) search-to-decision reduction. Case (iii) provides a sample-preserving version of the search-to-decision reduction proved in [6]. Finally, the search-to-decision reduction for LWE with modulus and noise distribution as in case (iv) appears to be new; no such (even non-sample-preserving) reduction has previously appeared in the literature. Setting $q = 2^\ell$ and χ to be the uniform distribution over $\mathbb{Z}_{2^{\ell'}}$ for some $\ell' = O(\log n)$ seems very appealing since arithmetic modulo 2 and sampling over uniform distributions can be implemented very efficiently in practice.*

5 Open Problems

Our work leaves many interesting open questions. To start with, sample-preserving search-to-decision reductions for LWE with *bounded* noise as considered in this work, don't seem to extend to the *unbounded* noise regime, i.e. when each coefficient e_i of the error vector $\mathbf{e}$ of LWE is drawn from a set with *superpolynomial* size. We note that such search-to-decision reductions are known [35] but are *not* sample-preserving. These reductions rely heavily on a Chinese Reminder Theorem (CRT) approach: using a *perfect*[14] distinguisher, they first learn the secret modulo p_i with *overwhelming success probability* for each *polynomially bounded* prime factor p_i of the modulus q; they then use the CRT to recover the entire secret. In sample-preserving reductions, where only an *imperfect* distinguisher is available, learning the secret modulo p_i can be performed in a much looser, list-decoding sense: the projection of the secret modulo p_i is included in the corresponding lists L_i but among possibly *many* other elements. And the only way to check which of the list elements corresponds to the actual projection of the secret modulo p_i seems to be first forming the entire secret using CRT and then verifying that the result is the LWE secret. Thus, one has to solve a superpolynomial number of CRT instances before recovering the correct value of the secret. It would be nice to extend the list-decoding approach to work even in that case.

As an additional motivation, we mention that extending our sample-preserving reductions to the unbounded error setting would pave the road to similar results for the Ring LWE (R-LWE) problem [28]. R-LWE is an algebraic variant of LWE that leads to much more efficient constructions than standard LWE while still enjoying strong security guarantees. Much like LWE with unbounded noise, existing search-to-decision reductions [28] decompose the secret (which is an element from a ring R) modulo $\mathfrak{q}_i$ where the $\mathfrak{q}_i$s are prime ideal factors.

Our work also highlights the importance of understanding the hardness of LWE under various noise distributions. Current hardness proofs for search LWE [38] based on worst-case lattice problems rely on the noise following a Gaussian distribution. Can lattice-based hardness results for search LWE be extended to noise distributions other than Gaussian? Can we show similar lattice-based hardness results if the noise is distributed uniformly at random modulo 2^i? The latter case is very attractive from a practical viewpoint since arithmetic modulo powers of 2 and sampling from uniform distributions can be implemented very efficiently.

6 Acknowledgments

A preliminary version of this work appears in the Proceedings of CRYPTO 2011 [31]. This is the full version of the paper. This research was supported in part by NSF under grants CNS-0831536 and CNS-0716790. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient Lattice (H)IBE in the Standard Model. In *EUROCRYPT*, pages 553–572, 2010.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In *CRYPTO*, pages 98–115, 2010.

[3] Adi Akavia. *Learning Noisy Characters, Multiplication Codes and Hardcore Predicates*. PhD thesis, MIT, February 2008.

[4] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. Proving Hard-Core Predicates Using List Decoding. In *FOCS*, pages 146–157, 2003.

[14] By perfect here we mean a distinguisher with advantage almost 1. Getting a perfect distinguisher out of an imperfect one (one with only a nonnegligible advantage) is the main reason for the blowup in the number of samples the reduction consumes.

[5] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In *TCC*, pages 474–495, 2009.

[6] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *CRYPTO*, pages 595–618, 2009.

[7] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with Constant Input Locality. *J. Cryptology*, 22(4):429–469, 2009.

[8] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP*, 2011. Available at http://www.eccc.uni-trier.de/report/2010/066/.

[9] Avrim Blum, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly Learning DNF and Characterizing Statistical Query Learning using Fourier Analysis. In *STOC*, pages 253–262, 1994.

[10] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In *CRYPTO*, pages 278–291, 1993.

[11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *FOCS*, page to appear, 2011.

[12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In *EUROCRYPT*, pages 523–552, 2010.

[13] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-Key Encryption Schemes with Auxiliary Inputs. In *TCC*, pages 361–381, 2010.

[14] Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *EUROCRYPT*, pages 245–255, 1996.

[15] Craig Gentry and Shai Halevi. Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits. In *FOCS*, page to appear, 2011.

[16] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A Simple BGN-Type Cryptosystem from LWE. In *EUROCRYPT*, pages 506–522, 2010.

[17] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206, New York, NY, USA, 2008. ACM.

[18] Oded Goldreich and Leonid A. Levin. A Hard-Core Predicate for All One-Way Functions. In *STOC*, pages 25–32, 1989.

[19] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning Polynomials with Queries: The Highly Noisy Case. In *Foundations of Computer Science (FOCS)*, pages 294–303, 1995.

[20] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the Learning with Errors Assumption. In *ICS*, 2010.

[21] R. Impagliazzo and D. Zuckerman. How to Recycle Random Bits. In *FOCS*, pages 248–253, Washington, DC, USA, 1989. IEEE Computer Society.

[22] Russell Impagliazzo and Moni Naor. Efficient Cryptographic Schemes Provably as Secure as Subset Sum. *J. Cryptology*, 9(4):199–216, 1996.

[23] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and Concurrent Security of the HB and HB$^+$ Protocols. *J. Cryptology*, 23(3):402–421, 2010.

[24] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit Cryptosystems Based on Lattice Problems. In *Public Key Cryptography*, pages 315–329, 2007.

[25] Eyal Kushilevitz and Yishay Mansour. Learning Decision Trees Using the Fourier Sprectrum. In *STOC*, pages 455–464, 1991.

[26] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *CT-RSA*, pages 319–339, 2011.

[27] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*, pages 577–594, 2009.

[28] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In *EUROCRYPT*, pages 1–23, 2010.

[29] Daniele Micciancio. *The LLL Algorithm: Survey and Applications*, chapter Cryptographic Functions from Worst-Case Complexity Assumptions, pages 427–452. Information Security and Cryptography. Springer, December 2009.

[30] Daniele Micciancio. Duality in Lattice Based Cryptography. In *Public Key Cryptography*, 2010. Invited Talk.

[31] Daniele Micciancio and Petros Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *CRYPTO*, pages 465–484, 2011.

[32] Daniele Micciancio and Oded Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[33] Daniele Micciancio and Oded Regev. Lattice-Based Cryptography. In *Post Quantum Cryptography*, pages 147–191. Springer Publishing Company, 2009.

[34] Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning Juntas. In *STOC*, pages 206–212, 2003.

[35] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In *STOC*, pages 333–342, New York, NY, USA, 2009. ACM.

[36] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO*, pages 554–571, Berlin, Heidelberg, 2008. Springer-Verlag.

[37] Chris Peikert and Brent Waters. Lossy Trapdoor Functions and Their Applications. In *STOC*, pages 187–196, New York, NY, USA, 2008. ACM.

[38] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of ACM*, 56(6):34, September 2009. Preliminary version in STOC 2005.

[39] Oded Regev. The Learning with Errors Problem (Invited Survey). In *IEEE Conference on Computational Complexity*, pages 191–204, 2010.

[40] Markus Rückert and Michael Schneider. Estimating the Security of Lattice-based Cryptosystems. Technical Report 2010/137, IACR ePrint archive, 2010.

[41] Daniel Stefankovic. Fourier Transform in Computer Science. Master's thesis, University of Chicago, October 2000.

[42] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In *ASIACRYPT*, pages 617–635, 2009.