UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Improving Cryptographic Constructions Using Coding Theory

Permalink

https://escholarship.org/uc/item/47j1j4wp

Author

Mol, Petros

Publication Date

2013

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Improving Cryptographic Constructions Using Coding Theory

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Petros Mol

Committee in charge:

Professor Daniele Micciancio, Chair
Professor Mihir Bellare
Professor Sanjoy Dasgupta
Professor Massimo Franceschetti
Professor Alex Vardy

2013

Copyright
Petros Mol, 2013
All rights reserved.

The dissertation of Petros Mol is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2013

DEDICATION

To Thanasis, Stavroula and Giorgos.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Chapter 3 is a reprint of the paper "Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions" [120], co-authored with Scott Yilek, published in the proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010). Both authors contributed equally to this paper.

Chapters 4 and 5 are a reprint of the paper "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions" [115] co-authored with Daniele Micciancio, published in the proceedings of the 31st Annual Cryptology Conference (CRYPTO 2011). The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, is a reprint of the paper "Secret-Key Authentication Beyond the Challenge-Response Paradigm: Denitional Issues and New Protocols" [119] co-authored with Stefano Tessaro. The dissertation author was the primary investigator and author of the relevant part of the paper.

VITA

2006	BEng in Electrical and Computer Engineering National Technical University of Athens, Greece
2011	Master of Science in Computer Science University of California, San Diego, CA, USA
2013	Doctor of Philosophy in Computer Science University of California, San Diego, CA, USA

ABSTRACT OF THE DISSERTATION

Improving Cryptographic Constructions Using Coding Theory

by

Petros Mol

Doctor of Philosophy in Computer Science

University of California, San Diego, 2013

Professor Daniele Micciancio, Chair

Despite having evolved as two distinct research areas, cryptography and coding theory have matured in parallel, deeply influencing each other leading to a long and successful intertwined history. This thesis explores further the connection between the two fields by demonstrating how borrowing appropriate tools from coding theory can have significant implications to cryptography. Concretely, we present three results, all motivated by cryptographic applications.

First, we prove that CCA-security, the standard security goal for public-key encryption, is implied by Lossy Trapdoor Functions (LTDFs) with minimal lossiness. LTDFs, introduced by Peikert and Waters (STOC 2008), have been extremely successful in cryptography. In a surprising application of LTDFs, Peikert and Waters show how to build CCA-secure encryption generically from LTDFs that

lose a $(1-1/\omega(\log n))$ fraction of their input bits. We drastically lower the lossiness required, showing that any LTDF that loses *only a noticeable fraction of a single bit* suffices. The key idea behind our result is the use of Reed-Solomon codes to appropriately instantiate a recent CCA-secure construction by Rosen and Segev (TCC 2009).

Second, we present powerful and general *sample preserving* search to decision reductions for the Learning With Errors (LWE) problem, introduced by Regev (STOC 2005), and used to substantially expand the scope of lattice based cryptography. Such reductions are of paramount importance in cryptography, bridging the gap between constructions, which require hard *decisional* problems, and hardness assumptions, which rely on *search* problems. Our proof draws upon recently developed techniques that generalize the list-decoding algorithm of Goldreich and Levin (STOC 89) to Hadamard codes over larger alphabets.

In our last result, we use Learning Parity with Noise (LPN), a problem closely related to that of decoding random linear codes, to construct a simple and efficient 3-round symmetric authentication protocol that is secure against active attacks. Symmetric authentication has recently attracted widespread interest due to the existence of lightweight protocols amenable to implementation on simple architectures, such as RFID tags. Compared to existing LPN-based protocols, ours achieves a better security-efficiency tradeoff leading to smaller communication complexity and key sizes for the same level of security.

Chapter 1

Introduction

Cryptography is the mathematical study of secure communication in the presence of adversaries. Its key goals are data confidentiality, integrity and authentication. Coding theory is the rigorous study of *codes* and is primarily concerned with the problem of communicating *efficiently* and *reliably* over an unreliable channel. Central to both fields is the concept of *protecting information* transmitted between a *sender* and a *receiver* in the face of external conditions that hinder the communication.

The main difference between the two fields lies in the fact that the contexts within which information needs to be protected are in principle disparate. In cryptography, the external conditions are modeled by a malicious entity, the *adversary*, who intercepts the communication between legitimate parties in an attempt to derive information about the messages exchanged.[1] The tremendous success of modern cryptography as a field is, to a large extent, a result of proposing realistic notions of security by properly modeling the power and capabilities of the adversary. This endeavor has been greatly assisted by the concurrent evolution of *computational complexity* and the formalization of *computational feasibility*, a key concept that laid out the theoretical foundations for rigorous security definitions. The de-facto context in cryptographic applications assumes the existence of adversaries that, while (possibly) acting arbitrarily and unpredictably, are confined to feasible computation. This paradigm shift led to the introduction of several formal

[1] This description is mostly tailored to adversaries against encryption schemes for simplicity.

notions, motivated by real-world cryptographic applications and supported by (efficient) protocols that fulfill them and paved the way for the field of *provably secure cryptography*. Along with this new perception of the external environment, came the need for new definitions, suitable for addressing the computational nature of the adversary. A substantial step towards this goal, was taken by Goldwasser and Micali [71] who introduced the classic notion of *semantic security*. The latter, roughly, requires that a cryptosystem hides *all partial information* of the encrypted message from all *computationally bounded* adversaries offering a computational analogue of Shannon's "perfect secrecy".

In the meanwhile, coding theory has adopted a different perspective. The external environment is modeled by the *channel*, the physical means through which information is transfered. While the channel can act "adversarially" by introducing errors to the messages transmitted, there is no notion of bounded computation attached to it. Instead, the physical properties of the channel are typically modeled in one of the following two ways: in the first model, introduced by Shannon [149], the channel is a random process which, independently of the transmitted messages, introduces errors according to a known probability distribution. Another, more conservative model was proposed by Hamming [74] who considers "adversarial" channels that can introduce errors in arbitrary positions so long as the total number of errors is bounded. We note that, computation, and especially feasible computation, is immaterial to both models even when the channel adversarially chooses which positions to corrupt (as in Hamming's model). Regardless of the underlying model, a central goal of coding theory is the development of (efficient) codes (known as *error correcting codes*) that allow the correction of a large number of errors while keeping the built-in redundancy of the transmitted messages to a minimum.

Even though cryptography and coding theory have evolved as two distinct research areas mostly owing to the aforementioned difference, the techniques and tools developed within the two communities have matured in parallel. As a result, since the seminal works of Shannon [149] and Hamming [74], the two fields have deeply influenced each other, mutually borrowing ideas, leading to more than 60

years of successful intertwined history. Below, we provide a brief overview of examples where methods from one field contributed to the development of the other. We focus primarily on how ideas from coding theory have been used in cryptography, not only because, historically, the relationship of the two fields has been unbalanced towards this direction, but mostly because this is the direction immediately related to the contributions presented in the thesis.

Cryptography in coding theory. There have been only a few examples where tools and ideas developed within cryptography have been used to solve problems in coding theory. All these examples are fairly recent and rely on the *computationally bounded channel model*, introduced by Lipton [102] as an intermediate model between Shanon's binary symmetric channel [149] and Hamming's adversarial model [74]. According to Lipton's model, errors are introduced in a worst-case fashion just like in Hamming's model but by a *computationally bounded* adversary who can corrupt up to a fixed number of the codeword's entries. Restricting the power of the channel to feasible computation paved the way for the use of cryptographic tools into the problem of error correction. However, the first useful result came only (more than) a decade later by Micali *et al* [112] who provided formal definitions on the requirements of such a channel and presented constructions of codes that can decode from error-rates beyond the classical bounds. The idea of modeling the channel with computationally bounded adversaries has since proven useful in other settings, most notably in locally-decodable codes [126, 76, 77].

The only, to our knowledge, work that deviates from the line of research following Lipton's computationally bounded channel model is a recent result by Bellare, Tessaro and Vardy [19] who revisited the wiretap channel model [161] under a cryptographic angle. Bellare *et al* introduced new, stronger and more realistic security definitions for the wiretap channel, inspired by the classic cryptographic notion of semantic security [71], and presented explicit constructions of schemes that achieve the new definitions based on randomness extractors, an object with long history in cryptography.

Coding theory in cryptography. On the other hand, the adoption of ideas and techniques from coding theory within cryptography has proven significantly more fruitful, benefiting a wide range of cryptographic applications. Roughly, the contributions of coding theory to cryptography can be divided into three categories: First, tools from coding theory have been extensively used to increase the resilience of cryptographic protocols, leading to elegant solutions to several problems in secure computation and threshold cryptography. A recurring goal in such protocols is the ability to reconstruct a secret when only partial (or corrupted) information is available. This setting closely resembles the typical use-case scenario of error-correcting codes whose built-in redundancy allows to reconstruct the initial message even after the latter has been transmitted through a noisy channel. Perhaps the most representative example, and one of the earliest applications of error-correcting codes in cryptography, are secret sharing schemes (SSS). A secret-sharing scheme is a protocol for distributing a secret among multiple users, so that the secret can be reconstructed only when a sufficient number of shares are combined together. In his seminal paper, Shamir [148] presented the first secret sharing scheme based on polynomial interpolation, an application now broadly viewed as a cryptographic twist on the Reed-Solomon family of codes [138] (this was first formalized in [111]). Since then, the connection between SSS (and secure computation in general) and error-correcting codes has been explored further leading to a series of constructions based on a variety of error correcting codes [45, 111, 89, 33, 39, 40]. Besides SSS, error-correcting codes have been used in the construction of other cryptographic objects including local randomizers and t-resilient functions [108], fuzzy commitments [86], extractors [157, 156, 50, 49] and more.

The impact of coding theory on cryptography is also due to the wealth of algorithmic ideas developed within the coding theory community over the years. Cryptanalysis is an area that has particularly benefited from these advances, with techniques such as Information Set Decoding (ISD) [135, 98, 99, 153, 109, 22, 14] and statistical decoding [84, 128] being powerful additions to cryptanalysts' toolbox. Another fundamental concept from coding theory whose use in cryptography

has been met with great success is the notion of list-decoding [56], an enhancement of standard (unique) decoding where a *list* of multiple messages is output among which (at most) one matches the initially transmitted message. The most prominent application of list-decoding in cryptography is the construction of a universal hard-core predicate for any one-way function by Goldreich and Levin [67] who presented an efficient list-decoding algorithm for Hadamard codes. This connection between list-decoding and hard-core predicates was later studied in more depth by Akavia *et al.* [4].

Finally, coding theory has traditionally been a great source of problems upon the hardness of which several cryptographic protocols have relied. The use of (hard) problems from coding theory has developed to such an extent that has led to an entire area, known as *code-based cryptography*, a promising alternative to number-theoretic cryptography for the post-quantum era. The most popular example is McEliece's cryptosystem [110], proposed in 1978 as one of the very first candidates for public key encryption. Its security is based on the assumption that the generator matrix of Goppa codes is pseudorandom. Ever since it was proposed, the assumption was used (in various forms) in several refinements of the initial scheme, achieving stronger notions, as well as in the construction of other cryptographic primitives including digital signatures [41], CCA2-secure encryption [95, 53] and oblivious transfer [54]. Another famous class of problems with long history in cryptography are those related to the hardness of decoding random linear codes. From this class, two problems have been particularly successful: *syndrome decoding* and the celebrated *Learning Parity with Noise* (LPN). Applications of syndrome decoding include identification schemes [154], pseudorandom generators [59], hash functions [11] and more. Similarly, LPN has proven extremely useful especially in symmetric key cryptography with applications to pseudorandom generators [8], encryption schemes [66], authentication protocols [81, 87, 94], Message Authentication Codes [94], commitments and zero-knowledge [85] (see also [7] and [52] for uses of LPN in public key encryption).

1.1 Summary of Contributions

The current thesis explores further the connection between the two fields. In particular, we demonstrate how borrowing the right machinery from coding theory can have significant implications to cryptography. We emphasize that, while the role of coding theory is central in all our results, cryptographic applications remain the exclusive motivation and focus of this thesis. Therefore, we are not concerned with defining new concepts or discovering new techniques in coding theory. Rather, we use existing tools and ideas and present novel ways of using them to substantially improve certain aspects of some cryptographic applications. We remark however that, even though our starting point are problems directly related to cryptography with no apparent connection to coding theory, the use of the latter is an indispensable component of the solutions we propose: the use of the appropriate tool results to improvements that would have otherwise been either minor or even impossible.

In a snapshot, our contributions are the following (see subsequent sections for further explanations and the corresponding chapters for full details): On the construction front, we show how to use Reed-Solomon codes to construct (in a black box way) CCA-secure encryption schemes from Lossy Trapdoor Functions (LTDFs) with minimal lossiness (Chapter 3). We also present efficient authentication protocols based on the hardness of Learning Parity with Noise (LPN), a well-studied variant of the problem of decoding random linear codes (Chapter 6). For the same security level, our protocol features smaller key-sizes and lower communication complexity over the previously known LPN-based protocols. In our final result (Chapters 4 and 5), we describe general and powerful sample preserving search to decision reductions for the Learning With Errors (LWE) problem, a problem that has recently attracted widespread interest in cryptography. Our reductions rely heavily on recently developed list-decoding algorithms that extend previously known algorithms to Hadamard codes over larger alphabets.

Structure of the thesis. In the remaining of the chapter we provide some background and motivation for the cryptographic applications of interest and give a more detailed overview of our contributions, pointing out, along the way, the ideas

and techniques from coding theory that led to each result. We review some definitions and background required in the rest of the thesis in Chapter 2. In Chapter 3, we show how to achieve Chosen-Ciphertext Security from Slightly LTDFs. Chapters 4 and 5 present sample preserving search to decision reductions for the LWE problem. We conclude in Chapter 6, where we present an efficient actively-secure symmetric key authentication protocol based on the hardness of LPN.

1.2 Chosen-Ciphertext Secure Encryption from Slightly Lossy Trapdoor Functions

Lossy Trapdoor Functions (LTDFs) were recently introduced by Peikert and Waters [132, 133] and have since proven to be a very powerful tool both for improving the construction of traditional cryptographic primitives and for constructing new ones. Informally, a family of LTDFs is a standard injective trapdoor function family with the additional property that (the description of) a member function f from the family is computationally indistinguishable from the description $\hat{f}$ of another function that *statistically* loses information about its input. In other words, unlike f, $\hat{f}$ is non-injective, i.e., there exist inputs that map to the same image under $\hat{f}$. Abusing terminology, we say that f (computationally) *loses* ℓ bits[2] if the effective range size of the indistinguishable function $\hat{f}$ is at most a $1/2^{\ell}$-fraction of its domain size.

The latter property – indistinguishability from functions that statistically lose information about their inputs – turns out to be very useful in security reductions giving rise to the following simple proof technique: in the honest execution of a protocol we use the injective function to get the correct functionality, while in the security proof the "challenge" given to the adversary is formed using the lossy function. One can then do a statistical argument to complete the proof. Using this simple and elegant idea, Peikert and Waters showed how to use LTDFs to construct one-way injective trapdoor functions, collision-resistant hash functions, hard-core predicates and functions as well as CPA-secure encryption.

[2]We refer to ℓ as the *lossiness* of f.

In the same paper, Peikert and Waters, provide generic constructions of encryption schemes that are secure against *Chosen-Ciphertext Attacks*[3] (CCA), a security notion introduced by Rackoff and Simon [137] and now widely viewed as the gold standard in public key encryption. CCA-secure schemes are much harder to construct than CPA-secure ones since they should remain robust even against powerful adversaries that can interact with the decryptor via queries. More specifically, the adversary is allowed to query the decryptor on any ciphertext including those that depend on the challenge ciphertext or on the replies to previous queries (to rule out the trivial attack, the only exception is the challenge ciphertext itself). To achieve CCA security from LTDFs, Peikert and Waters introduce an intermediate abstraction, namely *all-but-one trapdoor functions* (ABOs), and use the latter as the building block for their CCA-secure construction. They finally show how to construct ABOs generically from LTDFs whose lossy functions lose enough information about their inputs. In their case, "enough" turns out to be almost all of the input bits, a property that can be difficult to achieve. Indeed, while their DDH-based construction of LTDFs achieves the required amount of lossiness, their latticed-based construction turns out to be insufficient for the general construction. To get CCA security from lattice-based assumptions, the authors of [133] resort to a complex direct construction of an ABO, which defeats the purpose of using LTDFs as a candidate to achieve CCA-security in an elegant and generic way.

In an attempt to address the aforementioned shortcomings, Rosen and Segev [144] introduced a computational analogue of ABO based on the notion of *one-wayness under correlated inputs*. A function family is one-way under correlated inputs if, sampling multiple functions independently from the family and evaluating them on correlated (non-independent) inputs still results in a function that is hard to invert. Rosen and Segev went on to show how to achieve CCA security from function families that are one-way with respect to specific distributions of correlated inputs. Of course, this notion is only useful if there exist functions that are one-way under such correlations. To that end, Rosen and Segev show that LTDFs that are suffieciently lossy satisfy this requirement. However, the amount of

[3]By CCA we will always mean CCA2.

lossiness they require turns out to be approximately as high as the amount needed for the construction of ABO functions from LTDFs, which, as already mentioned, is more than any constant fraction of the input bits, ruling out numerous LTDFs.

Our Results. We significantly extend the results of [132] and [144] and show that *only a non-negligible fraction of a single bit of lossiness is sufficient* for building one-way injective trapdoor functions, CPA-secure encryption, and, perhaps most surprisingly, even CCA-secure encryption. Our results on CCA security drastically improve upon the previous results by lowering the required lossiness from a $(1 - 1/\omega(\log n))$-fraction of *all* the input bits to just a $1/\mathsf{poly}$ fraction of *a single* bit. This solves an open problem from (the most recent version [131] of) [132] and further supports the advanatages of the correlated product formalization of Rosen and Segev. As an additional contribution that highlights the usefulness of our result, we construct a family of LTDFs that loses only 1 bit of its input based on a number-theoretic assumption. Interestingly, attempting to construct LTDFs that lose bigger parts of the input renders the assumption wrong.

On a technical level, our results rely on two core ideas: First, we exploit the fact that LTDFs enjoy some type of *lossiness amplification*. In particular, we show a straightforward way to take an LTDF that loses less than 1 bit and construct an LTDF that loses $\mathsf{poly}(n)$ bits. Second, we observe that if we instantiate an alternative construction by Rosen and Segev with the appropriate error-correcting code, namely Reed-Solomon codes, we can achieve CCA-security from functions that are one-way under correlated input distributions with *very high entropy*. Thankfully, one-wayness under correlated input distributions with high entropy turns out to be a much milder requirement achievable (in a black box way) by LTDFs that are only slightly lossy.

1.3 Sample Preserving Search to Decision Reductions for the Learning With Errors Problem

The Learning With Errors (LWE) problem, introduced by Regev in [139], is the problem of recovering a secret n-dimensional 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. Among other attractive features, LWE enjoys strong security guarantees, supported by worst-case/average-case connections [139, 129, 31], 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. As a result, 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 [139, 91, 130] and active [132, 129] attacks, (hierarchical) identity based encryption [63, 37, 1, 2], digital signatures [63, 37], oblivious transfer protocols [130], several forms of leakage resilient encryption [5, 8, 47, 69], (fully) homomorphic encryption [62, 61, 32, 30] 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 [139], if recovering (with high[4] 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 $b_i \in \mathbb{Z}_q$ is uniformly and independently distributed. Compactly, LWE can be 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 [139] shows that if the LWE function family is uninvertible, then it is also a good pseudorandom

[4] 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".

generator. However, the reduction in [139] somehow hides an important detail: the value of m for which the function is assumed to be uninvertible 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. Indeed, recent attacks [118, 10, 101] indicate that, for certain ranges of the parameters, the number of available samples can have a significant impact on the computational hardness of the LWE problem. 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. (See Section 5.1, Chapter 5 for a more detailed discussion on the importance of sample-preserving reductions for LWE.)

Our Results. The discussion above 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? Our main result 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.

1.3.1 Pseudorandom Generators from Knapsack Functions.

Let $(\mathbb{G}, +)$ be a finite abelian group and $\mathbf{g} = (g_1, \ldots, g_m) \in \mathbb{G}^m$ be a sequence of group elements chosen uniformly at random. The group elements in the sequence $\mathbf{g}$ define a linear 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 $\mathbb{G}$, $f_{\mathbf{g}}$ is conjectured to be an uninvertible 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 [83], Impagliazzo and Naor showed that for some specific, but representative, choices of the group $\mathbb{G}$, 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 $\mathbb{G}^{m+1}$, when $\mathbf{g} \in \mathbb{G}^m$ and $\mathbf{x} \in \{0,1\}^m$ are chosen uniformly at random.

We generalize the results of [83] in two ways: First, we consider functions over *arbitrary* finite groups $\mathbb{G}$. Only cyclic groups of the form $\mathbb{Z}_N$ were considered in [83]. Second, we consider generalizations of the subset-sum function (typically referred to as "knapsack" functions) where the input coefficients x_i take values from a set $\{0,\ldots,s\}$ (or, more generally $\{-s,\ldots,s\}$) for any (*polynomially bounded*) s, rather than just $\{0,1\}$. Moreover, we consider *arbitrary* (possibly nonuniform) input distributions. Both extensions are essential for the sample-preserving search to decision LWE reduction, which requires the pseudorandomness of the knapsack function family over vector groups of the form $\mathbb{G} = \mathbb{Z}_q^k$, and for inputs $\mathbf{x}$ following a nonuniform (Gaussian) distribution over a sufficiently large set $\{-s,\ldots,s\}$.

Our main technical result (Theorem 4.3.1) shows that for any finite abelian group $\mathbb{G}$ and input distribution $\mathcal{X}$, the output of the knapsack function is pseudorandom provided the following two conditions hold: (a) $f_{\mathbf{g}}$ is computationally hard to invert with respect to input distribution $\mathcal{X}$, and (b) 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 $\mathbb{G}_d = \mathbb{G}/d\mathbb{G}$ for some $d \in \mathbb{Z}$,) have pseudorandom output. The power of our result lies in the fact that, for many interesting groups and input distributions, condition (b) is satisfised in a strong *statistical* sense (without any computational assumptions) so that the uninvertibility of the bounded knapsack function directly implies that knapsacks are good pseudorandom generators. We present specific

groups and input distributions for which this holds in Chapter 4.

Techniques. Here we briefly discuss the main technical ideas behind Theorem 4.3.1. A detailed overview along with the formal proof can be found in Section 4.3, Chapter 4. Our proof follows the blueprint of the one by Impagliazzo and Naor [83] for the pseudorandomness of the subset-sum function. Namely, we reduce the indistinguishability of $f_{\mathbf{g}}$ to its uninvertibility in two steps using the notion of *unpredictability* as an intermediate goal. However, generalizing from cyclic to *arbitrary* groups and, more significantly, from binary to larger inputs requires more advanced machinery and fresh technical insights. The core tool in proving that uninvertibility implies unpredictability is a powerful, recently developed, algorithm by Akavia et al. [4] (stated in Section 2.6), that essentially generalizes the list-decoding algorithm of [67] to Hadamard codes over larger (than {0,1}) alphabets. For the second step of the proof, we apply a non-trivial hybrid argument involving distributions that depend on the structure of the group and show that any distinguisher between two consecutive hybrid distributions gives rise to a predictor for $f_{\mathbf{g}}$. The two steps are explained in depth in Section 4.3.

1.3.2 Pseudorandomness of the LWE Function

Our results for LWE are obtained using the duality between LWE and the knapsack function over vector groups. 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 $\mathbf{e}$. Therefore, search to decision reductions for the knapsack functions can be readily translated into corresponding *sample-preserving* search to decision reductions for LWE. In particular, as a direct corollary to our main theorem, we get search to decision reductions for many interesting cases including (among others): any prime modulus q and *any polynomially bounded* error distribution, prime power modulus $q = p^e$ (for any prime $p = \mathsf{poly}(n)$) and arbitrary input distribution over $\mathbb{Z}_p = \{-(p-1)/2, \ldots, (p-1)/2\}$, prime power modulus $q = p^e$ (for any polynomially bounded prime p) and *uniform* error dis-

tribution over $\mathbb{Z}_{p^d}$ for some $d = O(\log_p n)$ and more (see Sections 5.1 and 5.4 for more examples and their proofs).

These results subsume (see Section 5.2 for an in-depth comparison with related work) several previous pseudorandomness results for LWE [139, 8] and LPN [90] but with an important difference. While the proofs in [139, 8, 90] require that LWE (resp. LPN) be 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[5].

1.4 An Actively Secure Authentication Protocol from Learning Parity with Noise

Consider two parties $\mathcal{P}$ (prover) and $\mathcal{V}$ (verifier), sharing a secret key K, and communicating over an insecure channel. $\mathcal{P}$ wishes to prove to $\mathcal{V}$ that he knows the key K, but no adversary $\mathcal{E}$ (eavesdropper), without knowledge of K, should be able to persuade $\mathcal{V}$ that he knows K. Research on this problem, known as *symmetric authentication*, has recently gained momentum, driven by the discovery of lightweight authentication protocols suitable for implementation on RFID devices [87, 34, 55, 122, 65, 90, 94, 48] and the existence of numerous ubiquitous-computing applications (item-labeling, payment systems, proximity cards just to name a few) requiring the existence of RFID tags that are capable of authenticating themselves to a reader.

Symmetric authentication can be achieved via the following simple 2-round *challenge-response* protocol, using s block cipher E (such as DES or AES) with a secret key K: in the first round, the verifier sends a random *challenge* R to the prover, which, upon receiving R, replies with $E_K(R)$. The verifier accepts if and only if the prover's response is the unique correct value. Provided the block cipher is a sufficiently strong message authentication code, this simple protocol achieves the strongest notion of *man-in-the-middle* (MIM) security: Roughly speaking, MIM

[5]For LPN, a sample-preserving reduction was presented in [9].

security requires that an adversary interacting at will with an arbitrary number of both prover and verifier instances cannot later bring a further verifier instance to accept.

Unfortunately, mainstream block-cipher designs such as AES are not well-suited for implementation on lightweight hardware such as RFID tags which are extremely simple devices (typically circuits with a few thousand gates). Seeking for alternatives, Juels and Weis [87] were the first to point out that a very simple protocol by Hopper and Blum [81] (called HB) is secure under the well-known Learning Parity with Noise (LPN) assumption[6] and can be implemented with very low hardware complexity. Yet, HB happens to only satisfy a fairly weak notion of security, called *passive security*, where an adversary observing transcripts of honest prover-verifier interactions cannot convince a further verifier instance that he knows the key. Every attempt to design HB-like protocols with MIM security [34, 55, 122, 65] turned out to miss a security proof, which, very often, resulted in the discovery of a fatal flaw [64, 127]. All *provably* MIM-secure protocols to date [94, 48] are challenge-response protocols derived from the construction of a suitable MAC[7]. While these elegant constructions do improve upon block-cipher based schemes, their hardware complexity remains far from that of the HB protocol.

The need for weaker security: Active security. To overcome the above gap, previous work has focused on an intermediate security notion, called *active security*, where one asks that even an adversary which can interact with the prover arbitrarily fails in later convincing a verifier that he knows the key. This is the secret-key version of the standard security notion for public-key identification schemes dating back to Fiat and Shamir [58] and has recently attracted the interest of cryptographic community [94, 48, 78]. Simply put, active security appears to have become a de-facto standard security notion, backed by the existence of very

[6]The (decisional) LPN assumption with error η asserts that for a random secret $\mathbf{s} \in \{0,1\}^n$, it is computationally hard to distinguish random independent $(n+1)$-bit strings from samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \in \{0,1\}^n$ is random and $e \in \{0,1\}$ is 1 with probability η.

[7]With one exception, perhaps, being the protocol where the verifier sends the encryption of a random plaintext to the prover under a OW-CCA-secure encryption scheme, and the prover returns its decryption; to the best of our knowledge, however, no efficient instantiations of this paradigm are known.

efficient protocols achieving it, its widespread acceptance in the public-key setting, and the inherent hardness of achieving anything stronger such as MIM security efficiently.

Our Results. We focus on the problem of achieving active security and, in particular, on the construction of protocols that are secure under the LPN assumption. Such protocols are very attractive in practice mainly due to the following two reasons: first, they are suitable for implementation on lightweight devices thanks to their low communication and computation complexity. Second, they are based on LPN, a standard, well-studied problem related to the problem of decoding random linear codes from coding theory which is known to be NP-hard in the worst-case [20]. Yet, we only know of two such schemes to date: The HB^+ [87] protocol and the recent two-round protocol by Kiltz *et al* [94]. From the perspective of *concrete security*, both however suffer from drawbacks: On the one hand, no "tight" security reduction to LPN is known for HB^+: if LPN is ϵ-hard for secret length n and complexity t, we can only prove that an active attacker with time complexity (roughly) t cannot break security of HB^+ for key length $2n$ with probability larger than $\sqrt{\epsilon}$. This looseness in the security reduction is undesirable. Kiltz *et al* [94] did take a substantial step towards solving this issue by presenting a protocol which enjoys a tight reduction to LPN in terms of the *advantage* ϵ, yet, if we assume as above that LPN is ϵ-hard for secret-size n, for their protocol to be ϵ-secure too, even under the most optimistic instantiation of their parameters, their key size becomes larger than $4n$ bits and the communication complexity is larger than the one of HB^+.

We hence ask the question: *Can we obtain the best of both worlds?* In other words, under the assumption that LPN is ϵ-hard for secret-size n, can we have an ϵ-secure protocol with key size and complexity comparable to HB^+? We answer this in the affirmative as long as we are interested in basic active security (only involving 2-phase adversaries, see Section 6.3 for the formal security definition). Concretely, we propose a new *generic* approach to obtain an efficient 3-round authentication protocol based on any *weak* MAC, i.e., a MAC which can be evaluated on *random* messages and which must be unforgeable on fresh, *random* messages.

When instantiated with an LPN-based weak MAC, our protocol is extremely simple and works as follows (omitting concrete parameters and technical details): The prover and the verifier share two binary vectors $\mathbf{s}_1, \mathbf{s}_2$ serving as their secret key. The prover first selects a random binary matrix $\mathbf{A}_1$ and sends it to the verifier. The verifier then picks another random binary matrix $\mathbf{A}_2$ and a binary vector $\mathbf{e}_1$ of low Hamming weight and sends $(\mathbf{A}_2, \mathbf{A}_1\mathbf{s}_1 + \mathbf{e}_1)$ to the prover. Upon receiving a pair $(\mathbf{A}_2, \mathbf{z}_1)$, the prover checks whether $\mathbf{z}_1 - \mathbf{A}_1\mathbf{s}_1$ has low Hamming weight, and if so, picks another binary low-weight vector $\mathbf{e}_2$, and sends $\mathbf{A}_2\mathbf{s}_2 + \mathbf{e}_2$ back to the verifier. Finally, the verifier, on input $\mathbf{z}_2$, accepts if and only if $\mathbf{z}_2 - \mathbf{A}_2\mathbf{s}_2$ has low weight. In terms of efficiency, our protocol has communication complexity only minimally larger than HB$^+$, but enjoys a tight reduction to LPN. In addition, for the same security level, it has lower communication complexity and at least 2 times smaller keys than the protocol of Kiltz *et al.*

Chapter 2

Preliminaries

NOTATION. 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 letters 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 $s \in \mathbb{N}$, $[s]$ is the set of the first s nonnegative integers, i.e., $[s] = \{0, 1, \ldots, s-1\}$. For asymptotic statements, we will use n (or λ when n is reserved for other quantities) to denote the security parameter.

2.1 Probability Background

We write $x \leftarrow \mathcal{X}$ both for the operation of selecting x according to a probability distribution $\mathcal{X}$ and for sampling the output x of a 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 set X, value $x \in X$ and probability distribution $\mathcal{X}$ over X, $\Pr[\, x \leftarrow \mathcal{X} \,]$ is the probability associated to x by the distribution $\mathcal{X}$. The *uniform* distribution over a set X is denoted $\mathcal{U}(X)$, and the *support* of a distribution $\mathcal{X}$ is denoted $[\mathcal{X}] = \{x \in X \mid \Pr[\, x \leftarrow \mathcal{X} \,] > 0\}$. If an element x is

sampled uniformly at random from a set X, we will sometimes write $x \xleftarrow{\$} X$ instead of $x \leftarrow \mathcal{U}(X)$ for brevity. 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.

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 $\mathcal{Y}$ defined over a (countable) 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 randomized) function f. Two distributions $\mathcal{X}, \mathcal{Y}$ are ϵ-*close* if $\Delta(\mathcal{X}, \mathcal{Y}) < \epsilon$. They are (t, ϵ)-*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, $\mathcal{X}, \mathcal{Y}$ 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) $\mathcal{U}(X)$.

Entropy and Bounds. We use Ber_η for the *Bernoulli* distribution with parameter η, i.e., Ber_η is a distribution over bits such that $\Pr[\, 1 \leftarrow \mathsf{Ber}_\eta \,] = \eta$. Accordingly, Ber_η^m is the distribution over $\{0,1\}^m$ where each bit is independently distributed according to Ber_η. For some of our bounds, it will be useful to work with the *binary entropy function*, defined as $\mathrm{H}_2(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2 (1-p)$ as well as the (binary) *relative entropy* function with parameters p and q defined as

$$\mathrm{D}(p \,||\, q) = p \cdot \log_2 \left(\frac{p}{q} \right) + (1-p) \cdot \log_2 \left(\frac{1-p}{1-q} \right) .$$

We will use the following form of the *Chernoff* bound. Let $\mathsf{X}_1, \ldots, \mathsf{X}_m$ be independent random binary variables such that $\mathbb{E}[\, \mathsf{X}_i \,] = q \; \forall \, i \in [m]$. If $\mathsf{X} = \sum_{i=1}^m \mathsf{X}_i$, then for any $p > q$,

$$\Pr[\, \mathsf{X} > p \cdot m \,] \leq 2^{-\mathrm{D}(p \,||\, q) \cdot m} . \tag{2.1}$$

For a random variable X with distribution $\mathcal{X}$, we define its *min-entropy* as

$$\mathrm{H}_\infty(\mathsf{X}) = -\log(\max_{x\in[\mathcal{X}]} \Pr[\, x \leftarrow \mathcal{X}\,]).$$

where $\max_{x\in[\mathcal{X}]} \Pr[\, x \leftarrow \mathcal{X}\,] = 2^{-\mathrm{H}_\infty(\mathsf{X})}$ denotes the *predictability* of the random variable X.

Another useful notion of entropy is the *average min-entropy* (defined in [49]) of a random variable X (given another random variable Y with distribution $\mathcal{Y}$) which is defined as follows:

$$\begin{aligned}\tilde{\mathrm{H}}_\infty(\mathsf{X}|\mathsf{Y}) &= -\log\big(\mathop{\mathbb{E}}_{y\leftarrow\mathcal{Y}}\big[\, 2^{-\mathrm{H}_\infty(X\,|\,\mathsf{Y}=y)}\,\big]\big)\\ &= -\log\Big(\mathop{\mathbb{E}}_{y\leftarrow\mathcal{Y}}\Big[\max_{x\in[\mathcal{X}]}\Pr[\, x\leftarrow\mathcal{X}\mid \mathsf{Y}=y\,]\Big]\Big)\end{aligned}$$

The average min-entropy expresses the average maximum probability of predicting X given Y. The following lemma gives a useful bound on the remaining entropy of a random variable X conditioned on a value of Y.

Lemma 2.1.1 ([49], Lemma 2.2b)**.** *Let* $\mathsf{X},\mathsf{Y},\mathsf{Z}$ *be random variables such that* Y *takes at most* 2^k *values. Then*

$$\tilde{\mathrm{H}}_\infty(\mathsf{X}\mid(\mathsf{Y},\mathsf{Z})) \geq \tilde{\mathrm{H}}_\infty((\mathsf{X},\mathsf{Y})\mid\mathsf{Z}) - k \geq \tilde{\mathrm{H}}_\infty(\mathsf{X}|\mathsf{Z}) - k.$$

In particular, if X *is independent of* Z *then* $\tilde{\mathrm{H}}_\infty(\mathsf{X}\mid(\mathsf{Y},\mathsf{Z})) \geq \mathrm{H}_\infty(\mathsf{X}) - k$.

The following lemma (proved in [49]) provides the conditions under which one can derive almost uniform bits from weakly random sources with high entropy.

Lemma 2.1.2 (**The Generalized Leftover Hash Lemma**)**.** *Let* $\mathcal{H}\colon X\to Y$ *be a* universal family *of hash functions and* h *a random variable with uniform distribution over* $\mathcal{H}$*. For any random variables* $\mathsf{X}\in X$ *and* Z *(independent of* h*),*

$$\Delta((h,h(\mathsf{X}),\mathsf{Z}),(h,\mathcal{U}(X),\mathsf{Z})) \leq \frac{1}{2}\sqrt{2^{-\tilde{\mathrm{H}}_\infty(\mathsf{X}|\mathsf{Z})}\cdot|Y|}$$

Function families. A function family $\mathsf{F}=(F,\mathcal{X})$ 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 domain $X\supseteq[\mathcal{X}]$. For simplicity, 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 here) is rather straightforward. Each function family $(F, \mathcal{X})$ naturally defines a probability distribution

$$\mathcal{F}(F, \mathcal{X}) = \{(f, f(x)) \mid f \leftarrow \mathcal{U}(F), x \leftarrow \mathcal{X}\} \tag{2.2}$$

obtained by selecting a function uniformly at random and evaluating it at an input randomly chosen according to $\mathcal{X}$.

A function family $\mathsf{F} = (F, \mathcal{X})$ is (t, ϵ)-*invertible* if there exists a (probabilistic) algorithm $\mathcal{I}$ running in time at most t such that

$$\Pr[\, x' = x \mid f \leftarrow \mathcal{U}(F), x \leftarrow \mathcal{X}, x' \leftarrow \mathcal{I}(f, f(x))\,] \geq \epsilon$$

We then say that $\mathcal{I}$ is a (t, ϵ)-*inverter* for F. If no such inverter exists, we say that F is (t, ϵ)-*uninvertible*. A (t, ϵ)-*pseudorandom generator family*[1] is a function family $(F, \mathcal{X})$ such that the associated distribution $\mathcal{F}(F, \mathcal{X})$ defined in (2.2) is (t, ϵ)-pseudorandom.

Asymptotics. We use n as a (security) parameter that controls all other quantities. 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 probability ensembles $\mathcal{X} = (\mathcal{X}_n)_{n \in \mathbb{N}}$ and $\mathcal{Y} = (\mathcal{Y}_n)_{n \in \mathbb{N}}$ are *statistically close* (denoted $\mathcal{X} \simeq_s \mathcal{Y}$) if $\mathcal{X}_n$ and $\mathcal{Y}_n$ are $\epsilon(n)$-close for some negligible function $\epsilon(n) = \mathsf{negl}(n)$. The ensembles $\mathcal{X}$

[1] Notice that the functions in a pseudorandom generator family are *not* pseudorandom functions, as they do not accept any input besides the (randomly generated) seed $x \leftarrow \mathcal{X}$. Each function $f \in F$ works like a pseudorandom generator which, 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 thesis, by pseudorandom family, we will always mean a pseudorandom generator family. We also remark that 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.

and $\mathcal{Y}$ are *computationally indistinguishable* (denoted $\mathcal{X} \simeq_c \mathcal{Y}$) 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 $\mathsf{F} = (F, \mathcal{X}) = (F_n, \mathcal{X}_n)_{n\in\mathbb{N}}$ 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 $\mathsf{F} = (\mathsf{F}_n)_{n\in\mathbb{N}}$ is *uninvertible* if F_n is $(t(n), \epsilon(n))$-uninvertible for any $t(n) = \mathsf{poly}(n)$ and some $\epsilon(n) = \mathsf{negl}(n)$. It is *pseudorandom* if the associated distribution ensemble $\mathcal{F}(F, \mathcal{X})$ 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 Cryptographic Definitions

Trapdoor Functions. We define injective trapdoor functions (TDFs) and also two different security roperties for TDFs: one-wayness and lossiness. Note that this somewhat departs from papers on lossy trapdoor functions in that we first define an injective trapdoor function as a syntactic object and then define security properties of the syntactic object, instead of mixing the two into one definition.

Definition 2.2.1 (Injective Trapdoor Functions)**.** *A collection of injective trapdoor functions is a tuple of PT algorithms $\mathcal{F} = (G, F, F^{-1})$ such that the (probabilistic) algorithm G outputs a pair (s, t) consisting of a function index s and a corresponding trapdoor t. The deterministic algorithm F, on input a function index s and $x \in \{0,1\}^n$ outputs $f_s(x)$. Finally, algorithm F^{-1}, given the trapdoor t, computes the inverse function $f_s^{-1}(\cdot)$.*

Definition 2.2.2 (One-Way Trapdoor Functions)**.** *Let λ be a security parameter and $\mathcal{F} = (G, F, F^{-1})$ be a collection of injective trapdoor functions with domain $\{0,1\}^{n(\lambda)}$. Let $\mathcal{X}(1^\lambda)$ be a distribution over $\{0,1\}^{n(\lambda)}$. We say $\mathcal{F}$ is* one-way with respect to $\mathcal{X}$ *if for all PPT adversaries A and every polynomial $p(\cdot)$ it follows that*

for all sufficiently large λ

$$\Pr\left[\, A(1^\lambda, s, F(s,x)) = F^{-1}(t, F(s,x)) \,\right] < \frac{1}{p(\lambda)}\ ,$$

where $(s,t) \xleftarrow{\$} G(1^\lambda)$ *and* $x \xleftarrow{\$} \mathcal{X}(1^\lambda)$.

Definition 2.2.3 (Lossy Trapdoor Functions)**.** *Let* λ *be a security parameter and* $\mathcal{F} = (G, F, F^{-1})$ *be a collection of injective trapdoor functions with domain* $\{0,1\}^{n(\lambda)}$*. We say that* $\mathcal{F}$ *is* $(n(\lambda), \ell(\lambda))$*-lossy if there exists a PPT algorithm* $\hat{G}$ *that, on input security parameter* 1^λ*, outputs* $\hat{s}$ *and* $\hat{t}$ *such that*

- *The first outputs of* G *and* $\hat{G}$ *are computationally indistinguishable.*
- *For any* $(\hat{s}, \hat{t})$ *output by* $\hat{G}$*, the map* $F(\hat{s}, \cdot)$ *has image size at most* $2^{n-\ell}$*.*

In the definition above, we call ℓ the *lossiness*. Also, will sometimes call a TDF that is lossy a lossy trapdoor function (LTDF).

Public-Key Encryption. A public-key encryption scheme is a triplet $\mathcal{A}E = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of PPT algorithms. The key generation algorithm $\mathcal{K}$, on input the security parameter 1^λ, outputs a pair of keys (pk, sk). The encryption algorithm $\mathcal{E}$ gets as its input the public key pk and a message $m \in \mathcal{M}$ (for some message space $\mathcal{M}$) and outputs a ciphertext c. The decryption algorithm $\mathcal{D}$ on input the secret key sk and a ciphertext c, outputs a message m or $\perp$ (failure). It is required that $\Pr\left[\, \mathcal{D}(\mathsf{sk}, \mathcal{E}(\mathsf{pk}, m)) \neq m \,\right] = \mathsf{negl}(\lambda)$, where the probability is taken over the randomness of $\mathcal{K}, \mathcal{E}$ and $\mathcal{D}$.

A standard security requirement for a public key cryptosystem $\mathcal{A}E = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is indistinguishability of ciphertexts under a chosen plaintext attack (IND-CPA) [70]. We define IND-CPA security as a game between and adversary $\mathcal{A}$ and an environment as follows. The environment runs $\mathcal{K}(1^n)$ to get a keypair (pk, sk) and flips a bit b. It gives pk to $\mathcal{A}$. $\mathcal{A}$ outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_0|$. The environment returns the challenge ciphertext $c \xleftarrow{\$} \mathcal{E}(\mathsf{pk}, m_b)$ to $\mathcal{A}$ and $\mathcal{A}$ returns a guess bit b'.

We say that $\mathcal{A}$ wins the above game if $b' = b$. Likewise, we define the IND-CPA *advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A},\mathcal{A}E}(\lambda) = 2 \cdot \Pr\left[\, \mathcal{A} \textit{ wins} \,\right] - 1\ .$$

We say that $\mathcal{A}E$ is CPA-secure if $\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A},\mathcal{A}E}(\lambda)$ is negligible in λ for all PPT adversaries $\mathcal{A}$.

Additionally, we can consider a stronger notion of security called indistinguishability under (adaptive) chosen-ciphertext attack (IND-CCA) [124, 137] . The IND-CCA security game is the same as above but with the additional property that throughout the entire game the adversary has access to a decryption oracle **Dec** that, on input c, outputs $\mathcal{D}(\mathsf{sk}, c)$. The one restriction we place on the adversary is that it may not query the challenge ciphertext to the decryption oracle, as this would lead to a trivial win. We define the IND-CCA advantage of an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathcal{A},\mathcal{A}E}(\lambda) = 2 \cdot \Pr\left[\, \mathcal{A} \textit{ wins} \,\right] - 1 \ .$$

We say that $\mathcal{A}E$ is CCA-secure if $\mathsf{Adv}^{\text{ind-cca}}_{\mathcal{A},\mathcal{A}E}(\lambda)$ is negligible in λ for all PPT adversaries $\mathcal{A}$.

2.3 Abelian Groups and Knapsack Function Families.

Abelian Groups. In this thesis, by group we always mean *finite abelian group*. We also assume that certain operations involving groups, such as sampling uniformly at random a group element, adding two elements or multiplying a group element with a scalar, can be efficiently performed. We use additive notation for groups; $0_{\mathbb{G}}$ is the *neutral element*, $|\mathbb{G}|$ is the *order* (size) of $\mathbb{G}$ and $\mathrm{M}_{\mathbb{G}}$ is its *exponent*, i.e., the smallest positive integer e such that $e \cdot g = 0_{\mathbb{G}}$ for all $g \in \mathbb{G}$. 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 $\mathbf{x}, \mathbf{y} \in \mathsf{R}^n$ with elements in a ring R, and also to take integer linear combinations $\mathbf{x} \in \mathbb{Z}^n$ of a vector $\mathbf{y} \in \mathbb{G}^n$ with elements in an additive group. For $\mathbf{x} = (x_1, \ldots, x_n) \in \mathsf{R}^n$ and $a \in \mathsf{R}$, we define $a \cdot \mathbf{x} = \mathbf{x} \cdot a = (x_1 \cdot a, \ldots, x_n \cdot a) \in \mathsf{R}^n$.

For any group $\mathbb{G}$ and (positive) integer d, $\mathbb{G}_d$ is the quotient group $\mathbb{G}/d\mathbb{G}$ where $d\mathbb{G}$ is the subgroup $\{d \cdot g \;\big|\; g \in \mathbb{G}\}$, in analogy with the usual notation $\mathbb{Z}_d = \mathbb{Z}/d\mathbb{Z}$ for the group of integers modulo d. Likewise, for any element $g \in \mathbb{G}$,

$g \bmod d\mathbb{G}$ (or just $g \bmod d$) is the image of g under the natural homomorphism from $\mathbb{G}$ to $\mathbb{G}_d$. For any integer vector $\mathbf{w} = (w_1, \ldots, w_r) \in \mathbb{Z}^r$, we write $\gcd_{\mathbb{G}}(\mathbf{w}) = \gcd(w_1, \ldots, w_r, \mathrm{M}_{\mathbb{G}})$ for the greatest common divisor of the elements of $\mathbf{w}$ and the group exponent. We recall that any finite abelian group $\mathbb{G}$ is isomorphic to a product of cyclic groups $\mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$ where $k_i | k_{i+1}$ for all i, and $k_\ell = \mathrm{M}_{\mathbb{G}}$. If $\mathbb{G} \simeq \mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$, then $\mathbb{G}_d \simeq \mathbb{Z}_{d_1} \times \mathbb{Z}_{d_2} \times \cdots \times \mathbb{Z}_{d_\ell}$ where $d_i = \gcd(d, k_i)$ for $i = 1, \ldots, \ell$. In particular

$$|\mathbb{G}_d| = \Pi_{i=1}^{\ell} d_i \qquad \text{and} \qquad |d\mathbb{G}| = \frac{|\mathbb{G}|}{|\mathbb{G}_d|} = \prod_{i=1}^{\ell} \frac{k_i}{d_i} \,. \tag{2.3}$$

Lemma 2.3.1. *For any group $\mathbb{G} = \mathbb{Z}_{k_1} \times \ldots \times \mathbb{Z}_{k_\ell}$ and integer vector $\mathbf{w} \in \mathbb{Z}^r$, $\{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \leftarrow \mathcal{U}(\mathbb{G}^r)\} = \mathcal{U}(d \cdot \mathbb{G})$ where $d = \gcd_{\mathbb{G}}(\mathbf{w})$. In particular,*

$$\Pr\left[\, \mathbf{w} \cdot \mathbf{g} = 0_{\mathbb{G}} \,\middle|\, \mathbf{g} \leftarrow \mathcal{U}(\mathbb{G}^r) \,\right] = 1/|d\mathbb{G}| = \prod_i \gcd(d, k_i)/k_i \,.$$

Proof. We want to analyze the probability distribution $\mathcal{W} = \{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \leftarrow \mathcal{U}(\mathbb{G}^r)\}$ for a fixed $\mathbf{w} \in \mathbb{Z}^r$ such that $\gcd_{\mathbb{G}}(\mathbf{w}) = d$. The function $\phi_{\mathbf{w}} \colon \mathbf{g} \mapsto \mathbf{w} \cdot \mathbf{g}$ maps $\mathbb{G}^r$ to $[\mathcal{W}] = \{\mathbf{w} \cdot \mathbf{g} \mid \mathbf{g} \in \mathbb{G}^r\} = d\mathbb{G}$. Let $\mathbb{G}_0 = \{\mathbf{g} \mid \mathbf{w} \cdot \mathbf{g} = 0_{\mathbb{G}}\}$ be the kernel of this function. Then, $\phi_{\mathbf{w}}$ partitions $\mathbb{G}^r$ into equivalence classes (cosets) of the form $\mathbf{g} + \mathbb{G}_0$. All cosets have the same size $|\mathbf{g} + \mathbb{G}_0| = |\mathbb{G}_0|$, and therefore $\phi_{\mathbf{w}}$ maps the uniform distribution over $\mathbb{G}^r$ to the uniform distribution over $\phi_{\mathbf{w}}(\mathbb{G}^r) = d\mathbb{G}$. This proves that $\mathcal{W} = \mathcal{U}(d\mathbb{G})$. The bound on $\Pr[\, \mathbf{w} \cdot \mathbf{g} = 0_{\mathbb{G}} \,]$ follows from (2.3). □

Knapsack Families. We consider generalized knapsack function families with domain the integer vectors $\mathbb{Z}^m$, input distributions $\mathcal{X}$ over $\mathbb{Z}^m$ and range any abelian group $\mathbb{G}$. More formally, a knapsack function family is defined as follows:

Definition 2.3.2 (Knapsack Function Family)**.** *For any $m \in \mathbb{N}$, group $\mathbb{G}$ and input distribution $\mathcal{X}$ over $\mathbb{Z}^m$, the* knapsack family $\mathsf{Knap}[\mathbb{G}, \mathcal{X}] = (F_{\mathsf{Knap}}, \mathcal{X})$ *is the function family with input distribution $\mathcal{X}$ and set of functions $F_{\mathsf{Knap}} = \{f_{\mathbf{g}} \colon [\mathcal{X}] \to \mathbb{G}\}_{\mathbf{g} \in \mathbb{G}^m}$ indexed by $\mathbf{g} \in \mathbb{G}^m$ and defined as $f_{\mathbf{g}}(\mathbf{x}) = \mathbf{g} \cdot \mathbf{x} \in \mathbb{G}$.*

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 F_{Knap}. We also often consider folded knapsack families $\mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ over quotient groups $\mathbb{G}_d$. For brevity, when $\mathbb{G}$ and $\mathcal{X}$ are clear from the context, we will simply write Knap (resp. Knap_d) instead of $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ (resp. $\mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$). The following lemma shows that the distribution $\mathcal{F}(\mathsf{Knap}_d)$ associated to a folded knapsack function family is closely related to the distribution

$$\mathcal{F}_d(\mathsf{Knap}) = \{(\mathbf{g}, g + h) \mid (\mathbf{g}, g) \leftarrow \mathcal{F}(\mathsf{Knap}), h \leftarrow \mathcal{U}(d\mathbb{G})\}. \tag{2.4}$$

Lemma 2.3.3. *For any knapsack family* Knap *and* $d \in \mathbb{Z}$, $\Delta_U(\mathcal{F}_d(\mathsf{Knap})) = \Delta_U(\mathcal{F}(\mathsf{Knap}_d))$. *Moreover,* $\mathcal{F}_d(\mathsf{Knap})$ *is pseudorandom if and only if* $\mathcal{F}(\mathsf{Knap}_d)$ *is pseudorandom.*

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

- Let $\mu : \mathbb{G}^m \times \mathbb{G} \to \mathbb{G}_d^m \times \mathbb{G}_d$ be the function $\mu(\mathbf{g}, g) = (\mathbf{g} \bmod d, g \bmod d)$. It is straightforward to verify that μ maps $\mathcal{U}(\mathbb{G}^m \times \mathbb{G})$ to $\mathcal{U}(\mathbb{G}_d^m \times \mathbb{G}_d)$ and $\mathcal{F}_d(\mathsf{Knap})$ to $\mathcal{F}(\mathsf{Knap}_d)$.

- In the other direction, let $\mu' \colon \mathbb{G}_d^m \times \mathbb{G}_d \to \mathbb{G}^m \times \mathbb{G}$ be the randomized transformation that on input $(\mathbf{h}, h)$ produces an output distributed according to $\{(\mathbf{h} + d \cdot \mathbf{g}, h + d \cdot g) \mid (\mathbf{g}, g) \leftarrow \mathcal{U}(\mathbb{G}^{m+1})\}$. Again, it is easy to see that μ' maps $\mathcal{U}(\mathbb{G}_d^m \times \mathbb{G}_d)$ to $\mathcal{U}(\mathbb{G}^m \times \mathbb{G})$ and $\mathcal{F}(\mathsf{Knap}_d)$ to $\mathcal{F}_d(\mathsf{Knap})$.

It follows that

$$\Delta_U(\mathcal{F}(\mathsf{Knap}_d)) = \Delta(\mu(\mathcal{F}_d(\mathsf{Knap})), \mu(\mathcal{U}(\mathbb{G}^m \times \mathbb{G}))) \leq \Delta_U(\mathcal{F}_d(\mathsf{Knap}))$$

and similarly

$$\Delta_U(\mathcal{F}_d(\mathsf{Knap})) = \Delta(\mu'(\mathcal{F}_d(\mathsf{Knap})), \mu'(\mathcal{U}(\mathbb{G}^m \times \mathbb{G})) \leq \Delta_U(\mathcal{F}(\mathsf{Knap}_d)) \ .$$

This proves $\Delta_U(\mathcal{F}_d(\mathsf{Knap})) = \Delta_U(\mathcal{F}(\mathsf{Knap}_d))$. Since the transformations μ and μ' are *efficiently* computable, they can also be used to turn any efficient distinguisher for $\mathcal{F}_d(\mathsf{Knap})$ into an efficient distinguisher for $\mathcal{F}(\mathsf{Knap}_d)$, and vice versa. □

We will need the following variant of the Leftover Hash Lemma [82], generalized to arbitrary abelian groups. The original Leftover Hash Lemma [82], 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 2.3.4. [Leftover Hash Lemma, variant] *For any knapsack function family* $\mathsf{Knap} = \mathsf{Knap}[\mathbb{H}, \mathcal{X}]$ *over a finite abelian group* $\mathbb{H}$*,*

$$\Delta_U(\mathcal{F}(\mathsf{Knap})) \leq \frac{1}{2}\sqrt{\sum_{1<d \mid \mathrm{M}_{\mathbb{H}}} |\mathbb{H}_d| \cdot \mathsf{Col}\left(\mathcal{X}_d\right)} \tag{2.5}$$

where $\mathcal{X}_d = \mathcal{X} \bmod d = \{\mathbf{x} \bmod d \mid \mathbf{x} \leftarrow \mathcal{X}\}$*, and* d *ranges over all divisors of the group exponent* $\mathrm{M}_{\mathbb{H}}$ *strictly greater than* 1 ($\mathrm{M}_{\mathbb{H}}$ *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 $\mathsf{Col}\left(\mathcal{Z}\right)$.

$$\begin{aligned}
\Delta_U(\mathcal{Z}) &= \frac{1}{2}\sum_{z\in Z}\left|\Pr\left[\, z \leftarrow \mathcal{Z}\,\right] - \frac{1}{|Z|}\right| \leq \frac{1}{2}\sqrt{|Z|}\sqrt{\sum_{z\in Z}\left(\Pr\left[\, z \leftarrow \mathcal{Z}\,\right] - \frac{1}{|Z|}\right)^2} \\
&= \frac{1}{2}\sqrt{|Z|}\sqrt{\sum_{z\in Z}\Pr\left[\, z \leftarrow \mathcal{Z}\,\right]^2 - \frac{2}{|Z|} + \frac{1}{|Z|}} \leq \frac{1}{2}\sqrt{|Z| \cdot \mathsf{Col}\left(\mathcal{Z}\right) - 1}. \qquad (2.6)
\end{aligned}$$

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

$$\begin{aligned}
\mathsf{Col}\left(\mathcal{F}(\mathsf{Knap})\right) &= \Pr\left[\, (\mathbf{h} = \mathbf{h}') \wedge (\mathbf{h} \cdot \mathbf{x} = \mathbf{h}' \cdot \mathbf{y})\,\right] \\
&= \Pr\left[\, \mathbf{h} = \mathbf{h}'\,\right] \cdot \Pr\left[\, \mathbf{h} \cdot (\mathbf{x} - \mathbf{y}) = 0\,\right] \\
&= \frac{1}{|\mathbb{H}|^m} \cdot \Pr\left[\, \mathbf{h} \cdot (\mathbf{x} - \mathbf{y}) = 0\,\right]. \qquad (2.7)
\end{aligned}$$

We finally compute $\Pr\left[\, \mathbf{h} \cdot (\mathbf{x} - \mathbf{y}) = 0\,\right]$ by conditioning on the value of $d = \gcd_{\mathbb{H}}(\mathbf{x} - \mathbf{y})$. Since $\gcd_{\mathbb{H}}(\mathbf{x} - \mathbf{y})$ divides $\mathrm{M}_{\mathbb{H}}$ (by definition), we can restrict d

to the divisors of $\mathrm{M}_{\mathbb{H}}$.

$$\begin{aligned}
\Pr\left[\,\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0\,\right] &= \sum_{d|\mathrm{M}_{\mathbb{H}}} \Pr\left[\,\mathbf{h}\cdot(\mathbf{x}-\mathbf{y})=0 \;\middle|\; \gcd_{\mathbb{H}}(\mathbf{x}-\mathbf{y})=d\,\right] \\
&\quad \cdot \Pr\left[\,\gcd_{\mathbb{H}}(\mathbf{x}-\mathbf{y})=d\,\right] \\
&\leq \sum_{d|\mathrm{M}_{\mathbb{H}}} \frac{1}{|d\mathbb{H}|}\cdot \mathsf{Col}\left(\mathcal{X}_d\right) \\
&= \frac{1}{|\mathbb{H}|} + \sum_{1<d|\mathrm{M}_{\mathbb{H}}} \frac{1}{|d\mathbb{H}|}\cdot \mathsf{Col}\left(\mathcal{X}_d\right)\ .
\end{aligned}$$

In the inequality above we used Lemma 2.3.1 and the fact that

$$\Pr\left[\,\gcd_{\mathbb{H}}(\mathbf{x}-\mathbf{y})=d\,\right] \leq \Pr\left[\,d\,|\,\mathbf{x}-\mathbf{y}\,\right] = \Pr\left[\,\mathbf{x} \bmod d = \mathbf{y} \bmod d\,\right] = \mathsf{Col}\left(\mathcal{X}_d\right)\ .$$

The bound in the lemma follows easily by combining (2.6), (2.7) and (2.8), and by using $|\mathbb{H}_d|\cdot|d\mathbb{H}| = |\mathbb{H}|$. □

2.4 Error Correcting Codes

We review some basic definitions and facts from coding theory. We restrict our attention only to the material that is required for the security proof of our CCA construction (Chapter 3) and refer the reader to [107] for a detailed treatment of the subject.

Let Σ be a set of symbols (alphabet) with $|\Sigma| = q$. For two strings $\mathbf{x}, \mathbf{y} \in \Sigma^w$, the *Hamming distance* $d_H(\mathbf{x}, \mathbf{y})$ is defined as the number of coordinates in which $\mathbf{x}$ differs from $\mathbf{y}$. Consider now a map (encoding) $\mathsf{ECC} : \Sigma^k \to \Sigma^w$. A *code* $\mathcal{C}$ is simply the image of such a map (that is $\mathcal{C} \subseteq \Sigma^w$), with $|\mathcal{C}| = q^k$. The *minimum distance* of a code $\mathcal{C}$ is defined as

$$d(\mathcal{C}) = \min_{\substack{\mathbf{x},\mathbf{y}\in\mathcal{C} \\ \mathbf{x}\neq\mathbf{y}}} \{d_h(\mathbf{x}, \mathbf{y})\}$$

We will use $[w, k, d]_q$ to denote a code $\mathcal{C}$ with *block length* w ($\mathcal{C} \subseteq \Sigma^w$), *information length* $k = \log_q |\mathcal{C}|$, minimum distance $d(\mathcal{C}) = d$ and alphabet size $|\Sigma| = q$.

For our CCA construction we need a code[2]. whose words are as "far apart" as possible. In particular, for a fixed k, we need a code which maximizes d/w

[2]For the purposes of the construction, we only need an appropriate encoding scheme and not a full -fledged error correcting code. (The ability to decode is unnecessary for the construction.)

under the restriction that w is polynomial in k. By the Singleton bound [150], $d \leq w - k + 1$ for any code and alphabet size which immediately gives an upper bound $1 - \frac{k-1}{w}$ for d/w. Codes that meet the Singleton bound are called *Maximum Distance Separable* (MDS) codes.

Reed-Solomon Codes. Reed-Solomon codes [138] are possibly the most famous example of MDS codes. We describe a (simplified) construction of a family of asymptotic Reed-Solomon codes. Let $RS^q_{w,k}$ denote a Reed-Solomon code (or more precisely a family of RS codes) with message length k, block length w and alphabet size $|\Sigma| = q$ (with $q \geq w$). The construction works as follows:

- *Generation:* Pick a field $\mathbb{F}_q$ (for convenience we use $\mathbb{Z}_q$ as the underlying field where q is the smallest prime such that $q \geq w$). Pick also w *distinct* elements $\alpha_1, ..., \alpha_w \in \mathbb{Z}_q$ (evaluation points).
- *Encoding:* Let $\mathbf{m} = (m_0, ..., m_{k-1}) \in \Sigma^k$ be a message and let $m(x) = \sum_{j=0}^{k-1} m_j x^j$ be the corresponding polynomial. The encoding of the message is defined as

  $$\mathsf{ECC}(\mathbf{m}) = \langle m(\alpha_1), ..., m(\alpha_w) \rangle \in \mathbb{Z}_q$$

  where the evaluation takes place over $\mathbb{Z}_q$.

The following lemma summarizes all the properties of the Reed-Solomon codes that are relevant for the application of interest.

Lemma 2.4.1. *The Reed-Solomon code $RS^q_{w,k}$ has minimum distance $d = w-k+1$. Also both the code length and the time complexity of the encoding are polynomial in w.*

2.5 Lattices and Gaussian Distributions

Gaussian-like distributions play a central role in the Learning With Errors (LWE) problem [140]. For each sample $(\mathbf{a}, b = \mathbf{a} \cdot \mathbf{s} + e)$, the error distribution χ used in sampling e, is typically a Gaussian-like distribution over the integers. Several (worst-case) lattice approximation problems can be reduced (under quantum or classic polynomial time reductions [139, 129]) to LWE with Gaussian error

distribution. Moreover, Gaussian noise is "LWE-complete" [141, 63] 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. Similar results hold for the *discretized* (or rounded) 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 understood to be the origin $\mathbf{c} = \mathbf{0}$. We will be primarily interested in discrete Gaussians over the set of integer vectors $S = \mathbb{Z}^m$. In that case, 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.,

$$\Pr[\, \mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,r} \,] = \prod_{i=1}^{m} \Pr[\, x_i \leftarrow \mathcal{D}_{\mathbb{Z},r} \,] = \prod_{i=1}^{m} \frac{\rho_r(x_i)}{\rho_r(\mathbb{Z})} \,. \tag{2.8}$$

Lattices. A (full-rank) m-dimensional *lattice* is the set Λ of integer linear 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 $\mathsf{det}(\Lambda)$) is the absolute value of the matrix determinant of

any basis $\mathbf{B}$ of Λ, i.e., $\mathsf{det}(\Lambda) = |\mathsf{det}(\mathbf{B})|$. The i-th *successive minimum* (in the ℓ_p norm) $\lambda_i^p(\Lambda)$ is the smallest radius r such that Λ contains i *linearly independent* vectors of ℓ_p-norm at most r. When the subscript p is omitted, we will always mean the ℓ_2 (Euclidean) norm. The *dual* of a lattice Λ is the set

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

For any $\epsilon \in \mathbb{R}^+$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ [117] is the smallest $r > 0$ such that $\rho_{1/r}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$. We will need the following bound on the smoothing parameter.

Proposition 2.5.1 ([63, Lemma 2.6])**.** *Let Λ be any lattice of dimension m. For any $\omega(\sqrt{\log m})$ function, there exists $\epsilon(m) = \mathsf{negl}(m)$ such that $\eta_\epsilon(\Lambda) \leq \omega(\sqrt{\log m})/\lambda_1^\infty(\Lambda^*)$.*

Random q-ary Lattices. Let $k, m, q \in \mathbb{N}$ be any positive integers (with $k < m$) and $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$. Each matrix $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$ gives rise to the following two (full-rank) m-dimensional integer lattices that are of particular interest in lattice-based cryptography:

$$\begin{aligned} \Lambda_q(\mathbf{A}) &= \{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{z} = \mathbf{A}^T\mathbf{s} \pmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^k\} \\ \Lambda_q^\perp(\mathbf{A}) &= \{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{z} = \mathbf{0} \pmod q\} \end{aligned}$$

The first lattice is precisely the linear (q-ary) code generated by the rows of $\mathbf{A}$ whereas the second lattice corresponds to the linear (q-ary) code with parity check matrix equal to $\mathbf{A}$. It is not hard to see that $q\mathbb{Z}^m \subseteq \Lambda_q(\mathbf{A}), \Lambda_q^\perp(\mathbf{A}) \subseteq \mathbb{Z}^m$ and that $\Lambda_q(\mathbf{A}), \Lambda_q^\perp(\mathbf{A})$ are dual up to a scaling factor, i.e., $\Lambda_q(\mathbf{A}) = q \cdot (\Lambda_q^\perp(\mathbf{A}))^*$ and $\Lambda_q^\perp(\mathbf{A}) = q \cdot (\Lambda_q(\mathbf{A}))^*$.

In order to establish the search-to-decision equivalence for LWE with discrete Gaussian error distribution, we need to bound the statistical distance between $(\mathbf{A}, \mathbf{A}\mathbf{e})$ and $\mathcal{U}(\mathbb{Z}_q^{k \times m} \times \mathbb{Z}_q^k)$ when $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{k \times m})$ and $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, r}$. Lemma 2.5.2 relates this statistical distance with the smoothing parameter of the lattice $\Lambda_q^\perp(\mathbf{A})$.

Lemma 2.5.2 ([63, Lemma 5.2])**.** *Let $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$ such that the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^k$. Then, for any $\epsilon \in (0, 1/2)$ and $r \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{A}))$, if $\mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, r}$, the syndrome $\mathbf{u} = \mathbf{A}\mathbf{e}$ is within statistical distance 2ϵ from uniform over $\mathbb{Z}_q^k$.*

2.6 Fourier Analysis

Fourier analysis and Gaussian distributions play an important role in basing the average-case hardness of LWE on worst-case lattice assumptions [139, 129]. In this thesis, 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.

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, 151] for more details.

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

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

where $\bar{z}$ is the complex conjugate of $z \in \mathbb{C}$. The ℓ_2-norm[3] and ℓ_∞-norm of h are defined as

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

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

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

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

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

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

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

FOURIER TRANSFORM. The *Fourier transform* of a function $h\colon \mathbb{H} \to \mathbb{C}$ is the function $\widehat{h}\colon \mathbb{H} \to \mathbb{C}$ defined as

$$\widehat{h}(\boldsymbol{\alpha}) = \langle h, \chi_{\boldsymbol{\alpha}} \rangle = \mathop{\mathbb{E}}_{x \leftarrow \mathcal{U}(\mathbb{H})} \left[h(x)\overline{\chi_{\boldsymbol{\alpha}}(x)} \right].$$

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

Learning Heavy Fourier Coefficients Let $\tau \in \mathbb{R}^+$, $\boldsymbol{\alpha} \in \mathbb{H}$ and $h\colon \mathbb{H} \to \mathbb{C}$ where $\mathbb{H}$ 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 $\text{Heavy}_\tau(h) = \{\boldsymbol{\alpha} \in \mathbb{H} \mid |\widehat{h}(\boldsymbol{\alpha})|^2 \geq \tau\}$. The following theorem states that it is possible to find $\text{Heavy}_\tau(h)$ given query access to h, and it is central in establishing the connection between the search and decision problems associated to bounded knapsack families.

Theorem 2.6.1. [Significant Fourier Transform,[3, Theorem 3.3]] *There exists a probabilistic algorithm (SFT) that on input a threshold $\tau \in \mathbb{R}^+$ and given query access to a function $h\colon \mathbb{H} \to \mathbb{C}$, returns all τ-heavy Fourier coefficients of h in time* $\mathsf{poly}(\log|\mathbb{H}|, 1/\tau, \|h\|_\infty)$ *with probability*[4] *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) $\sum_{\boldsymbol{\alpha} \in \mathbb{H}} |\widehat{h}(\boldsymbol{\alpha})|^2 = 1$. In this work, we are particularly interested in functions with very skewed Fourier spectrum, where a noticeable fraction of the total energy of the function is concentrated on a small number of coefficients. Namely, there exist characters $\boldsymbol{\beta} \in \mathbb{H}$ such that $|\widehat{h}(\boldsymbol{\beta})|^2 \geq \frac{1}{poly(\log|\mathbb{H}|)}$. In this context, Theorem 2.6.1 says that SFT, given query access to $h : \mathbb{H} \to \mathbb{T}$, can find all its $\frac{1}{poly(\log|\mathbb{H}|)}$-heavy Fourier coefficients in time polynomial in $\log|\mathbb{H}|$.

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

Chapter 3

Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions

Overview of the chapter. In this chapter we prove that Lossy Trapdoor Functions (LTDFs) with only minimal amount of lossiness imply the existence of CCA-secure encryption schemes. Section 3.1 summarizes our result and provides a non technical overview of the main ideas behind it. We review related work in Section 3.2. In Section 3.3, we recall the definition of security under correlated inputs, a notion introduced in [144] which plays an essential role in our result. We present the generalized CCA-secure scheme of Rosen and Segev [144] in Section 3.4, and show, as our main result, how to properly instantiate it in order to minimize the lossiness requirement on the underlying LTDFs. We conclude in Section 3.5 where we describe the construction of a new family of LTDFs with very small lossiness.

3.1 Results

The main result of this chapter is summarized in the following theorem.

Theorem 3.1.1. *CCA-secure schemes can be constructed in a black-box way from LTDFs*[1] *that lose* $\frac{1}{\mathsf{poly}(\lambda)}$ *bits.*

[1] Recall that a Lossy Trapdoor Function (LTDF) family is a standard injective trapdoor

A closer look at our proof. Here, we attempt to describe the core ideas from [132] and [144] that lead to our main result. We start by giving some intuition as to why LTDFs are sufficient for building a variety of cryptographic primitives. As a case study, consider CPA-secure encryption. For simplicity, let $\mathcal{F}$ be a family of LTDFs with inputs uniformly distributed over its domain $\{0,1\}^n$ that (computationally) loses 1 bit. Now consider a new family $\mathcal{F}'$ which is simply the w-wise product of $\mathcal{F}$ (for some polynomially bounded w) with inputs uniformly distributed over the domain $\{0,1\}^{nw}$. Sampling members of $\mathcal{F}'$ amounts to independently sampling w functions from $\mathcal{F}$. It is easy to see that $\mathcal{F}'$ computationally loses $w = \mathsf{poly}(\lambda)$ bits and thus (by applying the results of [132]) is one-way. Applying generic hardcore predicates, this immediately gives us a CPA-secure encryption scheme.

The Rosen-Segev encryption scheme [144] is similar, but with a fundamental difference: their construction revolves around the notion of one-wayness under *correlated inputs*, that is, the distributions they consider as inputs to members of $\mathcal{F}'$ are no longer independent (or uniform). Instead, they are correlated in such a way that any d (out of w) individual n-bit inputs are sufficient for reconstructing the remaining ones (we call such distributions (d, w)-subset reconstructible; see Section 3.3 for details.). This property is essential for their security proof, dictated by the need of the simulator to correctly answer decryption queries. Rosen and Segev focused on *highly* correlated distributions, where each of the w individual member functions of $\mathcal{F}$ that compose $\mathcal{F}'$ is applied to the same input (that is, $d = 1$). Unfortunately, for such highly correlated inputs, unless the members of $\mathcal{F}$ are *very* lossy, the w-wise product family leaks too much information about the input. That is the reason the CCA construction of [144] requires LTDFs that lose almost all their bits.

Our result directly relies upon the construction from [144]. However, we focus on distributions that are much less correlated. To that end, we first prove a straightforward theorem bounding the amount of lossiness required of an LTDF

function family with the following additional feature: each member function f from the family is computationally indistinguishable from the description $\hat{f}$ of another function that *statistically* loses information about its input.

in order to argue that its w-wise product is one-way with respect to a correlated input distribution $\mathcal{C}_w$ as a function of the min entropy μ of $\mathcal{C}_w$. We then show that, if we instantiate the error-correcting code in the Rosen-Segev construction with Reed-Solomon codes and carefully set all the parameters involved, then we can use a correlated input distribution $\mathcal{C}_w$ with sufficient min-entropy μ so that one-wayness for $\mathcal{C}_w$ is implied by LTDFs that only lose about 2 bits.

Constructing Slightly Lossy LTDFs. Another contribution of our work is the demonstration of a novel technique that allows the construction of LTDFs that lose a small amount of bits. Our technique can be briefly described as follows: We start with two non-injective trapdoor functions (say g and $\hat{g}$) that are computationally indistinguishable from each other. Both functions statistically lose information about their inputs but in different amounts (assume that $\hat{g}$ loses more bits of information). We then try to make g injective by appending to its evaluation on an input x, enough extra information about x (denoted by $h(x)$). The extra information is enough to make the pair (g, h) injective while $(\hat{g}, h)$ is still lossy. We use this technique to construct an LTDF from modular squaring that loses a fraction of one bit under the assumption that it is hard to distinguish the product of two primes from the product of three primes. This directly gives a CCA-secure encryption scheme from the latter assumption which might be of independent interest[2].

3.2 Related Work

Chosen-Ciphertext Security. Since the introduction of the notion in early nineties [124, 137], security against Chosen-Ciphertext Attacks (CCA) has been a central pursuit of cryptographers for over twenty years. Besides being theoretically intriguing, CCA-security is also motivated by practical attacks as first pointed out by Bleichenbacher [23] who demonstrated the feasibility of CCA attacks against schemes following the PKCS #1 encryption standard.

[2]It should be noted that this assumption is stronger than the quadratic residuosity assumption, from which we already know how to achieve CCA security (c.f. [43]).

Constructions of CCA-secure encryption schemes can be divided into 2 categories: On one hand, researchers have long focused on identifying the minimal *generic* assumptions that imply CCA-security. This line of work has led to a handful of distinct approaches for achieving CCA-security based on simpler and (possibly) more easily realizable primitives. The first approach, introduced by Naor and Yung [124], is based on the use of Non-Interactive Zero-Knowledge (NIZK) proofs for NP (the approach was further refined in [51, 146, 100]). A second approach, due to Cramer and Shoup [43], is based on the concept of universal hash proof systems. Hash proof systems and their extensions [159, 160] have proven a very useful notions both in the construction of CCA-secure schemes but also in other applications. Another approach, suggested by Canetti, Halevi and Katz [36], bases the construction of CCA-secure schemes on Identity Based Encryption, an interesting enhancement of standard public-key encryption in which the identities of the users serve as their public keys. Recently, several works have focused on generically building CCA-secure encryption from special types of Injective Trapdoor Functions (TDFs). This line of work has led to many interesting constructions including those from Lossy TDFs [132], TDFs secure under correlated inputs [144] and adaptive TDFs [92]. Finally, we mention a recently proposed alternative to building CCA-secure schemes based on the concept of Detectable Chosen Chiphertext Security (DCCA) [80]. DCCA is an intermediate security notion meant to capture schemes that, while not being directly CCA-secure, allow the detection of queries to the decryption oracle that can be useful for decrypting the challenge ciphertext.

In addition to generic assumptions, a fair amount of research has been concerned with the study of *concrete* and well-studied hardness assumptions that imply CCA security. As a result of this research, CCA-security is known be achievable based on the hardness of a wide range of computational problems such as Decisional Diffie Hellman (DDH) [42], (Bilinear) Computational Diffie Hellman (CDH, BCDH) [28, 38], Learning With Errors (LWE) [132, 129, 116], factoring [79], a variant of the Learning Parity with Noise (LPN) [52] and more.

Lossy Trapdoor Functions. Our work relies heavily on the notion of Lossy Trapdoor Functions (LTDFs) [132]. We conclude this section with a synopsis of

the existing literature on LTDFs pointing out, whenever applicable, connections to our work.

Since their introduction, LTDFs have been used (explicitly or implicitly) for the construction of a plethora of cryptographic primitives such as deterministic Public Key Encryption (PKE) schemes secure in the standard model [27], Injective TDFs secure under correlated inputs [144], PKE schemes secure under selective-opening attacks [16], Hedged PKE schemes [15], Leakage-Resilient Signatures [29], Lossy Encryption [75] and more.

Also, motivated by the work of [132], numerous constructions of LTDFs have been proposed. Rosen and Segev [143] and Boldyreva, Fehr, and O'Neill [27] both gave a construction based on the decisional composite residuosity (DCR) assumption, while Kiltz, O'Neill and Smith [93] recently showed that the RSA trapdoor permutation is lossy under the Φ-hiding assumption of [35]. Finally, Freeman *et al* [60] proposed an LTDF that loses one bit under the Quadratic Residuosity assumption (QR). While the DCR-based LTDF is sufficient to directly instantiate the CCA-secure scheme of [132], the LTDFs based on Φ-hiding and QR are not sufficiently lossy and thus cannot be used directly in the generic construction. In light of that, our result, combined with [93] and [60], shows that CCA-security is implied by both the Φ-hiding and the Quadratic Residuosity assumptions.

3.3 Products and Correlated Inputs

In this section we define w-wise products, prove the lossiness amplification lemma that we use throughout the chapter, and finally present the types of correlated input distributions we are interested in for our CCA result.

Products and Lossiness Amplification

We first define the w-wise product of a collection of functions and then show how such a product can amplify lossiness.

Definition 3.3.1 (w-wise product, Definition 3.1 in [144])**.** *Let $\mathcal{F} = (G, F)$ be a collection of efficiently computable functions where G, F are the function generation*

and evaluation algorithm respectively. For any integer w, we define the w-wise product $\mathcal{F}_w = (G_w, F_w)$ as follows:

- *The generation algorithm G_w on input 1^λ invokes $G(1^\lambda)$ for w times independently and outputs $(s_1, \ldots, s_w)$. That is, a function is sampled from $\mathcal{F}_w$ by independently sampling w functions from $\mathcal{F}$.*

- *The evaluation algorithm F_w on input $(s_1, \ldots, s_w, x_1, \ldots, x_w)$ invokes F to evaluate each function s_i on x_i. That is,*

$$F_w(s_1, \ldots, s_w, x_1, \ldots, x_w) = (F(s_1, x_1), \ldots, F(s_w, x_w)).$$

The following lemma states that w-wise products amplify the *absolute amount* of lossiness[3].

Lemma 3.3.2 (Lossiness Amplification)**.** *Let λ be a security parameter. For any family of TDFs $\mathcal{F} = (G, F, F^{-1})$ with message space $n(\lambda)$, if $\mathcal{F}$ is $(n(\lambda), \ell(\lambda))$-lossy, then the $w(\cdot)$-wise product family $\mathcal{F}_w$ (defined above) built from $\mathcal{F}$ is $(n(\lambda) \cdot w(\lambda), \ell(\lambda) \cdot w(\lambda))$-lossy.*

Proof. First, if there exists an efficient lossy key generation algorithm $\hat{G}$ that outputs indistinguishable function indices from G, then by a straightforward hybrid argument it follows that $\hat{G}_w$, the algorithm that runs $\hat{G}$ independently w times to get $(s_1, t_1), \ldots, (s_w, t_w)$ and outputs $(\mathbf{s}, \mathbf{t})$ where $\mathbf{s} = (s_1, \ldots, s_w)$ and $\mathbf{t} = (t_1, \ldots, t_w)$, outputs indistinguishable keys from G_w.

Second, since for each s_i output by $\hat{G}$ the range of $F(s_i, \cdot)$ has size at most $2^{n-\ell}$, it follows that for each $\mathbf{s}$ output by $\hat{G}_w$, the range of $F_w(\mathbf{s}, \cdot)$ has size at most $(2^{n-\ell})^w = 2^{nw-\ell w}$ which concludes the proof. □

As an immediate corollary of Lemma 3.3.2, we get that $(n, \frac{1}{poly(\lambda)})$-LTDFs imply injective trapdoor one-way functions and CPA-secure encryptions (the proofs of these statements are a rather straightforward combination of Lemma 3.3.2 and the results from [132] and are hence omitted). We simply state this observation as a corollary for completeness.

[3]We use the term "absolute amount of lossiness" to explicitly distinguish it from "rate of lossiness" (*aka* relative lossiness) defined as $\frac{k}{n}$ for a (n, k)-LTDF. It has been known [134], that amplifying the rate of lossiness in a black-box way is impossible beyond a certain threshold.

Corollary 3.3.3. *Let $p(\cdot)$ be any polynomial. Then $(n, \frac{1}{p(\lambda)})$-LTDFs imply injective trapdoor one-way functions and CPA-secure encryption schemes.*

Subset Reconstructible Distributions

While it is well-known that if $\mathcal{F}$ is one-way with respect to the uniform distribution on $\{0,1\}^n$, then the product $\mathcal{F}_w$ is one-way with respect to the uniform distribution over $\{0,1\}^{nw}$, we will be interested in the security of products when the inputs are *correlated* and not necessarily uniform. Of particular interest, are input distributions that are what we call (d,w)-*subset reconstructible*.

Definition 3.3.4 ((d,w)- Subset Reconstructible Distribution (SRD))**.** *Let $d, w \in \mathbb{N}$ such that $d \leq w$, $\mathcal{S}$ be a domain and $\mathcal{D}$ a distribution with support $[\mathcal{D}] \subseteq \mathcal{S}^w$. We say that $\mathcal{D}$ is (d,w)- Subset Reconstructible (and denote $\mathcal{S}RD_{d,w}$) if, each w-tuple $(x_1, ..., x_w) \in [\mathcal{D}]$ is fully and uniquely reconstructible from any subset $\{x_{i_1}, ..., x_{i_d}\}$ of d distinct elements of the tuple.*

It is easy to see that the special case where $d = 1$ and $\mathcal{S} = \{0,1\}^n$ is precisely the uniform w-repetition distribution used in the simplified construction of the CCA secure cryptosystems in [144]. For the CPA construction, choosing $d = w$ would suffice, in which case the distribution contains all w-tuples $(x_1, ..., x_w)$ where each x_i is chosen independently and uniformly at random from $\{0,1\}^n$. For the CCA-construction, however, we need to choose a value for d smaller than w (this is necessary for almost perfect simulation of the decryption oracle) but as close to w as possible in order to minimize the required lossiness of the TDF (the closer to 1 the value $\frac{d}{m}$ is, the less lossiness we need for the CCA construction).

Before describing how to sample efficiently from $\mathcal{S}RD_{d,w}$ we note the similarity of the above definition with two well studied notions from Coding Theory and Cryptography, namely *erasure codes* and *secret sharing schemes*. Even though our sampling algorithm for $\mathcal{S}RD_{d,w}$ uses techniques identical to those used in the construction of the most popular erasure codes and secret sharing schemes, we introduce this new definition here since, in principle, the goals (properties) of the two aforementioned notions are slightly different from those of a (d,w)-subset recon-

structible distribution. In particular, the goal of an erasure code is to recover the initial message and not necessarily the full codeword (even though the full codeword can trivially be constructed by re-encoding the recovered initial message) when at most $w - d$ symbols of the codeword have been lost during transmission. Likewise, in a (d, w)-threshold secret sharing scheme the goal is to recover a secret s when any d out of w distinct values are known (again here there is no requirement to recover all w values from the d known ones).

Sampling via Polynomial Interpolation. We use polynomial interpolation as a way to sample efficiently from $\mathcal{SRD}_{d,w}$ for any value of d and w. The construction is identical to the one used by Shamir [148] for a (d, w)-threshold secret sharing scheme. On input a prime q (with $\log q = O(\mathsf{poly}(\lambda))$) and integers d, w, the sampling algorithm picks independently d values $p_0, ..., p_{d-1}$ uniformly at random from $\mathbb{Z}_q$ (these correspond to the d coefficients of a $(d-1)$-degree polynomial $p \in \mathbb{Z}_q[x]$). The algorithm then simply outputs $(x_1, ..., x_w) = (p(1), ..., p(w))$ where evaluation takes place in $\mathbb{Z}_q$ and x_i's are represented by binary strings of length at most $\log q$. [4] The following lemma states that the output distribution of polynomial interpolation sampling is a (d, w)- subset reconstructible distribution with sufficient entropy.

Lemma 3.3.5. *Let $w = \mathsf{poly}(\lambda)$. Then the above algorithm is a $\mathsf{poly}(\lambda)$-sampling algorithm for $\mathcal{SRD}_{d,w}$. Also the min-entropy of the distribution $\mathcal{SRD}_{d,w}$ is $d \cdot \log q$.*

Proof. First notice that any distinct d values $(x_{i_1} = p(i_1), ..., x_{i_d} = p(i_d))$ (with $i_j \in [w]\ \forall j = 1, ..., d$) uniquely determine the polynomial p and hence the whole tuple $(x_1, ..., x_w)$. Also for any set $S = \{i_1, ..., i_d\} \subseteq [w]$ of distinct indices and any $\mathbf{y} = (y_1, ..., y_d) \in \mathbb{Z}_q^d$

$$\Pr\left[\, x_{i_1} = y_1 \wedge ... \wedge x_{i_1 d} = y_d \,\right] = \Pr\left[\, \mathbf{V}_{i_1,...,i_d}\mathbf{p} = \mathbf{y} \,\right] = \Pr\left[\, \mathbf{p} = \mathbf{V}^{-1}_{i_1,...,i_d}\mathbf{y} \,\right] = \frac{1}{q^d}$$

where $\mathbf{p}$ corresponds to the vector $[p_0, ..., p_{d-1}]^T$ and $\mathbf{V}_{i_1,...,i_d}$ is the (invertible) Vandermonde matrix with j-th row $[x_{i_j}^0, ..., x_{i_j}^{d-1}]$ It follows that $\mathrm{H}_\infty((x_1, ..., x_w)) = d \cdot \log q$. □

[4] Any (fixed and public) distinct values $a_1, ..., a_w \in \mathbb{Z}_q$ instead of $1, ..., w$ would work just fine.

3.4 CCA Secure Encryption from Functions with Small Lossiness

In this section we prove our main result: lossy TDFs that lose only a noticeable fraction of a single bit imply CCA-secure encryption. We start by describing the encryption scheme of Rosen and Segev [144] that shows that CCA security is implied by the security (one-wayness) of trapdoor injective functions under certain correlated products. We then show that $(n, 2)$-lossy TDFs imply injective trapdoor functions that are secure under these correlated products. We complete the proof by observing that, by a straightforward lossiness amplification argument, $(n, 2)$-lossy TDFs can be constructed in a black-box way from LTDFs that lose a $\frac{1}{poly(\lambda)}$ fraction of a single bit.

For ease of presentation, we describe a *single-bit* encryption scheme. Due to a recent result [123], this directly implies the existence of multi-bit CCA-secure schemes. We mention however that one can get a multi-bit encryption scheme directly by simply replacing the hardcore predicate h with a universal hash function, as in the PKE schemes of [132].

3.4.1 The Rosen-Segev Construction

We recall the cryptosystem from [144]. The main components of the cryptosystem are: a collection $\mathcal{F} = (G, F, F^{-1})$ of injective trapdoor functions, a (hardcore) predicate $h : \{0,1\}^* \to \{0,1\}$, an efficiently computable encoding function $\mathsf{ECC} : \Sigma^k \to \Sigma^w$ for an error-correcting code with distance d and an one-time signature scheme $\Pi = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Ver})$ whose verification keys are elements in Σ^k. (We could always use a universal hash function to hash keys into this space.) Finally, $\mathcal{C}_w(1^\lambda)$ is an input distribution such that any $\mathbf{x} = (x_1, \ldots, x_w)$ output by $\mathcal{C}_w(1^\lambda)$ can be efficiently reconstructed given any size $d < w$ subset of $\mathbf{x}$. The Rosen-Segev encryption scheme works as follows:

Key Generation: On input security parameter 1^λ, for each $\sigma \in \Sigma$ and each $1 \leq i \leq w$, run $(s_i^\sigma, t_i^\sigma) \xleftarrow{\$} G(1^\lambda)$, the key generation for the injective trapdoor function family. Return the pair (pk, sk) where

$$\begin{aligned} \mathsf{pk} &= (\{s_1^\sigma\}_{\sigma\in\Sigma}, \ldots, \{s_w^\sigma\}_{\sigma\in\Sigma}) \\ \mathsf{sk} &= (\{t_1^\sigma\}_{\sigma\in\Sigma}, \ldots, \{t_w^\sigma\}_{\sigma\in\Sigma}) \end{aligned}$$

Encryption : On input public key pk and one-bit message m, run $\mathsf{Kg}(1^\lambda)$ to generate $(\mathsf{VK}, \mathsf{SK})$ and sample $(x_1, \ldots, x_w)$ from $\mathcal{C}_w(1^\lambda)$. Apply the error correcting code to VK to get $\mathsf{ECC}(\mathsf{VK}) = (\sigma_1, \ldots, \sigma_w)$. The output is $c = (\mathsf{VK}, y_1, \ldots, y_w, c_1, c_2)$ where VK is as above and

$$\begin{aligned} y_i &= F(s_i^{\sigma_i}, x_i),\ 1 \le i \le w \\ c_1 &= m \oplus h(s_1^{\sigma_1}, \ldots, s_w^{\sigma_w}, x_1, \ldots, x_w) \\ c_2 &= \mathsf{Sign}(\mathsf{SK}, (y_1, \ldots, y_w, c_1)) \ . \end{aligned}$$

Decryption: On input secret key sk and ciphertext $c = (\mathsf{VK}, y_1, \ldots, y_w, c_1, c_2)$ check if $\mathsf{Ver}(\mathsf{VK}, (y_1, \ldots, y_w, c_1), c_2)$ equals 1. If not output $\perp$. Otherwise, compute $\mathsf{ECC}(\mathsf{VK}) = (\sigma_1, \ldots, \sigma_w)$ and pick d distinct indices $i_1, ..., i_d$. Use the trapdoors $t_{i_1}^{\sigma_{i_1}}, ..., t_{i_d}^{\sigma_{i_d}}$ to compute $x_{i_1} = F^{-1}(t_{i_1}^{\sigma_{i_1}}, y_{i_1}), \ldots, x_{i_d} = F^{-1}(t_{i_d}^{\sigma_{i_d}}, y_{i_d})$. Use these x_i's to reconstruct the entire vector $x_1, \ldots, x_w$. If $y_j = F(s_j^{\sigma_j}, x_j)$ for all $1 \le j \le w$ output $c_1 \oplus h(s_1^{\sigma_1}, \ldots, s_w^{\sigma_w}, x_1, \ldots, x_w)$. Else output $\perp$.

Rosen and Segev then proved the following theorem:

Theorem 3.4.1 (Theorem 5.1 in [142])**.** *If Π is a one-time strongly unforgeable signature scheme, $\mathcal{F}$ is one-way under a $\mathcal{C}_w$-correlated product, and h is a hardcore predicate for $\mathcal{F}_w$ with respect to $\mathcal{C}_w$, then the above PKE scheme is IND-CCA secure.*

3.4.2 Our Result

In this section we establish the following result.

Theorem 3.4.2 (Main Theorem)**.** *CCA-secure schemes can be constructed in a black-box way from LTDFs that lose $\frac{1}{\mathsf{poly}(\lambda)}$ bits.*

The proof proceeds in two steps. In the first step (Lemma 3.4.3), we show that lossy TDFs give rise to families of injective trapdoor functions that are secure

under correlated product distributions with sufficiently large entropy. Moreover, the more entropy the underlying distribution has, the less lossiness is required from the LTDFs. In the second and final step (Lemma 3.4.5), we show that, by choosing the appropriate error correcting code and a correlated input distribution with high entropy in the Rosen-Segev scheme, we can achieve one-wayness under correlated products (and hence CCA-security) starting from lossy TDFs with minimal lossiness requirements. More specifically, using the uniform $\mathcal{SRD}_{d,w}$ (which has high entropy, see Lemma 3.3.5) as our underlying distribution and Reed-Solomon codes for ECC, we show that $(n, 2)$-lossy TDFs suffice for CCA-secure encryption. We finally derive Theorem 4.3.1 by observing that $(n, 2)$-lossy TDFs can be constructed by $(n', \frac{1}{\mathsf{poly}(\lambda)})$-lossy functions (where $n = \mathsf{poly}(n')$) (see Lemma 3.3.2).

The following lemma provides an explicit bound on the required lossiness as a function of the *entropy* of the correlated input distribution.

Lemma 3.4.3. *Let $\mathcal{F} = (G, F, F^{-1})$ be a collection of (n, ℓ)-lossy trapdoor functions and let $\mathcal{F}_w = (G_w, F_w)$ be its w-wise product for $w = \mathsf{poly}(\lambda)$. Let $\mathcal{C}_w$ be an input distribution with min-entropy μ. Then $\mathcal{F}$ is one-way under a $\mathcal{C}_w$-correlated product provided that*

$$\ell \geq n - \frac{\mu}{w} + \frac{\omega(\log \lambda)}{w}. \tag{3.1}$$

Proof. The proof is similar with a proof from [132]. Assume for the sake of contradiction that there is an inverter $\mathcal{I}$ that succeeds at inverting $\mathcal{F}_w$ with probability $1/p(\lambda)$ for some polynomial p. We will show how to build a distinguisher $\mathcal{D}$ that can distinguish between the lossy keys and real keys. Because of a standard hybrid argument, it suffices to show that $\mathcal{D}$ can distinguish (with non-negligible advantage) $w = \mathsf{poly}(\lambda)$ lossy keys (generated with $\hat{G}$) from $w = \mathsf{poly}(\lambda)$ real keys (generated with G). $\mathcal{D}$ works as follows: on input keys $\mathbf{s} = (s_1, \ldots, s_w)$, it samples $\mathbf{x} = (x_1, \ldots, x_w)$ from $\mathcal{C}_w(1^\lambda)$ and runs the inverter $\mathcal{I}(1^\lambda, \mathbf{s}, F_w(\mathbf{s}, \mathbf{x}))$. If $\mathbf{s}$ are real, i.e., they are generated using G, then $\mathcal{I}$ will output $\mathbf{x}$ with non-negligible probability. If, however, $\mathbf{s}$ come from $\hat{G}$, then the probability of success for $\mathcal{I}$ is at most $2^{-\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F_w(\mathbf{s}, \mathbf{x})))}$. But, by Lemma 2.1.1,

$$\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F(\mathbf{s}, \mathbf{x}))) \geq \mathrm{H}_\infty(\mathbf{x} \mid \mathbf{s}) - w(n - \ell)\ . \tag{3.2}$$

Since the choice of the functions is independent from the choices of $\mathbf{x}$, the first term on the right hand side of the above equation is simply $\mathrm{H}_\infty(\mathbf{x}) = \mu$. Combining with (3.2), we get that

$$\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid (\mathbf{s}, F(\mathbf{s}, \mathbf{x}))) \geq \mu - w(n - \ell) \geq \omega(\log \lambda)$$

where in the last inequality we used the hypothesis for ℓ. It follows that the probability $\mathcal{I}$ succeeds when $\mathcal{D}$ is given lossy keys is upper bounded by $2^{-\omega(\log \lambda)} = \mathsf{negl}(\lambda)$. Therefore, $\mathcal{D}$ can distinguish between keys from G and keys from $\hat{G}$ which contradicts the fact that $\mathcal{F}$ is a collection of lossy functions and concludes the proof. □

Remark 3.4.4. *If $\mathcal{C}_w$ is the uniform w-repetition distribution (which has entropy n), then the bound from Lemma 3.4.3 matches exactly the one from [144, Theorem 3.3]. If in addition $w = 1$, then Lemma 3.4.3 is a restatement of the fact that any $(n, \omega(\log n))$-LTDFs is one-way [132].*

The following lemma shows that by appropriately instantiating the error correcting code ECC and the correlated inputs distribution C_w in the Rosen-Segev scheme, we can construct a CCA-secure scheme directly from any $(n, 2)$-lossy function.

Lemma 3.4.5. *CCA-secure schemes can be constructed in a black-box way from $(n, 2)$-lossy TDFs.*

Proof. Let $n = \mathsf{poly}(\lambda)$. Let also $\mathsf{ECC} \in RS^q_{w,k}$ be a Reed-Solomon code with $k = n^\epsilon$ (for some constant ϵ with $0 < \epsilon < 1$) , $w = n^c$ for some constant $c > 1 + \epsilon$, q the smallest prime such that $q \geq w$ and distance $d = w - k + 1$. Let also C_w be the distribution $\mathcal{S}RD_{d,w}$ sampled via polynomial interpolation (see Section 3.3) for some prime p such that $n - 1 \leq \log p \leq n$. Let finally $\mathcal{F} = (G, F, F^{-1})$ be a collection of $(n, 2)$-lossy trapdoor functions and $\mathcal{F}_w = (G_w, F_w)$ be its w-wise product. By construction (Lemma 3.3.5, Section 3.3) C_w has min-entropy $\mu = \mathrm{H}_\infty(C_w) = d \cdot \log p$ and can be sampled in time $\mathsf{poly}(w) = \mathsf{poly}(\lambda)$. In addition, by properties of the Reed-Solomon codes we have

$$\frac{d}{w} = \frac{w - k + 1}{w} \geq 1 - \frac{k}{w} = 1 - \frac{1}{n^{c-\epsilon}}$$

and hence

$$\frac{\mu}{w} = \frac{d}{w}\log p \geq (n-1)\cdot\left(1-\frac{1}{n^{c-\epsilon}}\right) = n-1-\frac{1}{n^{c-\epsilon-1}}+\frac{1}{n^{c-\epsilon}}\ .$$

As a result,

$$\begin{aligned} n-\frac{\mu}{w}+\frac{\omega(\log\lambda)}{w} &\leq n-\left(n-1-\frac{1}{n^{c-\epsilon-1}}+\frac{1}{n^{c-\epsilon}}\right)+\frac{\omega(\log\lambda)}{w} \\ &= 1+\frac{1}{n^{c-\epsilon-1}}-\frac{1}{n^{c-\epsilon}}+\frac{\omega(\log\lambda)}{n^c} \\ &< 2 \end{aligned}$$

for some $\omega(\log\lambda)$- function. Applying Lemma 3.4.3, we get that $\mathcal{F}$ is secure under the aforementioned C_w-correlated product. Let h be a hardcore predicate for the w-wise product $\mathcal{F}_w$ (with respect to C_w). Applying the construction of Rosen and Segev from Section 3.4.1 and Theorem 3.4.1 we get that $(n,2)$-lossy functions imply CCA-security (in a black-box sense). □

Possible Optimizations and Trade-offs. An obvious drawback of the above construction is that the alphabet size of the ECC increases with the security parameter λ. This leads to an increase in the sizes of the public and secret key by a factor of at least $n = \mathsf{poly}(\lambda)$ (where n is the length of the domain of $\mathcal{F}$). By replacing the Reed-Solomon code with a random linear code (RLC) with alphabet $\{0,1\}$ we can improve the space (and time) efficiency of the scheme. However, in this case, asymptotically we can only achieve relative distance $\delta' = \frac{d}{w} < \frac{1}{2}$. This translates to a much stronger lossiness requirement for the underlying family of lossy functions. In particular, we need to start with (n,l)-lossy functions where $l \geq n(1-\delta)$ for some $\delta < \frac{1}{2}$ (that is, the lossy function should lose more than half of the input bits).

Another possible optimization is to generalize the construction to encryption schemes that support longer (than a single bit) messages. For messages of length v, a universal hash function $h : \{0,1\}^{nw} \to \{0,1\}^v$ can be used instead of a hardcore predicate to pad the message. It can be proven that, by sampling the correlated inputs as above and applying the generalized leftover hash lemma, the distribution $h(x_1,...,x_w)$ (which is used as the padding of the message) is statistically close to the uniform over $\{0,1\}^v$. In this case the requirement for the lossiness

becomes

$$\ell \geq n - \frac{\mu}{w} + \frac{\omega(\log \lambda)}{w} + \frac{v}{w} \ .$$

As before, by picking all the parameters appropriately, we can instantiate the multi-bit construction using as a building block $(n, \frac{1}{poly(\lambda)})$-LTDFs.

3.5 A Slightly Lossy TDF from the 2v3Primes Assumption

In this section we construct a family of LTDFs that lose 1/4 bits of their input. We start by presenting the main idea of the construction, then define the underlying hardness assumption and conclude with the detailed construction along with a proof that the construction yields a $(n, 1/4)$-LTDF.

The Idea. Our technique generalizes previous approaches for constructing LTDFs and might serve as a paradigm for the construction of LTDFs from other hardness assumptions. Let g be a trapdoor function (with trapdoor t) that loses ℓ bits (where $\ell \geq 0$, and $\ell = 0$ corresponds to an injective trapdoor function). Let also $\hat{g}$ be a deterministic function such that $\hat{g} \simeq_c g$ (under some computational assumption $\mathcal{C}A$) and $\hat{g}$ loses $\hat{\ell}$ bits (that is $|\mathsf{Img}(\mathsf{Dom}(\hat{g}))| \leq \frac{|\mathsf{Dom}(\hat{g})|}{2^{\hat{\ell}}}$ for some $\hat{\ell} > \ell$). Consider now a function h such that $\mathsf{len}(h(x)) = \ell$ where len denotes length (bitsize) and $(g(x), h(x))$ uniquely determine the preimage x (which can be efficiently recovered given the trapdoor t) for all inputs x. Define $s = (g, h)$ and $\hat{s} = (\hat{g}, h)$. Then it is clear that s is a description of an injective trapdoor function whereas $\hat{s}$ corresponds to an $(\hat{\ell} - \ell)$-lossy function. Indeed $|\mathsf{Img}(\hat{s})| \leq |\mathsf{Img}(\mathsf{Dom}(\hat{g}))| \cdot 2^{\ell} \leq \frac{\mathsf{Dom}(\hat{g})}{2^{\hat{\ell}-\ell}}$. Finally the indistinguishability of $\hat{g}$ and g implies that $s \simeq_c \hat{s}$.

Hardness assumption. Our hardness assumption is similar to the 2OR3A assumption introduced by Blum *et al* [26] (in a slightly different form). It roughly says that it is hard to distinguish between products of two and products of three primes. In more detail, let $n = \mathsf{poly}(\lambda)$ where λ is the security parameter. Consider

the following two distributions[5]

$$\mathsf{2Primes}_n = \{N = pq \;\big|\; \mathsf{len}(N) = n;\ p, q \text{ primes such that } p \equiv_4 q \equiv_4 3\} \tag{3.3}$$

$$\mathsf{3Primes}_n = \{N = pqr \;\big|\; \mathsf{len}(N) = n;\ p, q, r \text{ primes such that } pqr \equiv_4 1\} \tag{3.4}$$

where all primes are distinct and $\mathsf{len}(N) = n$ implies that the most significant bit of N is 1.

Assumption 3.5.1 (2v3Primes)**.** *For any PPT algorithm $\mathcal{D}$ and any polynomial $p(\cdot)$*

$$\left| \Pr[\,\mathcal{D}(\mathsf{2Primes}_n) = 1\,] - \Pr[\,\mathcal{D}(\mathsf{3Primes}_n) = 1\,] \right| \leq \frac{1}{p(n)}$$

where the probability is taken over the randomness of sampling N and the internal randomness of $\mathcal{D}$.

The Construction. For our function g we use squaring modulo the product N of two large primes p and q. This function was the basis for the Rabin cryptosystem [136]. Let $n = \mathsf{poly}(\lambda)$. We define a family of injective trapdoor functions $\mathcal{F} = (G, F, F^{-1})$ as shown in Figure 3.1.

Note that even though the modulus N has bitsize $n + 1$ (that is $N > 2^n$) the domain of F is $\{0, 1\}^n$. For the proof we will need the following two standard lemmas which we prove for completeness.

Lemma 3.5.2. *Let $N = \prod_{i=1}^{k} p_i$ be a product of k distinct primes. Then the function $f(x) = x^2 \bmod N$ defined over $\mathbb{Z}_N^*$ is 2^k-to-1.*

Proof. Consider the isomorphism $\rho : \mathbb{Z}_N^* \leftrightarrow \mathbb{Z}_{p_1}^* \times \cdots \times \mathbb{Z}_{p_k}^*$ defined as

$$\rho(x) = (x_{p_1}, ..., x_{p_k}) \quad \text{where } x_{p_i} = x \bmod p_i$$

Let $z = (z_{p_1}, ..., z_{p_k}) \in \mathcal{QR}_N$ be an element of the image of f. Let $x = (x_{p_1}, ..., x_{p_k}) \in \mathbb{Z}_N^*$ such that $x^2 \equiv_N z$. It is not hard to see that the 2^k numbers x' of the form $x' = (\pm x_{p_1}, ..., \pm x_{p_k})$ are all distinct and such that $x'^2 \equiv_N x^2$.

[5]The requirement $pqr \equiv_4 1$ for the 3Primes distribution is essential since otherwise there exists a trivial algorithm that distinguishes between Ns sampled according to G and those sampled according to $\hat{G}$.

$\underline{G(1^\lambda)}$: Choose two large primes p, q such that $p \equiv q \equiv 3 \pmod 4$ and pq has bitsize $n+1$. Let $N = pq$. That is $N \xleftarrow{\$} \mathsf{2Primes}_{n+1}$. Return (s, t) where $s = N$ and $t = (p, q)$.

$\underline{\hat{G}(1^\lambda)}$: Choose three large (balanced) primes p, q and r such that $pqr \equiv_4 1$ and pqr has bitsize $n+1$. Let $N = pqr$, that is $N \xleftarrow{\$} \mathsf{3Primes}_{n+1}$. Return $(s, \perp)$ where $s = N$.

$\underline{F(s, x)}$: Parse s as N. On input $x \in \{0,1\}^n$, interpret x as an integer in $[0, 2^n - 1]$ and compute $y = x^2 \mod N$. Let $\mathcal{P}_N(x) = 1$ if $x > N/2$ and $\mathcal{P}_N(x) = 0$ otherwise. Let also $\mathcal{Q}_N(x) = 1$ if $\mathcal{J}_N(x) = 1$ and $\mathcal{Q}_N(x) = 0$ otherwise where $\mathcal{J}_N(x)$ is the Jacobi symbol of x modulo N. Return $(y, \mathcal{P}_N(x), \mathcal{Q}_N(x))$.

$\underline{F^{-1}(t, y')}$: Parse t as (p, q) and y' as (y, b_1, b_2). Compute the square roots $x_1, ..., x_k$ of y using p and q (the number of square roots is bounded by Lemma 3.5.2). Compute $\mathcal{P}_N(x_i)$ and $\mathcal{Q}_N(x_i)$ for all $i \in [k]$ and output the x_i such that $\mathcal{P}_N(x_i) = b_1$ and $\mathcal{Q}_N(x_i) = b_2$ (Lemma 3.5.3 says that there exists a unique x_i that is consistent with both b_1 and b_2).

Figure 3.1: A family of $(n, 1/4)$-LTDF based on the hardness of the 2v3Primes assumption.

Conversly, let $y = (y_{p_1}, ..., y_{p_k})$ be such that $y^2 \equiv_N z$. Then it should be the case that $y_{p_i}^2 \equiv_{p_i} z_{p_i}$ for all $i \in [k]$. However since p_i's are all primes, each z_{p_i} has exactly two square roots modulo p_i, namely $\pm x_{p_i}$. That means that the square roots of z modulo N are exactly those that have the form $(\pm x_{p_1}, ..., \pm x_{p_k})$ which conludes the proof. □

The following lemma is needed to prove that the family $\mathcal{F} = (G, F, F^{-1})$ is injective.

Lemma 3.5.3. *Let $N = pq$ where p, q are primes such that $p \equiv q \equiv 3 \pmod N$. Let also $x, y \in \mathbb{Z}_N^*$ such that $x \neq \pm y$ and $x^2 \equiv y^2 \equiv z \pmod N$. Then $\mathcal{J}_N(x) =$*

$-\mathcal{J}_N(y)$.

Proof. Let $z = (z_p, z_q)$ where $z_p = z \bmod p$ and $z_q = z \bmod q$. Since $z \in \mathcal{QR}_N$ there exists an element $x \in \mathbb{Z}_N^*$ such that $x^2 \equiv_N z$. Let $x = (x_p, x_q)$. Then (see Lemma 3.5.2), z has 4 square roots modulo N, namely $(x_p, x_q), (-x_p, x_q), (x_p, -x_q)$ and $(-x_p, -x_q)$. Since $y \neq \pm x$ and $y^2 \equiv_N z$, it must be the case that y equals either $(-x_p, x_q)$ or $(x_p, -x_q)$. Assume wlog that $y = (-x_p, x_q)$ (the other case is completely symmetric). Using the properties of the Jacobi symbol we have

$$\begin{aligned}\mathcal{J}_N(x) \cdot \mathcal{J}_N(y) &= \mathcal{J}_p(x) \cdot \mathcal{J}_q(x) \cdot \mathcal{J}_p(y) \cdot \mathcal{J}_q(y) \\ &= \mathcal{J}_p(x_p) \cdot \mathcal{J}_q(x_q) \cdot \mathcal{J}_p(-x_p) \cdot \mathcal{J}_q(x_q) \\ &= -\mathcal{J}_p^2(x_p) \cdot \mathcal{J}_q^2(x_q) = -1\end{aligned}$$

where in the last but one equality we used the fact that $\mathcal{J}_p(-x) = -\mathcal{J}_p(x)$ for all $x \in \mathbb{Z}_p^*$ and all primes p such that $p \equiv_4 3$. □

We are now ready to prove the following theorem.

Theorem 3.5.4. *Under the* 2v3Primes *assumption, the family $\mathcal{F}$ described in Figure 3.1 is a family of $(n, \frac{1}{4})$-LTDFs.*

Proof. We prove the properties one by one

- *Injectivity/Trapdoor:* Notice first that the Jacobi symbol $\mathcal{J}_N(x)$ can be efficiently computed even if the factorization of N is unknown. Hence $F(s, x)$ can be evaluated in polynomial time. Let now $(s, t) \leftarrow G(1^\lambda)$ (in particular $s = N$ where N is a Blum integer) and let $y' = F(s, x) = (y, b_1, b_2)$. We distinguish between the following two cases

  1. $y \in \mathbb{Z}_N^*$: Because of Lemma 3.5.2, y has 4 square roots modulo N which can be recovered using the trapdoor (p, q) (by first recovering the pairs of square roots modulo p and q separately and then combining them using the Chinese Remainder Theorem). Let $\pm x, \pm z$ be the 4 square roots of y modulo N. Since $\mathcal{P}_N(x) = -\mathcal{P}_N(-x)\ \forall x$ only one of $x, -x$ and one of $z, -z$ is consistent with b_1. Assume wlog that x, z are consistent

with b_1. Using Lemma 3.5.3 and since $x \neq \pm z$ $\mathcal{J}_N(z) = -\mathcal{J}_N(x)$ and hence only one of x, z is consistent with b_2 (recall that $x, z \in \mathbb{Z}_N^*$ and hence their Jacobi symbols are non-zero).

2. $\gcd(y, N) > 1$: Assume wlog that $\gcd(y, N) = p$. It is easy to see that in this case y has exactly 2 square roots (preimages) x and $-x$ (which can be recovered using the CRT) out of which, only one is consistent with b_1 (in this case we only need to check which of $x, -x$ satisfies $\mathcal{P}_N(\cdot) = b_1$).

This means that for all $(n+1)$-bit Blum Integers N output by $G(1^\lambda)$ and all $x \in \{0,1\}^n$ the triple $(x^2 \bmod N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ uniquely determines x. In addition, given (p, q), one can efficiently recover this unique preimage which concludes that $\mathcal{F}$ (defined over $\{0,1\}^n$) is a collection of injective trapdoor functions.

- *Lossiness:* Let $(\hat{s} = N, \perp) \leftarrow \hat{G}(1^\lambda)$. Consider the following sets

$$S_1 = \left\{ x \in \{0,1\}^n \,\middle|\, x \in \mathbb{Z}_N^* \text{ and } x < \frac{N}{2} \right\}$$
$$S_2 = \left\{ x \in \{0,1\}^n \,\middle|\, \gcd(x, N) > 1 \text{ and } x < \frac{N}{2} \right\}$$
$$S_3 = \left\{ x \in \{0,1\}^n \,\middle|\, x \geq \frac{N}{2} \right\}$$

Clearly S_1, S_2 and S_3 partition $\{0,1\}^n$. Also, because of lemma 3.5.2, squaring modulo $N = pqr$ is an 8-to-1 function over $\mathbb{Z}_N^*$. That means that y takes at most $\frac{\phi(N)}{8}$ values where $\phi(N)$ is Euler's totient function. Also for all $x \in S_1$ $\mathcal{P}_N(x) = 0$ by definition. Hence $(x^2 \bmod N, \mathcal{P}_N(x), \mathcal{Q}_N(x))$ for $x \in S_1$ takes at most $\frac{\phi(N)}{8} \cdot 2$ values, that is

$$|\mathsf{Img}(S_1)| \leq \frac{\phi(N)}{4} \,. \tag{3.5}$$

Also it is clear that $|S_2| = \frac{N-\phi(N)}{2}$ (there are $N - \phi(N)$ elements that are not coprime with N and exactly half of them are smaller than $N/2$). Finally, $|S_3| \leq 2^n - \frac{N}{2}$. That is,

$$|\mathsf{Img}(S_2)| \leq |S_2| \leq \frac{N - \phi(N)}{2} \quad \text{and} \quad |\mathsf{Img}(S_3)| \leq |S_3| \leq 2^n - \frac{N}{2}. \tag{3.6}$$

Combining equations (3.5) and (3.6) we get

$$\begin{aligned} |\mathsf{Img}(\{0,1\}^n)| &\leq |\mathsf{Img}(S_1)| + |\mathsf{Img}(S_2)| + |\mathsf{Img}(S_3)| \\ &\leq \frac{\phi(N)}{4} + \frac{N-\phi(N)}{2} + 2^n - \frac{N}{2} \\ &= 2^n - \frac{\phi(N)}{4} \leq 2^n - \frac{2^n}{5} = \frac{4}{5}2^n \\ &\leq 2^n 2^{-\frac{1}{4}} \end{aligned}$$

where in the penultimate inequality we used the fact that (for balanced primes p, q, r) $\phi(N) = (p-1)(q-1)(r-1) = N - O(N^{\frac{2}{3}})$ and hence $\frac{\phi(N)}{4} > \frac{N}{5} > \frac{2^n}{5}$. Therefore the image of $\{0,1\}^n$ when N is a product of 3 primes is at most $\frac{2^n}{2^{\frac{1}{4}}}$ which means that $F(\hat{s}, \cdot)$ loses (at least) $\frac{1}{4}$-bits.

- *Indistinguishability:* The fact that $s \simeq_c \hat{s}$ (where $(s, \cdot) \leftarrow \mathbb{G}(1^\lambda)$ and $(\hat{s}, \cdot) \leftarrow \hat{G}(1^\lambda)$) follows directly from the 2v3Primes assumption.

This concludes the proof that $\mathcal{F}$ as defined in Figure 3.1 is a family of $(n, \frac{1}{4})$-lossy functions. □

Remark 3.5.5. *It is interesting to note here that the above construction is an example of a lossy function for which lossiness cannot be amplified (by much) through the hardness assumption. That is, we can achieve higher lossines by assuming that is hard to distinguish products of 2 primes from products of k primes (for $k > 3$) but this increase in lossiness comes at the cost of strengthening the assumption which, in fact, becomes wrong when k is "large" (for modulus N of fixed length).*

Chapter 3 is, in part, a reprint of the paper "Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions" [120], co-authored with Scott Yilek, published in the proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010). Both authors contributed equally to this paper.

Chapter 4

Pseudorandom Generators from Knapsack Functions

Overview of the chapter. In this chapter we establish the connection between the search and decision problems associated to families of bounded knapsack functions. Namely, we explore the conditions on the underlying group $\mathbb{G}$ and the input distribution $\mathcal{X}$ under which a knapsack family, which is hard to invert, also produces pseudorandom outputs. We start by presenting our results in Section 4.1 and then briefly discuss related work in Section 4.2. The main result of this Chapter is Theorem 4.3.1 which provides sufficient (and in many cases necessary) conditions for the pseudorandomness of the output of a knapsack function. Section 4.3 contains the formal statement and proof of Theorem 4.3.1. The implications of Theorem 4.3.1 are laid out in Section 4.4 with emphasis on groups and input distributions that are important in cryptographic applications.

4.1 Results

Let $\mathbb{G}$ be any finite abelian group and $\mathbf{g} = (g_1, \ldots, g_m) \in \mathbb{G}^m$ be a sequence of m group elements. The sequence $\mathbf{g}$ gives rise to a linear function $f_{\mathbf{g}} \colon \mathbb{Z}^m \to \mathbb{G}$ defined as $f_{\mathbf{g}}(\mathbf{x}) = \sum_i x_i g_i$. $f_{\mathbf{g}}$ can be considered as the generalization of the subset sum function to arbitrary groups $\mathbb{G}$ (instead of just cyclic) and arbitrary

inputs (instead of binary). In this chapter we address the following question: under which conditions on the group $\mathbb{G}$ and the input distribution, the uninvertibility of $f_{\mathbf{g}}$ implies also that $f_{\mathbf{g}}$ produces pseudorandom inputs?

Our main technical result (Theorem 4.3.1) shows that for any finite abelian group $\mathbb{G}$ 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 $\mathbb{G}_d = \mathbb{G}/d\mathbb{G}$ 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 $\mathbb{G}_d$ considered are very small. So small that for many interesting groups and input distributions the folded knapsack function $f_{\mathbf{g}}(\mathbf{x})$ mod $d\mathbb{G}$ *compresses* the input (rather than stretching it) and produces an output which is *statistically* close to uniform. 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 of the form $\mathbb{G} = \mathbb{Z}_p^k$ for prime p fall into this category. This result generalizes those in [83] from uniform binary input to arbitrary input distributions.

- Distributions that, when folded (modulo small divisors of the order of $\mathbb{G}$,) maintain high entropy relative to the size of the quotient group $\mathbb{G}/d\mathbb{G}$. (See Theorem 4.4.3.) Groups of the form $\mathbb{G} = \mathbb{Z}_{2^\ell}^k$ 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.

4.2 Related Work

Our work relies upon and extends the work of Impagliazzo and Naor [83] who proposed simple and efficient pseudorandom generators based on the hardness of the *subset sum* problem. Roughly, the subset sum problem, parametrized by a set S, a positive integer n and a sequence of n elements $(a_1, ..., a_n)$ from S, is the problem of inverting a function $f : \{0,1\}^n \to S$ defined as $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i a_i \in S$. The main technical contribution of [83] is a reduction showing that, when S is a cyclic group, the hardness of inverting f implies that f is also a good pseudorandom generator. Borrowing techniques from [83], Fischer and Stern [59] extending the reduction to vector groups $\mathbb{Z}_2^k$ effectively constructing efficient pseudorandom generators based on the hardness of syndrome decoding.

We generalize the results of [83] and [59] in two ways:

- We consider functions over *arbitrary* finite groups $\mathbb{G}$. Only groups of the form $\mathbb{Z}_N$ were considered in [83] (for two representative choices of N, prime and power of 2), whereas [59] considered vector groups of the form $\mathbb{Z}_2^k$.

- We consider generalizations of the subset sum function (typically referred to as "knapsack" functions) where the input coefficients x_i take values from a set $\{0, \ldots, s\}$ (or, more generally $\{-s, \ldots, s\}$) for any (*polynomially bounded*) s, rather than just $\{0,1\}$. In addition, we consider *arbitrary* (possibly non-uniform) input distributions. By contrast, [83] and [59] considered only *binary* inputs. Moreover, in [83], the input is uniformly distributed over $\{0,1\}^n$, while in [59] it is distributed uniformly over all n-dimensional binary vectors with fixed Hamming weight.

Techniques. Our results rely heavily on Fourier analysis and, more specifically, on a recently developed algorithm [4] that, given oracle access to a function f (de-

fined over an arbitrary abelian group) "learns" its heavy Fourier coefficients. The use of Fourier analysis in inverting (learning) functions is by no means new. In fact, it has been extensively exploited within the learning theory community mostly in the study of functions defined over the boolean hybercube yielding elegant learning algorithms for a wide range of classes including decision trees [96], DNF formulas [24] juntas [121] and more. In Cryptography, two interesting applications of Fourier analysis are the Kushilevitz-Mansour [96] formulation of the proof of the Goldreich-Levin [67] hard-core predicate for any one-way function and the proof of hard-core predicates for several number-theoretic functions [4]. Both applications draw upon an intriguing connection between hard-core predicates and locally list-decodable codes which can be roughly summarized via the following analogy: Each predicate P defines an error correcting code $\mathcal{C}^P$, the existense of an imperfect predictor for P translates to local access to a corrupted codeword of $\mathcal{C}^P$ while recovering x from $f(x)$ (where f is presumably one-way) corresponds to list-decoding $\mathcal{C}^P$.

4.3 Pseudorandomness of Knapsack Functions

This section is dedicated to the proof of the following theorem.

Theorem 4.3.1 (Main). *For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$, finite abelian group $\mathbb{G}$ and input distribution $\mathcal{X}$ over $[s]^m \subset \mathbb{Z}^m$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is uninvertible and $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ is pseudorandom for all $d < s$, then Knap is also pseudorandom.*

Before we proceed to the proof a few remarks are in order.

Remark 4.3.2. *Theorem 4.3.1 as well as all its implications (see Section 4.4) 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 inputs $x_i \in \{a_i, a_i + 1, \ldots, b_i\}$ 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 $y_i = x_i - a_i \in \{0, \ldots, b_i - a_i\} \subseteq [s]$.*

Also, all statements remain essentially unchanged for distributions $\mathcal{X}$ such that $\Pr\{\mathbf{x} \notin [s]^m \mid \mathbf{x} \leftarrow \mathcal{X}\} = \mathsf{negl}(n)$ even if the support of $\mathcal{X}$ is possibly larger than $[s]^m$. For ease of exposition, we will omit dealing with these two technicalities throughout the rest of the chapter.

Remark 4.3.3. *For knapsack families $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ that* stretch *their input sufficiently, Theorem 4.3.1 is an* if and only if *statement. Indeed, if Knap is pseudorandom, then so is $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ for any d, because there is an efficiently computable regular transformation $(\mathbf{g}, g) \mapsto (\mathbf{g} \bmod d, g \bmod d)$ that maps $\mathcal{F}(\mathsf{Knap})$ to $\mathcal{F}(\mathsf{Knap}_d)$ and $\mathcal{U}(\mathbb{G}^m) \times \mathcal{U}(\mathbb{G})$ to $\mathcal{U}(\mathbb{G}_d^m) \times \mathcal{U}(\mathbb{G}_d)$. Moreover, if $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ stretches its input (or, more specifically, if the range $[\mathcal{F}(\mathsf{Knap})]$ is sparse in $\mathbb{G}^m \times \mathbb{G}$), then any inverter with noticeable success probability can be used as a distinguisher for $\mathcal{F}(\mathsf{Knap})$ in a straightforward way.*

4.3.1 Overview of the Proof

We prove Theorem 4.3.1 by contradiction. We show that for any finite abelian group $\mathbb{G}$ and any input distribution $\mathcal{X}$ over $[s]^m \subset \mathbb{Z}^m$ (for any $s = \mathsf{poly}(n)$), if $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ is pseudorandom for all $d < s$, then the existence of an efficient distinguisher for $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ implies the existence of an efficient inverter for $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$. The reduction proceeds into two steps and uses the notion of the *predictor*. Informally, a predictor is a weak form of an inverter algorithm that, on input a function $f \in F$, a target value $f(\mathbf{x})$, a value ℓ 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 $\mathbf{x}$ (see Section 4.3.2 for a formal definition). 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.

In the first step (Lemma 4.3.5) we show that any predictor for $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ (that guesses $\mathbf{x} \cdot \mathbf{r} \pmod{\ell}$ with "good" probability for "large" ℓ) can be efficiently transformed into an inverter for Knap. 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 4.3.10, Section 4.3.3), we prove that if there exists a distinguisher for Knap, but no distinguisher for $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$

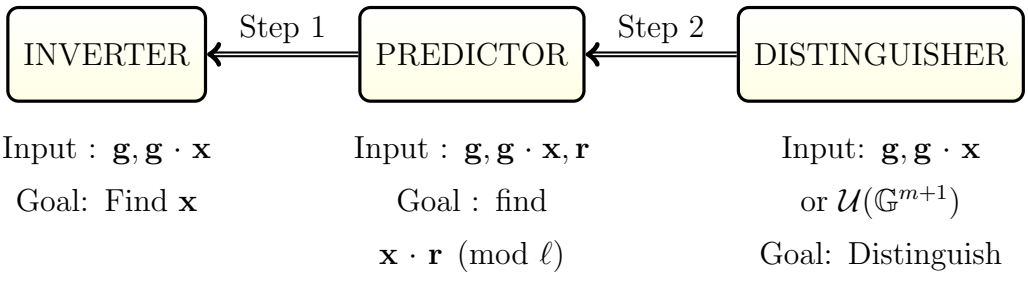

Figure 4.1: Overview of the proof of Theorem 4.3.1.

for small d, then there exists a sufficiently large ℓ and an associated predictor for Knap. This step is specific to knapsack families and relies on both the underlying group $\mathbb{G}$ and the distribution $\mathcal{X}$. Schematically, the proof is shown in Figure 4.1. Sections 4.3.2 and 4.3.3 are devoted to the first and the second step of the reduction respectively. The two steps combined yield Theorem 4.3.1 almost directly.

Proof. (of Theorem 4.3.1) Assume for the sake of contradiction that Knap is (t, θ)-distinguishable for some $t = \mathsf{poly}(n)$ and some noticeable θ. Since Knap_d is pseudorandom for all $d < s$, it follows by Proposition 4.3.10 that Knap is (t', ϵ, d^*)-biased for some $t' = O(t + \mathsf{poly}(n)) = \mathsf{poly}(n)$, noticeable ϵ and polynomially bounded $d^* \geq s$. Therefore, by Lemma 4.3.5, Knap is also $(t'/\delta, \delta)$-invertible for some $\delta = 1/\mathsf{poly}(n, \log d^*, 1/\epsilon) = 1/\mathsf{poly}(n)$ which contradicts the uninvertibility of Knap. □

4.3.2 Step 1: From Uninvertibility to Unpredictability

In this section we show how a "good" predictor can be turned to an inverter. We define two notions to measure the quality of a predictor: *accuracy* and *bias*. The first is probably the most natural notion, and directly measures the predictor's success probability. The second is more technical but carries more information about the error distribution of the predictor. The notion of bias is appropriate for applying the Fourier analytic techniques from Section 2.6, and its role will become evident in the proof of Lemma 4.3.5. For the special case of prime ℓ the two notions are closely related, as shown in Lemma 4.3.6.

Definition 4.3.4. *For any $\ell \in \mathbb{N}$ and function family $\mathsf{F} = (F, \mathcal{X})$ with domain $[\mathcal{X}] \subseteq \mathbb{Z}^m$ and range R, an* ℓ-predictor *for F is a probabilistic algorithm $\mathcal{P}$ that on input $(f, f(\mathbf{x}), \mathbf{r}) \in F \times R \times \mathbb{Z}_\ell^m$ outputs a value $\mathcal{P}(f, f(\mathbf{x}), \mathbf{r}) \in \mathbb{Z}_\ell$ 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*

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

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 $(F, \mathcal{X})$ is* (t, ϵ, ℓ)-biased *if it admits a (t, ϵ)-biased ℓ-predictor.*

We start by proving that if a function family $\mathsf{F} = (F, \mathcal{X})$ is uninvertible, then it is also unpredictable. This step of the reduction is not specific to knapsack families. Rather, it holds for any function family F with $[\mathcal{X}] \subseteq \mathbb{Z}^m$. At a high level, the proof first reduces the problem of inverting F, i.e., recovering the input $\mathbf{x}$ from $y = f(\mathbf{x})$ for some $f \in F$, to the problem of learning the heavy Fourier coefficients of a carefully chosen function h, and then uses the SFT algorithm from Theorem 2.6.1 for the latter problem. We remark that the learning takes place over the group $\mathbb{H} = \mathbb{Z}_\ell^m$, which is related to $[\mathcal{X}]$ but is unrelated to the group $\mathbb{G}$ of our knapsack function family. Lemma 4.3.5 provides sufficient conditions under which a predictor for a function family F implies the existence of an inverter for F.

Lemma 4.3.5. *For any $s \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$ and function family $\mathsf{F} = (F, \mathcal{X})$ defined over $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$, if F is (t, ϵ, ℓ)-biased for some $\ell \geq s$, then it is also $(t/\delta, \delta)$-invertible for $\delta = 1/\mathsf{poly}(n, \log \ell, 1/\epsilon)$.*

Proof. Let $\mathcal{P}$ be a (t, ϵ)-biased ℓ-predictor for F. We use $\mathcal{P}$ to devise an inverter $\mathcal{I}$ that on input $(f, f(\mathbf{x}))$ tries to recover $\mathbf{x}$ using the SFT algorithm from Theorem 2.6.1. In order to run SFT, the inverter $\mathcal{I}$ needs to provide answers to the queries $\mathbf{r} \in \mathbb{Z}_\ell^m$ made by SFT. The queries are answered invoking $\mathcal{P}$ on an appropriate input (to be defined). The goal is to present SFT with an oracle/function

$h : \mathbb{Z}_\ell^m \to \mathbb{C}$ which is *highly correlated* with the character $\chi_{\mathbf{x}}$ (for the unknown input $\mathbf{x}$), so that 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 SFT (from Theorem 2.6.1) with $\tau = \frac{\epsilon^2}{4}$. For every query $\mathbf{r} \in \mathbb{Z}_\ell^m$ issued by SFT, $\mathcal{I}$ runs $\mathcal{P}$ on input[1] $(f, f(\mathbf{x}), \mathbf{r}; coins)$ and returns $\omega_\ell^{\mathcal{P}(f,f(\mathbf{x}),\mathbf{r};coins)} \in \mathbb{T}$ to 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 SFT are answered according to a *deterministic* function

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

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 SFT upon termination and $S_{\mathbf{x}} = \{\mathbf{x}_i \in L \,\big|\, f(\mathbf{x}_i) = f(\mathbf{x})\}$ be the set of all values in L that map to $y = f(\mathbf{x})$ under f. $\mathcal{I}$ can construct $S_{\mathbf{x}}$ without knowing $\mathbf{x}$, by checking if $f(\mathbf{x}_i) = y$ for every element $\mathbf{x}_i \in L$. If $S_{\mathbf{x}}$ is non-empty, $\mathcal{I}$ selects a value $\mathbf{x}'$ randomly and uniformly from $S_{\mathbf{x}}$ and outputs it. Otherwise, $\mathcal{I}$ fails.

The running time of $\mathcal{I}$ is bounded by the running time of SFT times the running time of the predictor. Since $\|\widehat{h}_{f,f(\mathbf{x}),coins}\|_\infty = 1$, the running time of SFT is bounded by $\mathsf{poly}(m \log \ell, 1/\epsilon) = \mathsf{poly}(n, \log \ell, 1/\epsilon)$ (see Theorem 2.6.1) and hence the overall time of the inverter is $\mathsf{poly}(n, \log \ell, 1/\epsilon) \cdot t$.

It only remains to analyze the success probability of $\mathcal{I}$. Notice that, conditioned on $\mathbf{x} \in L$, $\mathcal{I}$ outputs $\mathbf{x}$ with probability $1/|S_{\mathbf{x}}| \geq 1/|L| \geq 1/\mathsf{poly}(n, \log \ell, 1/\epsilon)$. Therefore $\Pr\{\mathcal{I}(f, f(\mathbf{x})) = \mathbf{x}\} \geq \frac{\Pr\{\mathbf{x} \in L\}}{|L|}$. In order to bound $\Pr\{\mathbf{x} \in L\}$, we consider the Fourier transform of the function h used in answering the queries of SFT, and compute the Fourier coefficient corresponding to $\mathbf{x}$:

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

[1] The string *coins* is the internal randomness of $\mathcal{P}$.

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

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

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

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

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, $\Pr_{f,f(\mathbf{x}),coins}\{|\widehat{h}_{f,f(\mathbf{x}),coins}(\mathbf{x})| \ge \frac{\epsilon}{2}\} \ge \frac{\epsilon}{2}$. 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 hence it will be included in L with probability at least 2/3. So overall, $\Pr\{\mathcal{I}(f,f(\mathbf{x})) = \mathbf{x}\} \ge \frac{\Pr\{\mathbf{x}\in L\}}{|L|} \ge (\epsilon/2)\cdot(2/3)\cdot 1/|L| = 1/\mathsf{poly}(n, \log \ell, 1/\epsilon)$. □

The previous lemma uses the technical notion of bias to quantify the quality of a prediction algorithm. For the special case of a *prime* ℓ, it is sufficient to consider an *accurate* ℓ-predictor as the following lemma shows.

Lemma 4.3.6. *Let* $\mathsf{F} = (F, \mathcal{X})$ *be a function family with* $[\mathcal{X}] \subset \mathbb{Z}^m$. *For any prime* p, *if* F *admits a* (t,ϵ)*-accurate* p*-predictor, then it also admits a* (t,ϵ')*-biased* p*-predictor where* $\epsilon' = \epsilon p/(p-1) < \epsilon$.

Proof. Let $\mathcal{P}$ be a (t,ϵ)-accurate predictor for F. Consider now a predictor $\mathcal{P}'$, which 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: $\mathcal{P}'$ picks $y \leftarrow \mathcal{U}(\mathbb{Z}_p^*)$, runs $z \leftarrow \mathcal{P}(f, f(\mathbf{x}), y\mathbf{r})$, and returns 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

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

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

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

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

Combining the previous two lemmas, we obtain as a special case the results of [67, 68] for learning linear functions over a *field* given query access to a noisy version of the function.

Corollary 4.3.7. *For any $s \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$ and function family $\mathsf{F} = (F, \mathcal{X})$ with $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$, if there exists a (t, ϵ)-accurate p-predictor for some prime p and F, then F is $(t/\delta, \delta)$-invertible, for $\delta = 1/\mathsf{poly}(n, \log p, 1/\epsilon)$.*

Proof. Easily follows from Lemma 4.3.5 and Lemma 4.3.6. □

4.3.3 Step 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 $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ with noticeable distinguishing advantage can be turned into a predictor for Knap with *noticeable bias*. At a high level, the predictor works as follows: on input a modulus ℓ, function $\mathbf{g} \in \mathbb{G}^m$,

$y = \mathbf{g} \cdot \mathbf{x} \in \mathbb{G}$ and $\mathbf{r} \in \mathbb{Z}_\ell^m$, it first makes a guess for the inner product $\mathbf{x} \cdot \mathbf{r}$ mod ℓ; it then uses that guess to modify the knapsack instance $(\mathbf{g}, y = \mathbf{g} \cdot \mathbf{x}) \in \mathbb{G}^n \times \mathbb{G}$ into a related 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 [83]. However, in the restricted subset-sum setting considered in [83], the reduction is rather straightforward: if the guess for $\mathbf{x} \cdot \mathbf{r}$ mod ℓ is correct, then the modified knapsack instance $(\mathbf{g}', y')$ is distributed according to $\mathcal{F}(\mathsf{Knap})$, whereas if the guess is wrong, the distribution of $(\mathbf{g}', y')$ is (statistically close to) uniform over $\mathbb{G}^m \times \mathbb{G}$. But these are exactly the two distributions that $\mathcal{D}$ can tell apart and therefore a noticeable distinguishing advantage translates directly into an accurate (or biased) predictor.

When considering general abelian groups and distributions $\mathcal{X}$ with $[\mathcal{X}] \not\subseteq \{0,1\}^m$, several technical difficulties arise. Unlike [83], if the guess for $\mathbf{x} \cdot \mathbf{r}$ mod ℓ 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 $\mathcal{F}_d(\mathsf{Knap})$ for any divisor d of the group exponent $\mathrm{M}_\mathbb{G}$. Notice that for $d = 1$ and $d = \mathrm{M}_\mathbb{G}$ we get the two "extreme" distributions $\mathcal{F}_1(\mathsf{Knap}) = \mathcal{U}(\mathbb{G}^m \times \mathbb{G})$ and $\mathcal{F}_{\mathrm{M}_\mathbb{G}}(\mathsf{Knap}) = \mathcal{F}(\mathsf{Knap})$ respectively. However, other $\mathcal{F}_d(\mathsf{Knap})$ (with $1 < d < \mathrm{M}_\mathbb{G}$) can also arise. Depending on the order and structure of the underlying group $\mathbb{G}$, and the output distribution of the distinguisher $\mathcal{D}$ on the various auxiliary distributions $\mathcal{F}_d(\mathsf{Knap})$, the technical details of the reduction differ significantly. As a warm-up, we first present a weak form of our main theorem.

Proposition 4.3.8. *For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$, finite abelian group $\mathbb{G}$ and input distribution $\mathcal{X}$ over $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is (t, δ)-distinguishable from uniform for some noticeable δ, but $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ is pseudorandom for all $d < 2ms^2$, then for any prime p with $s \leq p < 2s = \mathsf{poly}(n)$, Knap is $(O(t + m), \epsilon, p)$-biased for some noticeable[2] ϵ.*

Proof. Let $\mathcal{D}$ be a (t, δ)-distinguisher for $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$. To simplify notation, we

[2] 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 $\epsilon \geq \delta/(2ms^2)$.

define $\beta_d = \Pr\{\mathcal{D}(\mathcal{F}_d(\mathsf{Knap})) = 1\}$. Notice that $\beta_{\mathrm{M}_\mathbb{G}} = \Pr\{\mathcal{D}(\mathcal{F}(\mathsf{Knap})) = 1\}$ and $\beta_1 = \Pr\{\mathcal{D}(\mathcal{U}(\mathbb{G}^m \times \mathbb{G})) = 1\}$. By hypothesis, $\beta_{\mathrm{M}_\mathbb{G}} - \beta_1 = \delta$ (because $\mathcal{D}$ is a (t, δ)-distinguisher for $\mathcal{F}(\mathsf{Knap})$), while $\beta_d - \beta_1 = \mathsf{negl}(n)$ for all $d < 2ms^2$ (because Knap_d is pseudorandom for all $d < 2ms^2$).

Let p be any prime between s and $2s$. We need to show that there is an ϵ-biased p-predictor. Since p is prime, it is enough to find a p-predictor which is ϵ-accurate, rather than ϵ-biased. The existence of an ϵ-biased predictor then follows by Lemma 4.3.6.

The ϵ-accurate predictor $\mathcal{P}$ is shown in Figure 4.2. 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 $\mathcal{F}_{\mathrm{M}_\mathbb{G}}(\mathsf{Knap}) = \mathcal{F}(\mathsf{Knap})$, otherwise it invokes $\mathcal{D}$ on $\mathcal{F}_d(\mathsf{Knap})$ for some $d < m(s-1)p < 2ms^2$. But for all such d, $\mathcal{F}(\mathsf{Knap}_d)$ and therefore $\mathcal{F}_d(\mathsf{Knap})$ (by Lemma 2.3.3) is pseudorandom, so $\mathcal{D}$ will behave as if it had been invoked on the uniform distribution $\mathcal{F}_1(\mathsf{Knap})$. 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. Details follow.

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

1. Pick $c \leftarrow \mathcal{U}(\mathbb{Z}_{m(s-1)p})$
2. Pick $g \leftarrow \mathcal{U}(\mathbb{G})$
3. $\bar{\mathbf{g}} \leftarrow \mathbf{g} - \mathbf{r} \cdot g$
4. Run $\mathcal{D}$ on input $(\bar{\mathbf{g}}, y - c \cdot g)$
5. **if** $\mathcal{D}$ outputs 1
6. $\quad$ *guess* $\leftarrow c \bmod p$
7. **else**
8. $\quad$ *guess* $\leftarrow \mathcal{U}(\mathbb{Z}_p \setminus (c \bmod p))$
9. **return** *guess*

Figure 4.2: Predictor for Proposition 4.3.8 (weak predictor).

Let $c' = \mathbf{x} \cdot \mathbf{r} = A \cdot p + v$ $(0 \le v < p)$ 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, Figure 4.2) is $(\bar{\mathbf{g}}, R)$ where

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

By Lemma 2.3.1, the input to $\mathcal{D}$ is $(\bar{\mathbf{g}}, \bar{\mathbf{g}}\cdot\mathbf{x} + d\cdot g) = \mathcal{F}_d(\mathsf{Knap})$ for $d = \gcd_{\mathbb{G}}(c'-c)$. So, the probability that $\mathcal{D}$ outputs 1 is by definition $\beta_{\gcd_{\mathbb{G}}(c'-c)}$.

For any d, let C_d be the event "$\gcd_{\mathbb{G}}(c'-c) = d$". Clearly, $\sum_{d|\mathrm{M}_{\mathbb{G}}} \Pr\{C_d\} = 1$. Notice that since $c, c' \in [m(s-1)p]$, we only need to consider either $d = \mathrm{M}_G$ (when $c' = c$), 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

$$\Pr\{guess = v\} = \sum_{d|\mathrm{M}_{\mathbb{G}}} \Pr\{guess = v \mid C_d\} \Pr\{C_d\}. \tag{4.1}$$

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

$$\Pr\{guess = v \mid C_d\} = \alpha_d \cdot \beta_d + \frac{1-\alpha_d}{p-1}(1-\beta_d) \tag{4.2}$$

where $\alpha_d = \Pr\{c = c' \pmod p \mid C_d\}$. It immediately follows from the definition that $\sum_{d|\mathrm{M}_{\mathbb{G}}} \alpha_d \Pr\{C_d\} = \Pr\{c = c' \pmod p\} = \frac{1}{p}$. Notice that for $d = \mathrm{M}_{\mathbb{G}}$ we have $\alpha_{\mathrm{M}_{\mathbb{G}}} = 1$, $\Pr\{C_{\mathrm{M}_{\mathbb{G}}}\} = 1/(m(s-1)p)$ and $\beta_{\mathrm{M}_{\mathbb{G}}} = \beta_1 + \delta$. For all other d (with $\Pr\{C_d\} \neq 0$,) we have $\beta_d = \beta_1 + \mathsf{negl}(n)$. Plugging (4.2) in (4.1) and simplifying, we obtain

$$\begin{aligned} \Pr\{guess = v\} &= \sum_{d|\mathrm{M}_{\mathbb{G}}} \Pr\{C_d\} \left(\alpha_d\beta_d + \frac{1-\alpha_d}{p-1}(1-\beta_d)\right) \\ &= \Pr\{C_{\mathrm{M}_{\mathbb{G}}}\} \left(\alpha_{\mathrm{M}_{\mathbb{G}}}\beta_{\mathrm{M}_{\mathbb{G}}} + \frac{1-\alpha_{\mathrm{M}_{\mathbb{G}}}}{p-1}(1-\beta_{\mathrm{M}_{\mathbb{G}}})\right) \\ &+ \sum_{d|\mathrm{M}_{\mathbb{G}}, d<\mathrm{M}_{\mathbb{G}}} \Pr\{C_d\} \left(\alpha_d(\beta_1 + \mathsf{negl}) + \frac{1-\alpha_d}{p-1}(1-\beta_1-\mathsf{negl})\right) \\ &\geq \frac{1}{p} + \frac{\delta}{m(s-1)p} - \mathsf{negl}(n)\ . \end{aligned}$$

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 4.3.6, there is also an ϵ-biased predictor with essentially the same running time $O(m+t)$ as $\mathcal{P}$. □

Proposition 4.3.8 along with Lemma 4.3.5 already gives search to decision reductions for some interesting knapsack families, but it requires (as an assumption) the pseudorandomness of $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ for values of d in a larger range than what Theorem 4.3.1 specifies. The following lemma plays a crucial role in the proof of Proposition 4.3.10, which extends Proposition 4.3.8 to hold under the assumptions in Theorem 4.3.1.

Lemma 4.3.9. *For any $d \in \mathbb{N}$, $\sum\{r^2 \colon 1 \leq r < d, r|d\} \leq \left(\frac{\pi^2}{6} - 1\right) \cdot d^2$.*

Proof. Let $r_1 > r_2 > \ldots > r_k = 1$ be all the *proper* divisors of d. Clearly $r_i \leq \frac{d}{i+1}$ (the largest proper divisor of d is at most $d/2$, the second largest is at most $d/3$ and so on). It follows that

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

Proposition 4.3.10. *For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$, finite abelian group $\mathbb{G}$ and input distribution $\mathcal{X}$ over $[\mathcal{X}] \subseteq [s]^m \subset \mathbb{Z}^m$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is (t, δ)-distinguishable from uniform for some noticeable δ, but $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ is pseudorandom for all $d < s$, then Knap 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 Knap_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 [83] and Proposition 4.3.8, is unlikely to work for the following reason: Even though a correct guess for $\mathbf{x} \cdot \mathbf{r}$ results in the distribution $\mathcal{F}(\mathsf{Knap}) = \mathcal{F}_{\mathrm{M}_\mathbb{G}}(\mathsf{Knap})$ being input to the distinguisher as desired, a wrong guess produces $\mathcal{F}_d(\mathsf{Knap})$ for some d smaller than $2ms^2$ *but possibly larger than s*. In that case, we have no guarantee that the distinguishing advantage between $\mathcal{F}_{\mathrm{M}_\mathbb{G}}(\mathsf{Knap})$ and $\mathcal{F}_d(\mathsf{Knap})$ (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[3] d of $\mathrm{M}_\mathbb{G}$ (with $s \leq d < 2ms^2$). For such a divisor d, our predictor runs the distinguisher with

[3]If the interval $[s, 2ms^2)$ does not contain any divisor of $\mathrm{M}_\mathbb{G}$, we can apply Proposition 4.3.8 directly.

input $\mathcal{F}_d(\mathsf{Knap})$ whenever the guess for $\mathbf{x}\cdot\mathbf{r} \pmod d$ is correct or with $\mathcal{F}_{d'}(\mathsf{Knap})$ 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 divisor d for which the distinguishing gap of $\mathcal{D}$ between $\mathcal{F}_d(\mathsf{Knap})$ and $\mathcal{F}_{d'}(\mathsf{Knap})$ is sufficiently large for all $d' \mid d$. 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 [68] hold over *fields*). Here is where the power of Lemma 4.3.5 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 the proof of Proposition 4.3.8. Namely for a (t,δ)-distinguisher $\mathcal{D}$ we define $\beta_d = \Pr\{\mathcal{D}(\mathcal{F}_d(\mathsf{Knap})) = 1\}$. 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, $\beta_{\mathrm{M}_\mathbb{G}} - \beta_1 = \delta$ while $\beta_d - \beta_1 = \mathsf{negl}(n)$ for all $d < s$. We can further assume that there exists $\tilde{d}$ with $s \le \tilde{d} \le 2ms^2 = \mathsf{poly}(n)$ such that $\beta_{\tilde{d}} - \beta_1 = \tilde{\delta}$ for some *noticeable* $\tilde{\delta}$ (otherwise the proof follows directly from Proposition 4.3.8). Let d^* be the *smallest* divisor of $\tilde{d}$ such that[4] $|\beta_{d^*} - \beta_1| \ge \frac{d^{*3}\tilde{\delta}}{\tilde{d}^3}$. Notice that d^* has the following two useful properties: (a) $d^* \ge s$. This is true because $|\beta_{d^*} - \beta_1| \ge \frac{d^{*3}\tilde{\delta}}{\tilde{d}^3} = 1/\mathsf{poly}(n)$ whereas, by hypothesis, $|\beta_d - \beta_1| = \mathsf{negl}(n)$ 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 Knap. $\mathcal{P}$ is shown in Figure 4.3. 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 $c' = \mathbf{r}\cdot\mathbf{x} = A\cdot d^* + v$ $(0 \le v < d^*)$ be the inner product of $\mathbf{x}\cdot\mathbf{r}$ *over*

[4]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: $guess \in \mathbb{Z}_{d^*}$

1. Pick $c \leftarrow \mathcal{U}(\mathbb{Z}_{d^*})$
2. Pick $g_1 \leftarrow \mathcal{U}(\mathbb{G}), g_2 \leftarrow \mathcal{U}(\mathbb{G})$
3. $\bar{\mathbf{g}} \leftarrow \mathbf{g} - \mathbf{r} \cdot g_1$
4. Run $\mathcal{D}$ on input $(\bar{\mathbf{g}}, y - c \cdot g_1 + d^* \cdot g_2)$
5. **if** $\mathcal{D}$ returns 1
6. $\quad guess \leftarrow c$
7. **else**
8. $\quad guess \leftarrow \mathcal{U}(\mathbb{Z}_{d^*} \setminus c)$
9. **return** *guess*

Figure 4.3: Predictor for Proposition 4.3.10 (strong predictor).

the integers. The input to the distinguisher $\mathcal{D}$ (line 4, Figure 4.3) 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 2.3.1, $\mathcal{D}$ is invoked on $(\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + d^* g) = \mathcal{F}_{d^*}(\mathsf{Knap})$.
If $c \neq v$ then $\mathcal{D}$ gets $(\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + \gcd(Ad^* + v - c, d^*) \cdot \mathcal{U}(G)) = (\bar{\mathbf{g}}, \bar{\mathbf{g}} \cdot \mathbf{x} + d' \cdot \mathcal{U}(G)) = \mathcal{F}_{d'}(\mathsf{Knap})$ for some $d' \mid d^*$ with $d' < d^*$ (notice that $c, v \in [d^*]$ and hence if $c \neq v$, then $d^* \nmid Ad^* + v - c$). More specifically, $d' = \gcd(v - c, d^*) = \gcd(c - v, d^*)$.

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, since $c \in \mathbb{Z}_{d^*}$ is chosen uniformly at random, $\Pr\{C_j\} = 1/d^*$ 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

$$\begin{aligned} Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} &= \Pr\{guess \equiv_{d^*} v+k\} \\ &= \sum_{j=0}^{d^*-1} \Pr\{guess \equiv_{d^*} v+k \,\big|\, C_j\} \Pr\{C_j\} \qquad (4.3) \\ &= \frac{1}{d^*} \sum_{j=0}^{d^*-1} \Pr\{guess \equiv_{d^*} v+k \,\big|\, C_j\} \ . \end{aligned}$$

If we further condition on whether $\mathcal{D}$ outputs 1 or 0, it is not hard to see that

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

Replacing in (4.3) 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

$$\begin{aligned} \left| \sum_{k=0}^{d^*-1} Pr\{k \leftarrow \mathcal{E}_{d^*}(\mathcal{P})\} \omega_{d^*}^k \right| &= \frac{1}{d^*-1} \left| \sum_{k=0}^{d^*-1} (\beta_{\gcd(k,d^*)} - \beta_1) \omega_{d^*}^k \right| \\ &\geq \frac{1}{d^*-1} \left[|\beta_{d^*} - \beta_1| - \sum_{k=1}^{d^*-1} \left| \beta_{\gcd(k,d^*)} - \beta_1 \right| \right] \qquad (4.4) \end{aligned}$$

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

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

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

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 (4.4) 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 4.3.9. □

4.4 Implications and applications

Theorem 4.3.1 provides explicit and usable criteria for checking if the output of a knapsack function family, that is (assumed to be) uninvertible, is pseudorandom. Given a group $\mathbb{G}$ and an input distribution $\mathcal{X}$, one needs only to check that the folded knapsack families $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$ are pseudorandom. As it turns out, for many choices of $(\mathbb{G}, \mathcal{X})$, the folded knapsack functions Knap_d *compress* their input and map the input distribution $\mathcal{X}$ to a distribution which is *statistically close* to the uniform distribution over G_d. More specifically, $\Delta_U(\mathcal{F}(\mathsf{Knap}_d)) = \mathsf{negl}(n)$, and Knap_d is pseudorandom in a strong statistical sense. Below, we provide some representative examples where that happens, focusing on those that are most interesting in applications. But before we do that, it is instructive to digress a little and explore a choice of $(\mathbb{G}, \mathcal{X})$ for which uninvertibility does *not* imply pseudorandomness[6]. According to Theorem 4.3.1, in any such counter-example, there should exist a divisor d of $|\mathbb{G}|$ such that $\mathcal{F}(\mathsf{Knap}_d)$ can be efficiently distinguished from the uniform distribution.

[6]Strictly speaking, what we prove is: there exist group $\mathbb{G}$ and distribution $\mathcal{X}$ for which $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is widely believed to be uninvertible whereas it is provably not pseudorandom.

Lemma 4.4.1. *If there exists a group $\mathbb{G}$ and an input distribution $\mathcal{X}$ such that $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is uninvertible, then there exist a group $\mathbb{G}'$ and a distribution $\mathcal{X}'$ such that $\mathsf{Knap}[\mathbb{G}', \mathcal{X}']$ is uninvertible, but not pseudorandom.*

Proof. Let p be a small prime such that $\gcd(p, \mathrm{M}_{\mathbb{G}}) = \gcd(p, |\mathbb{G}|) = 1$. Notice that $\mathrm{M}_{\mathbb{G}} \leq |\mathbb{G}|$ has less than $\log_2 |\mathbb{G}|$ distinct prime factors. So, we can always choose p among the first $\log_2 |\mathbb{G}|$ primes, and $p = O(\log |\mathbb{G}| \log\log |\mathbb{G}|)$. (If $|\mathbb{G}|$ or $\mathrm{M}_{\mathbb{G}}$ 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 $\mathrm{M}_{\mathbb{G}} \leq B$ 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 $\mathbb{G}' \simeq \mathbb{G} \times \mathbb{Z}_p$, and $\mathcal{X}' = p\mathcal{X} = \{px \mid x \leftarrow \mathcal{X}\}$. First notice that $\mathsf{Knap}[\mathbb{G}', \mathcal{X}']$ is not pseudorandom: on input a function $\mathbf{g}' = (g'_1, \ldots, g'_m) \in \mathbb{G}'^m$ and element $(r'_1, r'_2) \in \mathbb{G}'$, a distinguisher simply outputs 1 if and only if $r'_2 = 0_{\mathbb{G}'}$. It is easy to check that the distinguisher has advantage $1 - 1/p \geq 1/2$.

However, $\mathsf{Knap}[\mathbb{G}', \mathcal{X}']$ is uninvertible. Indeed, assume that there exists a (t', ϵ)-inverter $\mathcal{I}'$ for $\mathsf{Knap}[\mathbb{G}', \mathcal{X}']$. Consider the following inverter $\mathcal{I}$ against $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$. On input $\mathbf{g} \leftarrow \mathcal{U}(\mathbb{G}^m)$ and target $y = \mathbf{g} \cdot \mathbf{x} \in \mathbb{G}$, $\mathcal{I}$ 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 $y' = (py, 0) = \mathbf{g}' \cdot (p\mathbf{x})$. Notice that $(\mathbf{g}', y')$ is distributed according to $\mathcal{F}(\mathsf{Knap}[\mathbb{G}', \mathcal{X}'])$, exactly as required by $\mathcal{I}'$. If $\mathcal{I}'$ outputs $\mathbf{z}$, then $\mathcal{I}$ outputs $\mathbf{z}/p$. (Since $\gcd(p, \mathrm{M}_{\mathbb{G}}) = 1$, multiplication by p is an invertible function from $\mathbb{G}$ to $\mathbb{G}$.) The success probability of $\mathcal{I}$, is $\Pr\{\mathcal{I}(\mathbf{g}, \mathbf{g}\cdot\mathbf{x}) = \mathbf{x}\} = \Pr\{\mathcal{I}'(\mathbf{g}', \mathbf{g}' \cdot (p\mathbf{x})) = p\mathbf{x}\} \geq \epsilon$. This proves that $\mathcal{I}$ is a (t, ϵ)-inverter where $t \approx t'$. □

4.4.1 Examples of Pseudorandom Knapsack Families

In this section, we present broad choices of groups $\mathbb{G}$ and distributions $\mathcal{X}$ that give rise to pseudorandom knapsack families. We start with groups $\mathbb{G}$ whose order does not contain any factors that are within the maximum value the input can take. In this case uninvertibility implies pseudorandomness for *any* input distribution.

Lemma 4.4.2. *Let $\mathbb{G}$ by any finite abelian group and p be the smallest prime factor of $|\mathbb{G}|$. For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$ and input distribution $\mathcal{X}$ over $[\mathcal{X}] \subseteq [s]^m \subseteq [p]^m$, if $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ is uninvertible, then it is also pseudorandom.*

Proof. Consider Knap_d for any $d < s \leq p$. Since $\gcd(d, |\mathbb{G}|) = 1$ for all $d < p$, we have $d\mathbb{G} = \mathbb{G}$. It follows that the range of Knap_d is $\mathbb{G}_d = \mathbb{G}/d\mathbb{G} = \{0_{\mathbb{G}}\}$ and therefore Knap_d is trivially pseudorandom for every $d < s$. The lemma then follows directly from Theorem 4.3.1. □

Lemma 4.4.2 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 4.4.2 significantly extends the results from [83] and [59]. More specifically, it asserts that *any* knapsack family $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ with $[\mathcal{X}] \subseteq \{0,1\}^m$ is pseudorandom provided it is uninvertible, for *any* abelian group $\mathbb{G}$ and *any* (not necessarily uniform) binary input distribution $\mathcal{X}$. Lemma 4.4.2 applies directly to other choices of $(\mathbb{G}, \mathcal{X})$ including groups with prime order (and any distribution $\mathcal{X}$), vector groups with prime exponent, i.e. $\mathbb{G} \simeq \mathbb{Z}_p^k$ and, more broadly, groups of the form $\mathbb{Z}_{p^e}^k$ where p is a prime and $[\mathcal{X}] \subseteq [s]^m$ for some $s = \mathsf{poly}(n) \leq p$.

For groups with small prime factors (smaller than s, where $[\mathcal{X}] \subseteq [s]^m$), the connection between uninvertibility and pseudorandomness is more subtle. In order to prove search to decision reductions in such cases, the group and input distribution need to be *restricted* somehow. Still, we can use our main theorem for a wide range of groups and input distributions. In the remaining of the section we provide a few representative examples, focusing on *vector groups* $\mathbb{G} = \mathbb{Z}_q^k$ both for simplifying the exposition and because these groups are more interesting from a cryptographic viewpoint (see Section 5.4, Chapter 5).

For a vector group $\mathbb{G} = \mathbb{Z}_q^k$ consider the *folded* knapsack function $\mathsf{Knap}_d = \mathsf{Knap}[\mathbb{G}_d, \mathcal{X}]$. First notice that $\mathrm{M}_{\mathbb{G}} = q$ and $d\mathbb{G} = d\mathbb{Z}_q^k = \gcd(d,q) \cdot \mathbb{Z}_q^k \simeq \mathbb{Z}_{q/\gcd(d,q)}$. According to Theorem 4.3.1, proving pseudorandomness of $\mathsf{Knap}[\mathbb{G}, \mathcal{X}]$ reduces to proving that the folded families Knap_d are pseudorandom for all $d < s$ with $d \mid q$. In fact, below we prove that in many interesting settings the function families Knap_d are *statistically* random. Lemma 4.4.3 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 $|\mathbb{G}_d| = |\mathbb{Z}_d^k| = d^k$.

Lemma 4.4.3. *For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$, $q \in \mathbb{N}$ and input distribution $\mathcal{X}$ over $[\mathcal{X}] \subseteq [s]^m$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{Z}_q^k, \mathcal{X}]$ is uninvertible and $\mathsf{Col}(\mathcal{X}_d) = \mathsf{negl}(n)/d^k$ for all $d \mid q$ with $d < s$, then Knap is also pseudorandom.*

Proof. For any divisor d of q, $G_d = \mathbb{Z}_q^k / d\mathbb{Z}_q^k \sim \mathbb{Z}_d^k$. Given Theorem 4.3.1, it suffices to prove that $\Delta_U(\mathcal{F}(\mathsf{Knap}[\mathbb{Z}_d^k, \mathcal{X}]) = \mathsf{negl}(n)$ for all divisors $d \mid q$ with $d < s$. For the latter, we apply Lemma 2.3.4 with $\mathbb{H} = \mathbb{G}_d = \mathbb{Z}_d^k$, and $\mathbb{H}_{\tilde{d}} = \mathbb{Z}_{\tilde{d}}^k$ to get

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

where we used the hypothesis that $\mathsf{Col}(\mathcal{X}_{\tilde{d}}) = \mathsf{negl}(n)/\tilde{d}^k$ 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 $\tilde{d}$. □

Uniformly Folded Distributions. We now present a natural family of distributions which have small collision probability when folded, and thereby result in pseudorandom knapsack families. For a given group $\mathbb{G}$, we say that a distribution $\mathcal{X}$ with $[\mathcal{X}] \subseteq [s]^m$ is *uniformly folded* with respect to $\mathbb{G}$, if $\mathcal{X}_d \simeq_s \mathcal{U}(\mathbb{Z}_d^m)$ is (statistically close to) the uniform distribution for all $d < s$ such that $d \mid \mathrm{M}_{\mathbb{G}}$.

Lemma 4.4.4. *For any $s = \mathsf{poly}(n) \in \mathbb{N}$, $m = \mathsf{poly}(n) \in \mathbb{N}$, $q \in \mathbb{N}$, $k \leq m - \omega(\log n) \in \mathbb{N}$ and distribution $\mathcal{X}$ over $[\mathcal{X}] \subseteq [s]^m$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{Z}_q^k, \mathcal{X}]$ is uninvertible and $\mathcal{X}$ is uniformly folded with respect to $\mathbb{Z}_q^k$, then Knap is also pseudorandom.*

Proof. Directly follows from Lemma 4.4.3 and from the fact that if $\mathcal{X}_d = \mathcal{U}(\mathbb{Z}_d^m)$, then $\mathsf{Col}(\mathcal{X}_d) = 1/d^m$. □

Two examples of uniformly folded distributions are $\mathcal{X} = \mathcal{U}(\mathbb{Z}_q^m)$ (with respect to group $\mathbb{G} = \mathbb{Z}_q^k$ for any q and k) and $\mathcal{X} = \mathcal{U}(\mathbb{Z}_{p^i}^m)$ (with respect to group $\mathbb{G} = \mathbb{Z}_{p^e}^k$ for prime p and $i \leq e$). As an immediate corollary to Lemma 4.4.4, we obtain the following.

Corollary 4.4.5. *For any $m = \mathsf{poly}(n) \in \mathbb{N}$, $k \leq m - \omega(\log n) \in \mathbb{N}$, prime p, and $i \in \mathbb{N}$ such that $p^i = \mathsf{poly}(n)$, if $\mathsf{Knap} = \mathsf{Knap}[\mathbb{Z}_{p^e}^k, \mathcal{U}(\mathbb{Z}_{p^i}^m)]$ is uninvertible, then Knap is also pseudorandom.*

Gaussian Distributions. 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 function families defined over $\mathbb{G} \simeq \mathbb{Z}_q^k$ with Gaussian-like input distribution. We state the result for *discrete Gaussians* (defined in Section 2.5). Qualitatively similar results hold for *discretized* (rounded) Gaussians.

Lemma 4.4.6. *For any positive integers k, m, q such that $m = \mathsf{poly}(n)$ and $k \leq m - \omega(\log n)$ and for any $r \in \mathbb{R}^+$ such that[7] $r \leq \mathsf{poly}(n)$, if $\mathsf{Knap}[\mathbb{Z}_q^k, \mathcal{D}_{\mathbb{Z}^m,r}]$ is uninvertible and q has no divisors in the interval $\left(\left(\frac{r}{\omega(\sqrt{\log n})} \right)^{m/k}, r \cdot \omega(\sqrt{\log n}) \right)$, then it is also pseudorandom.*

Proof. By a standard tail inequality,

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

This means that effectively $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m,r}$ takes values in $\mathcal{S}^m$ where $\mathcal{S} = \{-\lfloor r \cdot \omega(\sqrt{\log n})/2 \rfloor + 1, \ldots, \lfloor r \cdot \omega(\sqrt{\log n})/2 \rfloor - 1\}$ is an interval of size $s < r \cdot \omega(\sqrt{\log n}) = \mathsf{poly}(n)$. According to Theorem 4.3.1, it suffices then to prove that for every $d < r \cdot \omega(\sqrt{\log n})$ that divides q, the knapsack family $\mathsf{Knap}[\mathbb{Z}_d^k, \mathcal{D}_{\mathbb{Z}^m,r}]$ is statistically close to the uniform distribution over $\mathbb{Z}_d^{k \times m} \times \mathbb{Z}_d^k$. Let $d < r \cdot \omega(\sqrt{\log n})$ be any divisor of q. By hypothesis, $d < (r/\omega(\sqrt{\log n}))^{m/k}$. For a vector $\mathbf{g} = (g_1, \ldots, g_m) \in (\mathbb{Z}_d^k)^m$ the member function $f_{\mathbf{g}}$ of $\mathsf{Knap}[\mathbb{Z}_d^k, \mathcal{D}_{\mathbb{Z}^m,r}]$ can be compactly represented as $f_{\mathbf{g}}(\mathbf{x}) = \mathbf{G} \cdot \mathbf{x}$ where $\mathbf{G}$ is a $k \times m$ matrix whose columns are the group elements $g_1, \ldots, g_m \in \mathbb{Z}_d^k$. Consider the lattice $\Lambda_d(\mathbf{G})$ generated by the rows of $\mathbf{G}$. The following claim states that if $\mathbf{G}$ is sampled uniformly at random from $\mathbb{Z}_d^{k \times m}$, then the shortest vector of $\Lambda_d(\mathbf{G})$ (in l_∞-norm) cannot be very short except with very small probability.

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

Claim 4.4.7. *Let k, m be positive integers. Then for all but at most 2^{-m} fraction of matrices $\mathbf{G} \in \mathbb{Z}_d^{k \times m}$,*

$$\lambda_1^\infty(\Lambda_d(\mathbf{G})) \geq \frac{d^{1-k/m}}{4}.$$

Proof. (of Claim 4.4.7) For a matrix $\mathbf{G} \in \mathbb{Z}_d^{k \times m}$ the shortest vector of $\Lambda_d(\mathbf{G})$ can be written as $\mathbf{v} = \mathbf{G}^T \cdot \mathbf{x}$ where $\mathbf{x} \in \mathbb{Z}_d^k$ and $\gcd(\mathbf{x}, d) = 1$ (if $\gcd(\mathbf{x}, d) = d' > 1$, then $\mathbf{G}^T\mathbf{x}/d'$ also belongs to $\Lambda_d(\mathbf{G})$ and is shorter than $\mathbf{G}^T\mathbf{x}$). Fix any $\mathbf{x} \in \mathbb{Z}_d^k$ with $\gcd(\mathbf{x}, d) = 1$. Then $\{\mathbf{G}^T\mathbf{x} \,\big|\, \mathbf{G} \leftarrow \mathcal{U}(\mathbb{Z}_d^{k \times m})\}$ is randomly and uniformly distributed over $\mathbb{Z}_d^m$ (see Lemma 2.3.1) . Also there exist at most $\left(\frac{2d^{1-k/m}}{4}\right)^m = \frac{d^{m-k}}{2^m}$ vectors $\mathbf{v} \in \mathbb{Z}_d^m$ such that $\|\mathbf{v}\|_\infty < \frac{d^{1-k/m}}{4}$. Therefore

$$\Pr_{\mathbf{G}}\left[\, \|\mathbf{G}^T\mathbf{x}\|_\infty < \frac{d^{1-k/m}}{4} \,\right] \leq \frac{d^{m-k}/2^m}{d^m} = d^{-k}/2^m.$$

Taking the union bound over all $\mathbf{x} \in \mathbb{Z}_d^k$, we finally get that

$$\Pr_{\mathbf{G}}\left[\, \lambda_1^\infty(\Lambda_d(\mathbf{G})) < \frac{d^{1-k/m}}{4} \,\right] \leq 1/2^m. \qquad \square$$

Let Good be the set of matrices $\mathbf{G} \in \mathbb{Z}_d^{k \times m}$ whose columns generate $\mathbb{Z}_d^k$ and for which $\lambda_1^\infty(\Lambda_d(\mathbf{G})) \geq \frac{d^{1-k/m}}{4}$. If $\mathbf{G} \in \mathsf{Good}$, we can use Proposition 2.5.1 to bound the smoothing parameter of $\Lambda_d^\perp(\mathbf{G})$ as follows: for all $\omega(\sqrt{\log m}) = \omega(\sqrt{\log n})$ functions, there exists $\epsilon(m) = \mathsf{negl}(m) = \mathsf{negl}(n)$ such that

$$\eta_\epsilon(\Lambda_d^\perp(\mathbf{G})) \leq \frac{\omega(\sqrt{\log m})}{\lambda_1^\infty(\Lambda_d^\perp(\mathbf{G})^*)} = \frac{d \cdot \omega(\sqrt{\log m})}{\lambda_1^\infty(\Lambda_d(\mathbf{G}))} \leq d^{k/m} \cdot \omega(\sqrt{\log m}) \leq r.$$

The penultimate inequality stems from the fact that $\mathbf{G} \in \mathsf{Good}$ while the last inequality holds by the hypothesis that $d < (r/\omega(\sqrt{\log n}\,))^{m/k}$. We can then use Lemma 2.5.2 to conclude that if $\mathbf{G} \in \mathsf{Good}$ and $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m, r}$ then $\mathbf{G}\mathbf{x}$ is within $2 \cdot \epsilon(n) = \mathsf{negl}(n)$ statistical distance from uniform over $\mathbb{Z}_d^k$. The proof of Lemma 4.4.6 then follows by observing that a random matrix $\mathbf{G} \in \mathbb{Z}_d^{k \times m} \in \mathsf{Good}$ except with negligible probability. Indeed,

$$\begin{aligned}
\Pr[\mathbf{G} \notin \mathsf{Good}] \;&\leq\; \Pr\left[\, \lambda_1^\infty(\Lambda_d(\mathbf{G})) < \frac{d^{1-k/m}}{4} \,\right] \\
&\quad + \Pr[\mathbf{G}\text{'s columns don't generate } \mathbb{Z}_d^k] \\
&\leq\; \frac{1}{2^m} + \frac{1}{p_{\min}^{m-k-1}} = \mathsf{negl}(n)
\end{aligned}$$

where $p_{\min}$ is the smallest prime factor of d and $m - k = \omega(\log n)$ by assumption. For the last inequality, we used Theorem 5.3.1, to bound the probability the columns of $\mathbf{G}$ do not generate $\mathbb{Z}_d^k$. □

When the number of columns of $\mathbf{G}$ is much larger than the number of its rows, the restriction on the divisors of q can be dropped. The formal statement is given in the following corollary which follows immediately from Lemma 4.4.6

Corollary 4.4.8. *For any positive integers k, m, q such that $m = \mathsf{poly}(n)$ and $m - k = \Omega(m)$ and for any $r \in \mathbb{R}^+$ such that $r = \omega(\log m)^{\omega(1)}$, if $\mathsf{Knap}[\mathbb{Z}_q^k, \mathcal{D}_{\mathbb{Z}^m, r}]$ is uninvertible, then it is also pseudorandom.*

Chapter 4 is, in part, a reprint of the paper "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions" [115] co-authored with Daniele Micciancio, published in the proceedings of the 31st Annual Cryptology Conference (CRYPTO 2011). The dissertation author was the primary investigator and author of this paper.

Chapter 5

Sample Preserving Search to Decision Reductions for LWE

Overview of the chapter. In this chapter, we show how our results for knapsack functions (Chapter 4) imply similar search to decision reductions for the Learning With Errors (LWE) problem. We start in Section 5.1 with a non technical overview of our results. We review related work in Section 5.2. Section 5.3 describes how inverting (resp. distinguishing) LWE can be seen as the dual problem of inverting (resp. distinguishing) a specific knapsack family with underlying group and input distribution related to the parameters of the LWE function. We then use this duality along with the results from Chapter 4, to present explicit parameter sets for which search LWE can be reduced to decision LWE in a sample preserving way (Section 5.4).

5.1 Results

The Learning With Errors (LWE) Problem. 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 where n, m, q are positive integers and χ a distribution over $\mathbb{Z}_q$. For

any $q, n \in \mathbb{N}$, $\mathbf{s} \in \mathbb{Z}_q^n$, and χ, define the distribution

$$\mathcal{A}_{\mathbf{s},\chi} = \{(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e) \mid \mathbf{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n), e \leftarrow \chi\}. \tag{5.1}$$

The (search) 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. Often, it is more convenient to work with the matrix notation of LWE (or DLWE). 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 $\mathbf{e} \leftarrow \chi^m$. Notice that once the secret $\mathbf{s}$ has been recovered, one can also recover the error vector $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{s}$ and vice versa. So, we can equivalently define (search) 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 5.1.1. *Let n, m, q be positive integers and χ a probability distribution over $\mathbb{Z}_q$.* $\mathrm{LWE}(n, m, q, \chi)$ *is 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 $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$. Under this notation, it can be shown that $\mathrm{LWE}(n, m, q, \chi)$ is 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 χ as the LWE error $\mathbf{e}$. This duality is by no means new, and has been noticed and used in different settings [152, 114]. Here we observe that the duality holds both for the search and decision variants of both problems, which, combined with the results from Chapter 4, can be readily translated into corresponding *sample preserving* search to decision reductions for LWE.[1] As a

[1] We note that there exists a *direct* (i.e., without exploiting the duality with the knapsack function) sample-preserving reduction between search LWE and its decision variant. Nevertheless, we choose to use the aforementioned duality for the following reasons: i) Our results for

direct 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 [25, 9, 90].
- Prime modulus q and *any polynomially bounded* error distribution.
- Prime power modulus $q = p^e$ (for any prime $p = \mathsf{poly}(n)$) and arbitrary input distribution over $\mathbb{Z}_p = \{-(p-1)/2, \ldots, (p-1)/2\}$.
- Prime power modulus $q = p^e$ (for any prime $p = \mathsf{poly}(n)$) and *uniform* error distribution over $\mathbb{Z}_{p^d}$ for some $d = O(\log_p n)$.

We emphasize that, besides being powerful and general, another distinctive characteristic of our reductions is that they are *sample preserving*: the pseudorandomness of LWE holds, provided the same problem is computationally hard to solve in its search version with the *same* number of samples.

On the significance of sample preserving reductions. On the theoretical side, cryptography based on LWE is supported by deep worst-case/average-case connections [139, 129, 31], showing that any algorithm that solves LWE (on the average) can be efficiently converted into an 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 [105]. 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

knapsack functions are much more general. Indeed, the LWE function family can be seen as the dual of a 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, reducing search LWE to decision LWE directly is not any less technical than reducing it indirectly (using the duality with knapsack families). iii) There exist parameters for which the indirect reduction is sample-preserving whereas the direct one is not. One such example is when q is *superpolynomial* but the noise is *polynomially bounded*.

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 [113, 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 [118, 145, 101, 6, 103] follows this approach, and investigates experimentally the concrete hardness of solving LWE on the average.

Even though theoretical results based on worst-case lattice problems are fairly insensitive to the number of samples used in the LWE instance, this number becomes more important when considering concrete attacks on the average-case hardness of LWE. For instance, recent algorithmic results [10] 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 [118]. However, LWE-based encryption schemes (e.g., see [101]) 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.

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 [63, 8, 141], but at the cost of increasing the 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 [10]. 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.

5.2 Related Work

The first search to decision reduction for LWE was given by Regev [139] for prime modulus q and (discretized) gaussian error distribution. Using the reduction, Regev presented the first public-key encryption scheme that is IND-CPA secure assuming the worst-case hardness of certain lattice problems. Applebaum *et al.* [8] extended the reduction to prime power modulus and any error with small (polynomially bounded) range. Peikert [129] was the first to extend the reduction to large (non polynomially bounded) noise for moduli with small (polynomially bounded) factors. This reduction was further improved in [116, 97, 31]. For LPN, a similar search to decision reduction was already given in [9] and [90].

Comparison With Our Work

As we have already mentioned, with the exception of [9], all other reductions are not sample preserving[2]. We remark, however, that previous results are often phrased as reductions from solving the LWE search problem *with high probability*, to solving the LWE decision problem *with 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 [139, 129, 31] 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 com-

[2]In fact, some of the search to decision reductions presented in [116, 97] are not even *noise* preserving.

putationally hard on average. Under the standard conjecture that search LWE is a *strong one-way function*, the results in this thesis 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 the sample complexity through amplification, and lead to quantitatively uninteresting results.

Comparison of Techniques. At a high level, all previously known search to decision reductions [139, 8, 129, 90] work in two steps: First, they exploit the self-reducibility of LWE (with respect to the secret $\mathbf{s}$) and standard amplification techniques in order to transform a distinguisher with noticeable advantage for a noticeable fraction of secrets $\mathbf{s}$ to an "almost perfect" distinguisher for *arbitrary* secret $\mathbf{s}$. This step incurs a large (yet polynomial) multiplicative factor in the sample complexity of the reduction. In the second step, the value of each entry s_i is guessed and the perfect distinguisher is then invoked to either confirm or refute the guess with success probability almost 1. The latter step contributes a small factor to the total number of samples consumed and outputs (with probability almost 1) the correct secret $\mathbf{s}$.

Our reduction proceeds in a fundamentally different way: first, it transforms the LWE instance to a knapsack instance where the error vector $\mathbf{e}$ of LWE becomes the (unknown) input for knapsack (see Section 5.3). The two steps are then merged in one by viewing the recovery of $\mathbf{e}$ as a *list decoding* problem which we solve using the SFT algorithm from Section 2.6. More specifically, the (hypothetical) distinguisher is used to predict the inner product $\mathbf{er} \pmod q$ where q is the LWE modulus and $\mathbf{r} \in \mathbb{Z}_q^m$. Based on the guesses for several (polynomially many) vectors $\mathbf{r}$, the SFT algorithm narrows down the search space for $\mathbf{e}$ to a list L of small (polynomially bounded) size. The crucial difference is that we can use

an *imperfect* distinguisher *directly* (without amplifying its advantage first) for the list decoding problem. The distinguishing advantage comes into play only in the quality of the guess $\mathbf{er} \pmod q$, which in turn determines the size of the list L as well as the probability the correct $\mathbf{e}$ is contained in L. By using an imperfect distinguisher, the need for advantage amplification, which accounted for the largest multiplicative factor in the sample complexity of previous reductions, is completely removed. As an additional benefit of using list decoding, we can recover $\mathbf{e}$ *entirely* (as opposed to one coordinate at a time) thus completely eliminating the need for more samples.

We remark that this fundamental difference (partially) explains why our approach is unlikely 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 all (non sample preserving) known search to decision reductions for large noise [129, 116, 97] rely heavily on a Chinese Reminder Theorem (CRT) approach: using a *perfect* 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 by first forming the entire secret using CRT and then verifying that the result is the target LWE secret. Thus, one has to solve a superpolynomial number of CRT instances before recovering the correct value of the secret.

5.3 Duality Between LWE and Knapsack Functions over Vector Groups

Recall that the $\mathrm{LWE}(n, m, q, \chi)$ is the problem of recovering $\mathbf{s} \in \mathbb{Z}_q^n$ given m samples from distribution $\mathcal{A}_{\mathbf{s},\chi}$ defined in (5.1). Likewise, in $\mathrm{DLWE}(n, m, q, \chi)$ one

is given m samples drawn (independently at random) either from $\mathcal{A}_{\mathbf{s},\chi}$ (for some secret $\mathbf{s}$) or from $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ and the goal is to tell the two distributions apart with noticeable probability.

In this chapter, we are interested in reductions from LWE to DLWE that *preserve all* the parameters n, m, q, χ, including the *number of samples* m. Since $\text{LWE}(n, m, q, \chi)$ is not a knapsack function family, in order to apply the results from Sections 4.3 and 4.4, we exploit the duality between the LWE problem and an associated knapsack family described in the following lemmas. Even though the aforementioned duality has been noticed before [152, 114, 101], to our knowledge, no detailed description of the steps that establish it has appeared in the literature. We start by proving some supporting lemmas in Section 5.3.1 and then describe both directions of the duality in Sections 5.3.2 and 5.3.3 respectively.

5.3.1 Supporting Lemmas

For a matrix $\mathbf{A} \in R^{m \times n}$ where R is a commutative ring, we use $\mathsf{row}(\mathbf{A}) = \{\mathbf{r}_1, ..., \mathbf{r}_m\}$ and $\mathsf{col}(\mathbf{A}) = \{\mathbf{c}_1, ..., \mathbf{c}_n\}$ denote the sets of the rows and columns of $\mathbf{A}$ respectively. We use $\mathsf{span}(\mathsf{row}(\mathbf{A}))$ for the set generated by considering R-linear combinations of the rows of $\mathbf{A}$, that is

$$\mathsf{span}(\mathsf{row}(\mathbf{A})) = \{\mathbf{y} \in R^n \colon \exists \mathbf{r} \in \mathbb{R}^m \text{ such that } \mathbf{y} = \mathbf{r}\mathbf{A}\}.$$

If $\mathsf{span}(\mathsf{row}(\mathbf{A})) = R^n$ we simply say that $\mathsf{row}(\mathbf{A})$ *spans* R^n. Likewise, $\mathsf{span}(\mathsf{col}(\mathbf{A}))$ is the set generated by considering R-linear combinations of the columns of $\mathbf{A}$. For a set $T = \{\mathbf{t}_1, ..., \mathbf{t}_k\}$ where $\mathbf{t}_i \in R^m$, T is said to be linearly independent if $\sum_{i=1}^k x_i \mathbf{t}_i = 0_R$ implies that $x_1 = ... = x_k = 0_R$ where 0_R is the identity element of R (with respect to addition). Below we focus on the ring $\mathbb{Z}_q$ for some (possibly composite) positive integer q. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, and a divisor d of q, we let $\mathbf{A}_d$ be the (projected) matrix derived by reducing all entries of $\mathbf{A}$ modulo d, i.e $[\mathbf{A}_d]_{ij} = [\mathbf{A}]_{ij} \pmod{d}$. Theorem 5.3.1 gives a lower bound on the probability the rows of a matrix $\mathbf{A}$ with uniform and random entries from $\mathbb{Z}_q$ span $\mathbb{Z}_q^n$.

Theorem 5.3.1. *Let $m, n \in \mathbb{Z}$ with $m \geq n$ and $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ where $q = \Pi_{i=1}^{\ell} p_i^{e_i}$.*

Then

$$\Pr\left[\ \mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_q^n\ \right] \geq 1 - \frac{1}{p_{\min}^{m-n-1}}$$

$p_{\min}$ *is the minimum prime factor of* q *and the probability is taken over the entries of* $\mathbf{A}$. *In particular, if* $m-n = \omega(\log n)$, *then* $\Pr\left[\ \mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_q^n\ \right] \geq 1 - negl(n)$.

Theorem 5.3.1 follows from the sequence of Lemmas given below.

Lemma 5.3.2. *Let* $\mathbf{A} \in \mathbb{Z}_p^{m\times n}$ *where* $m, n \in \mathbb{Z}$ *with* $m \geq n$ *and* p *is prime. Then* $\mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_p^n$ *if and only if* $\mathsf{col}\,(\mathbf{A})$ *are linearly independent.*

Proof.
($\Rightarrow$) Let $\mathbf{Ax} = \mathbf{0} \in \mathbb{Z}_p^n$. Since $\mathsf{row}\,(\mathbf{A})$ generates $\mathbb{Z}_p^n$, for all $i = 1, ...n$ there exists $\mathbf{r} \in \mathbb{Z}_p^m$ such that $\mathbf{rA} = \mathbf{e}_i$ where $\mathbf{e}_i = (0, ..., 0, 1, 0, ..., 0)$ is the m-dimensional vector with its i-th entry set to 1 and the remaining set to 0. Left multiplying $\mathbf{Ax}$ with $\mathbf{r}$ gives $x_i = 0$ for all $i = 1, ..., n$ which implies that $\mathsf{col}\,(\mathbf{A})$ are linearly independent.
($\Leftarrow$) Since p is prime (i.e. $\mathbb{Z}_p$ is a field), the row rank of $\mathbf{A}$ is the same as its column rank, that is row rank=n and therefore $\mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_p^n$. □

The following lemma considers the case where q is a power of a prime.

Lemma 5.3.3. *Let* $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ *where* $m, n \in \mathbb{Z}$ *with* $m \geq n$ *and* $q = p^e$ *for some prime* p. *Then* $\mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_q^n$ *if and only if* $\mathsf{span}(\mathsf{row}\,(\mathbf{A}_p)) = \mathbb{Z}_p^n$.

Proof.
($\Rightarrow$) Let $\mathbf{z} \in \mathbb{Z}_p^n$. Since $\mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_q^n$, there exists $\mathbf{r} \in \mathbb{Z}_q^m$ such that $\mathbf{rA} \equiv \mathbf{z} \pmod q$. Let $\mathbf{r} = \mathbf{r}_0 + p\mathbf{r}_1$ where $\mathbf{r}_0$ has all its entries in $\mathbb{Z}_p$. Write also $\mathbf{A}$ as $\mathbf{A} = p\bar{\mathbf{A}} + \mathbf{A}_p$. Then

$$\mathbf{r}_0\mathbf{A}_p = (\mathbf{r} - p\mathbf{r}_1)(\mathbf{A} - p\bar{\mathbf{A}}) \equiv \mathbf{rA} \equiv \mathbf{z} \pmod p.$$

($\Leftarrow$) We prove the inverse direction for $q = p^i$ for every $i \geq 1$ using induction. The basis step ($i = 1$) holds by the hypothesis. Now assume claim holds for p^j $\forall j \leq i$. Let $\mathbf{z} \in \mathbb{Z}_{p^{i+1}}^n$. Write $\mathbf{z} = \mathbf{z}_0 + p^i\mathbf{z}_1$ where $\mathbf{z}_0 \in \mathbb{Z}_{p^i}^n$ and $\mathbf{z}_1 \in \mathbb{Z}_p^n$. $\mathbf{A} \in \mathbb{Z}_{p^{i+1}}^{m\times n}$ can be written as $\mathbf{A} = \mathbf{A}_0 + p^i\mathbf{A}_1$ (with $[\mathbf{A}_0]_{ij} \in \mathbb{Z}_{p^i}$) and $\mathbf{A}_0$ can be written as

$\mathbf{A}_0 = \mathbf{A}_p + p\bar{\mathbf{A}}$ (with $[\mathbf{A}_p]_{ij} \in \mathbb{Z}_p$). By induction $\exists \mathbf{r}_0 \in \mathbb{Z}_{p^i}^m$ and $\mathbf{y} \in \mathbb{Z}^n$ such that $\mathbf{r}_0\mathbf{A}_0 + p^i\mathbf{y} = \mathbf{z}_0$. Also there exists $\mathbf{r}_1 \in \mathbb{Z}_p^m$ such that

$$\begin{aligned} \mathbf{r}_1\mathbf{A}_p \equiv \mathbf{z}_1 + \mathbf{y} - \mathbf{r}_0\mathbf{A}_1 \pmod{p} \quad &\Rightarrow \quad \mathbf{r}_1(\mathbf{A}_0 - p\bar{\mathbf{A}}) \equiv \mathbf{z}_1 + \mathbf{y} - \mathbf{r}_0\mathbf{A}_1 \pmod{p} \\ &\Rightarrow \quad \mathbf{r}_1\mathbf{A}_0 = \mathbf{z}_1 + \mathbf{y} - \mathbf{r}_0\mathbf{A}_1 + p\mathbf{w} \end{aligned}$$

for some $\mathbf{w} \in \mathbb{Z}^n$. Now let $\mathbf{r} = \mathbf{r}_0 + p^i\mathbf{r}_1$. Then

$$\begin{aligned} \mathbf{rA} &= (\mathbf{r}_0 + p^i\mathbf{r}_1)(\mathbf{A}_0 + p^i\mathbf{A}_1) = \mathbf{r}_0\mathbf{A} + p^i(\mathbf{r}_1\mathbf{A}_0 + \mathbf{r}_0\mathbf{A}_1) + p^{2i}\mathbf{r}_1\mathbf{A}_1 \\ &\equiv \mathbf{z}_0 + p^i(\mathbf{r}_1\mathbf{A}_0 + \mathbf{r}_0\mathbf{A}_1 - \mathbf{y}) \pmod{p^{i+1}} \\ &\equiv \mathbf{z}_0 + p^i\mathbf{z}_1 \pmod{p^{i+1}} \end{aligned}$$

which concludes the proof. □

Lemma 5.3.4. *Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ where $m, n \in \mathbb{Z}$ with $m \geq n$ and $q = M_1 \cdot M_2$ for positive integers M_1, M_2 with $\gcd(M_1, M_2) = 1$. Then $\mathsf{span}(\mathsf{row}(\mathbf{A})) = \mathbb{Z}_q^n$ if and only if $\mathsf{span}(\mathsf{row}(\mathbf{A}_{M_1})) = \mathbb{Z}_{M_1}^n$ and $\mathsf{span}(\mathsf{row}(\mathbf{A}_{M_2})) = \mathbb{Z}_{M_2}^n$.*

Proof. The proof is a straightforward application of the Chinese Remainder Theorem. □

Lemma 5.3.5 bounds the probability n random vectors from $\mathbb{Z}_p^m$ are linearly dependent.

Lemma 5.3.5. *Let $m, n \in \mathbb{Z}$ with $m \geq n$ and p prime. Let $\mathbf{v}_1, ..., \mathbf{v}_n$ such that $\mathbf{v}_i \leftarrow \mathcal{U}(\mathbb{Z}_p^m)$. Then*

$$\Pr[\, \mathbf{v}_1, ..., \mathbf{v}_n \text{ are linearly independent} \,] \geq 1 - \frac{1}{(p-1)p^{m-n}}.$$

Proof. For $\mathbf{v}_i$, $\Pr[\, \mathbf{v}_i \in \mathsf{span}(\mathbf{v}_1, ..., \mathbf{v}_{i-1}) \,] \leq \frac{1}{p^{m-i+1}}$. Applying the union bound, we get

$$\begin{aligned} \Pr[\, \mathbf{v}_1, ..., \mathbf{v}_n \text{ linearly dependent} \,] &\leq \sum_{i=1}^{n} \Pr[\, \mathbf{v}_i \in \mathsf{span}(\mathbf{v}_1, ..., \mathbf{v}_{i-1}) \,] \\ &\leq \sum_{i=1}^{n} \frac{1}{p^{m-i+1}} \leq \frac{1}{(p-1)p^{m-n}}. \end{aligned}$$

□

Proof. (*of Theorem 5.3.1*). By lemmas 5.3.2, 5.3.3 and 5.3.4, $\mathsf{span}(\mathsf{row}\,(\mathbf{A})) = \mathbb{Z}_q^n$ if and only if the columns of the (projected) matrices $\mathbf{A}_{p_i}$ are linearly independent for every $i = 1, ..., \ell$. Since $p_i \,|\, q$, $\forall i = 1, ..., \ell$, entries of $\mathbf{A}_{p_i}$ are uniformly, randomly and independently distributed over $\mathbb{Z}_{p_i}$ for all $i = 1, ..., \ell$. Let p_1 be the smallest prime factor of q. Then, using Lemma 5.3.5 and the union bound, we get

$$\begin{aligned}
\Pr\left[\, \exists\, i \colon \mathsf{span}(\mathsf{row}\,(A_{p_i})) \subset \mathbb{Z}_{p_i}^n \,\right] &\leq \sum_{i=1}^{\ell} \frac{1}{(p_i - 1)p_i^{m-n}} \\
&= \frac{1}{p_1^{m-n-1}} \left(\frac{1}{p_1(p_1-1)} + \sum_{i=2}^{l} \frac{p_1^{m-n-1}}{(p_i-1)p_i^{m-n}} \right) \\
&\leq \frac{1}{p_1^{m-n-1}} \left(\frac{1}{2} + \sum_{i=2}^{l} \frac{1}{(p_i-1)p_i} \right) \leq \frac{1}{p_1^{m-n-1}}
\end{aligned}$$

where in the last inequality we used the fact that $\sum_{i=2}^{l} \frac{1}{(p_i-1)p_i}$ can be upper bound (using elementary arithmetic) by $1/2$. □

5.3.2 From LWE to Knapsack

The following lemma states that an inverter for a specific knapsack family can be turned into an inverter for the LWE function with only a negligible loss in the success probability.

Lemma 5.3.6. *For any*[3] *positive integers* $n, m \geq n + \omega(\log n), q$ *and distribution* χ *over* $\mathbb{Z}_q$*, 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* $\mathsf{Knap}[\mathbb{Z}_q^{m-n}, \chi^m]$ *with probability* $\epsilon' = \epsilon + \mathsf{negl}(n)$.

Proof. The transformation from an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$ into an equivalent knapsack instance requires that the matrix $\mathbf{A}$ be nonsingular, i.e., the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^n$. By Theorem 5.3.1, 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.

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

Consider now the set of all (row) vectors $\mathbf{g} \in \mathbb{Z}_q^m$ such that $\mathbf{gA} = 0 \pmod{q}$. 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 $\mathbf{U} \in \mathbb{Z}_q^{(m-n)\times(m-n)}$. It can be checked that, if $\mathbf{A}$ is chosen uniformly 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 $\mathbb{G} = \mathbb{Z}_q^{m-n}$. The reduction now works as follows: On input an LWE instance $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \in \mathbb{Z}^{m\times n} \times \mathbb{Z}^m$, it computes $\mathbf{G} \in \mathbb{Z}^{(m-n)\times m}$ as described above and outputs $(\mathbf{G}, \mathbf{y})$ where $\mathbf{y} = \mathbf{Gb} = \mathbf{GAs} + \mathbf{Ge} = \mathbf{Ge}$. Notice that the distribution $(\mathbf{G}, \mathbf{y})$ is *statistically close* to a *random instance* of the knapsack problem with group $\mathbb{G} = \mathbb{Z}_q^{m-n}$ and input distribution $\mathcal{X} = \chi^m$ which completes the proof. □

5.3.3 From Knapsack to LWE

Lemma 5.3.7 is essentially the inverse of lemma 5.3.6 stating that a distinguisher for LWE can be used to distinguish the outputs of a knapsack family from random ones with only a negligible loss in the advantage. Even though the reduction is phrased in terms of decisional problems, we remark that reductions exist also in the directions opposite to those described in Lemma 5.3.6 and Lemma 5.3.7, but this is all we need here.

Lemma 5.3.7. *For any positive integers $n, m \geq n + \omega(\log n), q$ and distribution χ over $\mathbb{Z}_q$, there is a polynomial time reduction from the problem of distinguishing $\mathcal{F}(\mathsf{Knap}[\mathbb{Z}_q^{m-n}, \chi^m])$ 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. The distinguisher for the knapsack function is obtained similarly, transforming a knapsack instance into a corresponding LWE one. This transformation essentially reverses the steps taken to transform LWE into knapsack. The input is a pair $(\mathbf{G}, \mathbf{y})$ where $\mathbf{y}$ is either $\mathbf{Ge}$ (for some $\mathbf{e} \leftarrow \chi^m$) or $\mathcal{U}(\mathbb{Z}_q^{m-n})$. As before,

we can assume without loss of generality (up to negligible statistical error, see Theorem 5.3.1) 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 $\mathbf{G}\mathbf{a}' = \mathbf{0} \pmod q$. Similar to the proof of Lemma 5.3.6, 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 $\mathbf{A}$. Let $\mathbf{r} \in \mathbb{Z}_q^m$ be an arbitrary solution to the equation $\mathbf{G}\mathbf{r} = \mathbf{y} \pmod q$ (since $\mathsf{span}(\mathsf{col}(\mathbf{G})) = \mathbb{Z}_q^{m-n}$ such a vector $\mathbf{r}$ always exists). The reduction then works as follows: On input $(\mathbf{G}, \mathbf{y}) \in \mathbb{Z}_q^{(m-n) \times m} \times \mathbb{Z}_q^m$, it computes $\mathbf{A}$ as described above, samples $\mathbf{s}' \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and outputs $(\mathbf{A}, \mathbf{A}\mathbf{s}' + \mathbf{r})$. Assume first $\mathbf{y} = \mathbf{G}\mathbf{e}$. Consider an alternative way of sampling $\mathbf{s}'$ where we first sample $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and then set $\mathbf{s}' = \mathbf{s} - \mathbf{z}$ where $\mathbf{z}$ is an arbitrary solution to the equation $\mathbf{A}\mathbf{z} = \mathbf{r} - \mathbf{e}$ (notice that $\mathbf{G}(\mathbf{r} - \mathbf{e}) = \mathbf{y} - \mathbf{y} = \mathbf{0}$, that is, $\mathbf{r} - \mathbf{e}$ is in the nullspace of $\mathbf{G}$ and hence there exists $\mathbf{z} \in \mathbb{Z}_q^n$ such that $\mathbf{A}\mathbf{z} = \mathbf{r} - \mathbf{e} \pmod q$). Clearly $\mathbf{s}'$ is uniformly and randomly distributed over $\mathbb{Z}_q^n$. Also $\mathbf{A}\mathbf{s}' + \mathbf{r} = \mathbf{A}(\mathbf{s} - \mathbf{z}) + \mathbf{r} = \mathbf{A}\mathbf{s} + \mathbf{e}$ which is exactly the LWE distribution with error vector $\mathbf{e}$ being the input to the knapsack function. If on the other hand, $\mathbf{y} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$, then $\mathbf{A}\mathbf{s}' + \mathbf{r}$ is uniformly and randomly distributed over $\mathbb{Z}_q^m$ since $\mathbf{s}' \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $\mathbf{r}$ is a solution to $\mathbf{G}\mathbf{r} = \mathbf{y}$ for uniform and random $\mathbf{y} \in \mathbb{Z}_q^m$. □

5.4 Applications to LWE

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 described in Lemma 5.3.6 and Lemma 5.3.7 with the results from Sections 4.3 and 4.4.1 on $\mathsf{Knap}[\mathbb{Z}_q^{m-n}, \chi^m]$. 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. We focus on parameters that seem attractive from an application viewpoint but emphasize that our results can be applied to much broader sets of parameters. Throughout, it is assumed that $m \geq n + \omega(\log n)$.

Proposition 5.4.1. *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 $q = p^e$ and error distribution χ which is uniform over $\mathbb{Z}_{p^i}$ for some $i < e$ such that $p^i = \mathsf{poly}(n)$.

(v) Any modulus q and Gaussian error distribution $\chi = \mathcal{D}_{\mathbb{Z},r}$ where $\omega(\log n)^{\omega(1)} \leq r \leq \mathsf{poly}(n)$ and $m = O(n)$.

Proof. The proof for all cases follows easily by combining Lemmas 5.3.6 and 5.3.7 with the results for bounded knapsack families from Section 4.4.1. More specifically, (i), (ii) and (iii) are direct applications of Lemma 4.4.2, case (iv) is immediate from Corollary 4.4.5 and case (v) follows from Corollary 4.4.8. □

Remark 5.4.2. *Case (i) provides a sample preserving search to decision reduction for* LPN*. Such a reduction was already given in [9]. In contrast, other reductions appearing in the literature [25, 90] 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 [139] who also presented a (non sample preserving) search to decision reduction. Case (iii) provides a sample preserving version of the reduction proved in [8]. Finally, the search to decision reduction for LWE with modulus and noise distribution as in cases (iv) and (v) appear to be new; no such (even non-sample preserving) reductions have 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.*

Chapter 5 is, in part, a reprint of the paper "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions" [115] co-authored with Daniele Micciancio, published in the proceedings of the 31st Annual Cryptology Conference (CRYPTO 2011). The dissertation author was the primary investigator and author of this paper.

Chapter 6

An Efficient Authentication Protocol Secure Against Active Attacks from Learning Parity with Noise

Overview of the chapter. The main result of this chapter is an efficient, 3-round, actively secure authentication protocol based on the hardness of the Learning Parity with Noise (LPN) problem. Section 6.1 contains an overview of our results. A brief review of related work as well as comparison with our work can be found in Section 6.2. We provide a formal description of our security model and relevant definitions in Section 6.3. In Section 6.4 we present a generic construction of an actively secure authentication protocol based on any weak message authentication code. Finally, we present an efficient instantiation of the generic protocol from LPN (Section 6.5).

6.1 Results

The main focus of this chapter is the construction of symmetric authentication protocols that are secure under active attacks. Roughly, active attacks involve

adversaries that run in two phases. In the first phase, the adversary gets to see multiple honest prover-verifier interactions, much like in a passive attack, but can also arbitrarily interact with multiple instances of the prover in an attempt to gain information about the key. In the second phase, the adversary tries to convince the verifier that he knows the key. We remark that (strictly) stronger notions than active security do exist. For instance, Man-in-the-Middle (MIM) security considers even stronger adversaries that can interact (arbitrarily) with the prover and the verifier concurrently. Even though adopting a conservative viewpoint when defining security goals has several benefits, it is an unavoidable fact that existing protocols with attractive implementation features fall short of achieving the strongest notions;[1] our goal is to precisely address what these protocols achieve instead.

For achieving active security, we propose a new *generic* construction that yields an efficient 3-round authentication protocol based on any *weak* MAC, i.e., a MAC which can be evaluated on *random* messages and which must be unforgeable on fresh, *random* messages. This is the weakest generic assumption on which such a protocol can be based, with previous generic constructions being either from stronger MACs or from a Weak PRF [48]. Given such a MAC, our three-round protocol, which we call **DM** (for Double Mac), is extremely simple. The secret key of the protocol consists of two keys K_1, K_2 for the underlying MAC. In the first round, the prover sends a random message r_1 to the verifier, which replies with $(\mathrm{MAC}_{K_1}(r_1), r_2)$, for a random message r_2, in the second round. The prover, upon receiving (τ_1, r_2), subsequently checks whether τ_1 happens to be a valid tag for r_1, and if so, sends $\mathrm{MAC}_{K_2}(r_2)$ back to the verifier, which finally accepts if and only if it receives a valid tag τ_2 for r_2 under key K_2.

As the most compelling application of our new construction, we provide an efficient instantiation from the Learning Parity with Noise Problem which, besides being simpler, also achieves a better security-efficiency trade-off when compared to previous LPN-based actively secure protocols [87, 94]. We provide a thorough

[1] This phenomenon is especially pronounced in the case of RFID protocols where requiring very strong cryptography drives the manufacturing of RFID tags to prohibitively high costs and slows down the widespread deployment of the technology.

qualitative and quantitative comparison in Sections 6.2 and 6.5 respectively.

6.2 Related Work

Theoretical research on authentication protocols has been initially concerned with the public-key setting, where a prover $\mathcal{P}$ in possession of a public/secret-key pair (pk, sk) wishes to prove its identity to a verifier $\mathcal{V}$ who only knows pk (such protocols have often been called *identification* protocols). Starting from the seminal work of Fiat and Shamir [58], a long series of protocols have been proposed (among others, cf. e.g. [73, 147, 125, 155]) mostly leveraging techniques from zero-knowledge proofs [72, 57].

Research on symmetric authentication has mainly been concerned with two important aspects: On one hand, a vast body of literature has focused on privacy concerns related to RFID protocols [12, 46, 13, 88, 158, 44] and especially traceability of tags. We view privacy as an orthogonal goal and do not consider it in this work.

The other aspect that has attracted broad interest is security, leading to a long history of security notions and corresponding protocols. Most relevant to active security and our work is a security notion introduced by Gilbert *et al* [65] who considered an intermediate model (*aka* GRS-MIM model) in which an adversary can interact with both the tag and the reader in the first phase of the attack, but can only modify messages from the reader. Even though security in the GRS-MIM is strictly stronger than active security, the protocols that are known to achieve the former notion [65] are either inefficient ($RANDOM - HB^{\#}$) or based on assumptions that are not well studied ($HB^{\#}$). It has also been questioned whether there exist real-world attack scenarios in which an attacker can modify messages from the reader but not from the tag – and in the full-fledged MIM case, none of the protocols from [65] is secure [127].

LPN-based Protocols. Thanks to its extremely simple structure (involving only simple operations over bits or binary vectors), LPN has turned out to be a very promising candidate for the construction of efficient authentication proto-

cols. Below we recall the two previously known LPN-based protocols which are secure under active attacks and point out their advantages and disadvantages when compared to ours.

The HB^+ is a simple and elegant 3-round (commit-challenge-response) protocol proposed by Juels and Weiss [87]. Its main drawback is the lack of a "tight" security reduction to LPN. Current reductions [87, 90] assert that no active attacker with time complexity (roughly) t can break the security of HB^+ for key length $2n$ with probability larger than $\sqrt{\epsilon}$, assuming LPN is ϵ-hard for secret length n and complexity t. This is unsatisfactory. For example, if we have $t = 2^{40}$ and $\epsilon = 2^{-40}$, an adversary attacking 2^{20} independent instances of the protocol may break at least one of them spending overall effort $t' = 2^{60}$, which may still be feasible. Also, we point out that this loss is the inevitable result of using rewinding in the security reduction, and, at least from a theoretical perspective, that this makes it impossible to prove HB^+ secure against quantum attackers (based on the quantum hardness of LPN).

Kiltz *et al* [94] presented a two-round protocol based on subspace-LPN, a variant of LPN. While their protocol enjoys a tight reduction to LPN in terms of *advantage* ϵ, yet, due to looseness in the reduction from (standard) LPN to subspace-LPN, if we assume as above that LPN is ϵ-hard for secret-size n, for their protocol to be ϵ-secure too, even under the most optimistic instantiation of their parameters, their key size becomes larger than $4n$ bits and the communication complexity is larger than the one of HB^+.

Even more, the security of our LPN-based protocol scales significantly better than both HB^+ and the protocol of Kiltz *et al.* in the face of *multiple verification* attempts where, in the latter two protocols, an adversary essentially increases its success probability by a factor which is *linear* in the number of interactions with the verifier.

Of course, it is fair to note that a drawback of our protocol compared to existing challenge-response protocols [87, 90] is that it is not, in general, *strongly* actively secure, i.e., it does not remain secure against adversaries that are allowed multiple, alternating (yet non-overlapping), interactions with the prover and the

verifier, a clear advantage of existing challenge-response protocols. However, we point out that, to the best of our knowledge, strong active security was never considered prior to our work, hence indicating that (non-strong) active security is considered sufficient in many settings.

Other Related Work. We conclude this section by briefly mentioning two recent protocols proposed concurrently and independently of our work. Heyse *et al* [78] presented Lapin, a simple and elegant 2-round protocol that is secure against active attacks. The security of Lapin relies on the assumption that the Ring-LPN problem, a structured variant of the standard LPN problem, is hard. However, the hardness of Ring-LPN is much less understood[2] than the hardness of LPN and thus, given our current understanding of algorithmic attacks, any comparison with LPN-based protocols is hardly meaningful (see also a recent attack by Bernstein and Lange [21] which exploits the ring structure of Ring-LPN to drastically reduce the resources needed for an active attack).

Another interesting LPN-based protocol was very recently proposed by Lyubashevsky and Masny [104]. Their protocol is proved secure under MIM attacks. While the paper makes significant progress towards efficient protocols that meet the strongest security notion, their construction still suffers from certain drawbacks such as loose security reduction to LPN and lack of security proof for parallel executions of the protocol. So, in practical scenarios where MIM attacks are not a threat, our scheme remains a more suitable option.

6.3 Definitions and Security Model

In this Section we provide basic definitions for secret-key authentication protocols and describe formally a framework to model their security under active attacks. Our framework, which inherits heavily from that of Bellare and Rogaway [17], can be easily generalized to support various, fine-grained attack models (see

[2]Ring-LPN can be also seen as a special case of Ring-LWE [106] with modulus $q = 2$. However, unlike Ring-LWE, Ring-LPN is *not* backed by a worst-case/average-case connection with (ideal) lattices.

$\underline{(\mathcal{P}_1 \leftrightarrow \mathcal{P}_2)(x)}$:

$msg_0 \leftarrow \mathsf{start}$; $i \leftarrow 0$

$y_1, y_2 \leftarrow \perp$; $\sigma_1, \sigma_2 \leftarrow \varepsilon$

While $y_1 = \perp$ or $y_2 = \perp$ do

  If $i = 0 \mod 2$ and $y_1 = \perp$ then

    $(y_1, \sigma_1, msg_{i+1}) \xleftarrow{\$} \mathcal{P}_1(x, \sigma_1, msg_i)$

  Else if $y_2 = \perp$ then

    $(y_2, \sigma_2, msg_{i+1}) \xleftarrow{\$} \mathcal{P}_2(x, \sigma_2, msg_i)$

  $i \leftarrow i + 1$

Ret `true`

Figure 6.1: Pseudocode for an interactive two-party protocol.

[119] for the full description of the framework). However, we only present a restricted version of it here focusing on the security notion of interest, i.e., security against active attacks. Throughout the entire chapter, in our security definitions and proofs, we will often use games, as defined by Bellare and Rogaway [18], and adopt their computational model and notational conventions.

Algorithms and protocols. Throughout this section, we model stateful randomized algorithms as follows: A stateful algorithm $\mathcal{A}$ has an initial input, keeps a *state*, and processes messages. Formally, $\mathcal{A}$ is a randomized algorithm $\mathcal{A} : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to (\{0,1\}^* \cup \{\perp\}) \times \{0,1\}^* \times (\{0,1\}^* \cup \{\perp\})$, where $(y, \sigma', msg') \xleftarrow{\$} \mathcal{A}(x, \sigma, msg)$ means that starting from state σ, on initial input x, and upon receipt of message msg, $\mathcal{A}$ changes its internal state to σ', sends message msg' and, if $y \neq \perp$, terminates with output y. Here, $y = \varepsilon$ indicates termination without any output.

An *interactive two-party protocol*, is a pair $(\mathcal{P}_1, \mathcal{P}_2)$ of interactive algorithms, where exactly one of $\mathcal{P}_1$ and $\mathcal{P}_2$ accepts a special designated message start. (We assume that it is $\mathcal{P}_1$ in the following.) The protocol execution is defined via the procedure shown in Figure 6.1:

We say that $(\mathcal{P}_1, \mathcal{P}_2)$ is *well-formed* if the above procedure always terminates returning `true`. Moreover, it is an *r-round protocol* if $i = r + 1$ upon termination. We denote as $(y_1, y_2) \xleftarrow{\$} (\mathcal{P}_1 \leftrightarrow \mathcal{P}_2)(x)$ the process of sampling the outputs of $\mathcal{P}_1$ and $\mathcal{P}_2$ after an interaction. We also overload notation by writing $\text{Tran} \xleftarrow{\$} (\mathcal{P}_1 \leftrightarrow \mathcal{P}_2)(x)$ for the process of sampling the transcript of the interaction between $\mathcal{P}_1$ and $\mathcal{P}_2$, i.e., the sequence consisting of the messages $(msg_1, \ldots, msg_r)$ exchanged. Notice that $msg_0 = \mathsf{start}$ and the very last message are *not* part of the transcript.

Authentication protocols. A (secret-key) authentication protocol is a triple $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ such that $\mathcal{K}$ is a randomized *key generation algorithm* that generates a key K, while $\mathcal{P}$ and $\mathcal{V}$ are interactive algorithms, both taking as input a key K in the range of $\mathcal{K}$, and such that $(\mathcal{P}, \mathcal{V})$ is a well-formed interactive protocol. In addition, $\mathcal{P}$ always outputs ε, whereas $\mathcal{V}$ outputs a decision value $d \in \{\mathsf{A}, \mathsf{R}\}$. For any real value $\delta \in [0, 1]$, we say that the protocol Π is *δ-complete* (or has completeness δ) if $\Pr\left[K \xleftarrow{\$} \mathcal{K}, (\varepsilon, d) \xleftarrow{\$} (\mathcal{P} \leftrightarrow \mathcal{V})(K) : d = \mathsf{A} \right] \geq \delta$. We assume without loss of generality that the last message is sent from $\mathcal{P}$ to $\mathcal{V}$, which then terminates with a decision, and does not send any further messages.

Active security of authentication protocols. Let $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ be an authentication protocol. To model active security of Π, we consider adversaries that run in 2 phases (stages) and participate in the game $\text{AUTH}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}$ shown in Figure 6.2. The game $\text{AUTH}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}$ starts by sampling a key $K \xleftarrow{\$} \mathcal{K}$, and allows the attacker to arbitrarily interact with instances of the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ under key K, addressed via session ids *sid*s in $SID_{\mathcal{P}}$ and $SID_{\mathcal{V}}$, respectively, for two understood disjoint sets of integers $SID_{\mathcal{P}}, SID_{\mathcal{V}} \subset \mathbb{N}$. We remark that a session id *sid* characterizes an interaction between the adversary and an instance of $\mathcal{P}$ (or $\mathcal{V}$) and *not* between an instance of $\mathcal{P}$ and an instance of $\mathcal{V}$. Also, the same key K is shared across all instance $sid \in SID_{\mathcal{P}} \cup SID_{\mathcal{V}}$. The global variables $\mathsf{state}[sid]$, $\mathsf{decision}[sid]$ and $\mathsf{done}[sid]$ maintain information associated with each *sid*, i.e., the state of the corresponding instance, whether it has accepted an interaction (in case $sid \in SID_{\mathcal{V}}$) or whether it has terminated. The game consists of 2 *phases* and involves respective adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ where $\mathcal{A}_1$ can pass on arbitrary state

Game $\text{AUTH}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}$:

procedure main():

$K \xleftarrow{\$} \mathcal{K}$;

For all $sid \in \mathbb{N}$ do

  $\mathsf{state}[sid] \leftarrow \varepsilon$;

  $\mathsf{decision}[sid] \leftarrow \perp$;

  $\mathsf{done}[sid] \leftarrow \texttt{false}$;

$\sigma_1 \xleftarrow{\$} \mathcal{A}_1^{\mathbf{T},\mathbf{P}}$; // Phase 1

$\mathcal{A}_2^{\mathbf{V}}(\sigma_1)$; // Phase 2

If $\exists sid \in SID_{\mathcal{V}}$: $(\mathsf{decision}[sid] = \mathsf{A})$

  Ret true;

Ret false;

oracle $\mathbf{T}()$:

$\text{Tran} \xleftarrow{\$} (\mathcal{P} \leftrightarrow \mathcal{V})(K)$

Ret Tran

oracle $\mathbf{P}(sid, msg)$:

If $(sid \notin SID_{\mathcal{P}}) \vee \mathsf{done}[sid]$ Ret $\perp$

Else

  $(\mathsf{state}[sid], msg', y') \xleftarrow{\$} \mathcal{P}(K, \mathsf{state}[sid], msg)$

  If $y' \neq \perp$ then

    $\mathsf{done}[sid] \leftarrow \texttt{true}$

  Ret msg'

oracle $\mathbf{V}(sid, msg)$:

If $(sid \notin SID_{\mathcal{V}}) \vee \mathsf{done}[sid]$ Ret $\perp$

Else

  $(\mathsf{state}[sid], msg', y') \xleftarrow{\$} \mathcal{V}(K, \mathsf{state}[sid], msg)$

  If $y' \neq \perp$ then

    $\mathsf{done}[sid] \leftarrow \texttt{true}$; $\mathsf{decision}[sid] \leftarrow y'$

    Ret y'

  Ret msg'

Figure 6.2: General pseudocode description of Game $\text{AUTH}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}$ that defines security under active attacks. Here, $\Pi = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ is an authentication protocol and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a 2-phase adversary.

information to $\mathcal{A}_2$. Over the course of the entire attack, the adversary is granted access to the following oracles:

- The *prover oracle* $\mathbf{P}$ accepts queries of the form (sid, msg) where $sid \in SID_{\mathcal{P}}$ and $msg \in \{0,1\}^*$. Upon such a query, it runs $\mathcal{P}(K, \mathsf{state}[sid], msg)$, obtaining output (σ', msg', y). It then sets $\mathsf{state}[sid]$ to σ', and if $y' = \perp$, returns msg' to the adversary; otherwise it returns (y', msg'). In the latter case, $\mathbf{P}$ does not accept any further queries of the form $(sid, *)$ until the end of the current phase.
- The *verifier oracle* $\mathbf{V}$ operates as $\mathbf{P}$, using $\mathcal{V}$ instead of $\mathcal{P}$. In addition, upon terminating, i.e., when returning $(d, \perp)$ for $d \in \{\mathsf{A}, \mathsf{R}\}$ after a query (sid, msg), it sets $\mathsf{decision}[sid] \leftarrow d$.
- The *transcript oracle* $\mathbf{T}$ samples a transcript $\text{Tran} \xleftarrow{\$} (\mathcal{P} \leftrightarrow \mathcal{V})(K)$ and returns

it.

Specifically, in phase 1, $\mathcal{A}_1$ gets access to $\mathbf{P}$ and $\mathbf{T}$ while, in phase 2, it gains access to $\mathbf{V}$. To address the randomized nature of $\mathcal{P}$ and $\mathcal{V}$, we assume that each oracle has access to a fresh randomness source and that oracles associated with different *sid*s use fresh random coins each time they are invoked.

The AUTH game finally returns `true` if $\mathcal{A}_2$ manages to make the verifier accept in phase 2 for some *sid* (i.e., $\mathsf{decision}[sid] = \mathsf{A}$ for some $sid \in SID_{\mathcal{V}}$). It returns `false` otherwise. For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we say that $\mathcal{A}$ makes q_{P} queries to $\mathbf{P}$ (in phase 1) if the number of *distinct* $sid \in SID_{\mathcal{P}}$ that appear across all queries of the form (sid, msg) during phase 1 are q_{P}. q_{V} is defined similarly (for phase 2). Queries to $\mathbf{T}$ are not interactive and hence q_{T} is precisely the number of calls to $\mathbf{T}$ during phase 1. The $(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})$-*auth advantage* of $\mathcal{A}$ is defined as

$$\mathsf{Adv}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}(\mathcal{A}) = \Pr\left[\,(\mathrm{AUTH}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})})^{\mathcal{A}} \Rightarrow \mathtt{true}\,\right] .$$

Moreover, for all positive t and $q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}}$ we define

$$\mathsf{Adv}_{\Pi}^{((\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\}))\text{-auth}}(t, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}}) = \max_{\mathcal{A}}\{\mathsf{Adv}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}(\mathcal{A})\} .$$

The maximum here is over all adversaries $\mathcal{A}$ running in time t and making $q_{\mathsf{T}}, q_{\mathsf{P}}$ and q_{V} queries to the corresponding oracles (during the corresponding phase). Informally, we will say that a protocol Π is *actively secure* or $(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})$-secure if for all *efficient* adversaries $\mathcal{A}$, $\mathsf{Adv}_{\Pi}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}(\mathcal{A})$ is small.

6.4 Active Security Based on Random-Message / Random-Challenge Secure MACs

In this section, we present a generic 3-round protocol that is secure against active attacks. The security of our protocol is based on weak Message Authentication Codes (MAC) that are unforgeable on *random* messages even when evaluations of multiple *random* messages are available to the adversary. More formally, for a MAC protocol $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{TAG}, \mathsf{VRFY})$, we define unforgeability under *random message-random challenge* attacks (uf-rmrc) via the game UF-RMRC depicted on Figure 6.3. **Tag** queries return pairs $(m, \mathsf{TAG}_K(m))$ for fresh *random* messages

Game UF-RMRC$_{\mathsf{MAC}}$	**oracle Tag**():	**oracle Vrfy**(m,τ):
procedure main:	$m \xleftarrow{\$} \mathcal{M}$	If $m \notin C$
$K \xleftarrow{\$} \mathsf{KGen}$	$\tau \leftarrow \mathsf{TAG}_K(m)$	Ret $\perp$
$\mathsf{Forge} \leftarrow \mathtt{false}$	Ret (m,τ)	$C \leftarrow C \setminus \{m\}$
$C \leftarrow \emptyset$	**oracle Chal**():	If $\mathsf{VRFY}_K(m,\tau) = 1$
Run $\mathcal{A}^{\mathbf{Tag},\mathbf{Chal},\mathbf{Vrfy}}_{\mathsf{MAC}}$	$m \xleftarrow{\$} \mathcal{M}$	$\mathsf{Forge} \leftarrow \mathtt{true}$
Ret Forge	$C \leftarrow C \cup \{m\}$	Ret 1
	Ret m	Ret 0

Figure 6.3: The Game UF-RMRC$_{\mathsf{MAC}}$ which defines security against *random message-random challenge* attacks.

m. Moreover, **Vrfy** queries are only allowed if of the form (m,τ) for m previously output by the random challenge generator oracle **Chal**, and only a single verification query to **Vrfy** per valid challenge is allowed. For $t, q_{\mathsf{T}}, q_{\mathsf{C}}, q_{\mathsf{V}} > 0$, the uf-rmrc advantage function is defined as

$$\mathsf{Adv}^{\text{uf-rmrc}}_{\mathsf{MAC}}(t, q_{\mathsf{T}}, q_{\mathsf{C}}, q_{\mathsf{V}}) = \max_{\mathcal{A}}\{\Pr\left[\,(\text{UF-RMRC}_{\mathsf{MAC}})^{\mathcal{A}} \Rightarrow \mathtt{true}\,\right]\}\,,$$

where the maximum is over all adversaries $\mathcal{A}$ running in time t and making q_{T}, q_{C} and q_{V} queries to **Tag**, **Chal** and **Vrfy**, respectively.

The DM protocol. Our new 3-round authentication protocol, $\mathsf{DM}[\mathsf{MAC}] = (\mathcal{K}, \mathcal{P}, \mathcal{V})$ (DM stands for Double Mac) proceeds as follows where K_1, K_2 are generated using KGen.

$\mathcal{P}(K_1,K_2)$		$\mathcal{V}(K_1,K_2)$
$r_1 \xleftarrow{\$} \mathcal{M}$	$\xrightarrow{\quad r_1 \quad}$	$\tau_1 \xleftarrow{\$} \mathsf{TAG}_{K_1}(r_1)$;
If $\mathsf{VRFY}_{K_1}(r_1,\tau_1) = 1$	$\xleftarrow{\quad r_2,\tau_1 \quad}$	$r_2 \xleftarrow{\$} \mathcal{M}$
$\tau_2 \xleftarrow{\$} \mathsf{TAG}_{K_2}(r_2)$	$\xrightarrow{\quad \tau_2 \quad}$	Accept iff $\mathsf{VRFY}_{K_2}(r_2,\tau_2) = 1$

The intuition behind the proof is fairly simple: Each prover instance commits to a value r_1, and hence, in order for the prover to do something useful for an active adversary, such as tagging an arbitrary message under K_2, the attacker must provide a valid tag for r_1 under K_1. Yet, the attacker can only obtain valid tags through the transcript oracle in the first phase, and the used r'_1 values are very unlikely to collide with one of the values the prover instances commit to. Hence, with very high probability, the attacker never goes past the second round when interacting with the prover. The proof of the following theorem formalizes this intuition, but requires some care, mainly due to the interplay between the roles of the keys K_1 and K_2 in the reduction.

Theorem 6.4.1. [**Security of** DM] *For all* $t, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}} > 0$,

$$\begin{aligned} \mathit{Adv}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}_{\mathsf{DM}}(t, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}}) \;\leq\; & \mathit{Adv}^{\text{uf-rmrc}}_{\mathsf{MAC}}(t_1, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{P}}) \\ & + \mathit{Adv}^{\text{uf-rmrc}}_{\mathsf{MAC}}(t_2, q_{\mathsf{T}}, q_{\mathsf{V}}, q_{\mathsf{V}}) \end{aligned} \tag{6.1}$$

where $t_1 = t + \mathcal{O}(q_{\mathsf{T}} \cdot t_{\mathsf{TAG}})$, $t_2 = t + \mathcal{O}((q_{\mathsf{T}} + q_{\mathsf{V}}) \cdot t_{\mathsf{TAG}})$ *and* t_{TAG} *is the time to evaluate a single tag.*

Proof. The proof uses the games G_0 and G_1, whose main procedure and oracles are described in Figure 6.4. In order to avoid overloading our presentation, we omit the checks for correct input format in both games and assume that any input in incorrect format results in $\perp$. Also, throughout this proof, let us fix an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ making q_{T} queries to $\mathbf{T}$, q_{P} queries to $\mathbf{P}$, q_{V} queries to $\mathbf{V}$ and running in t steps.
Game G_0 is a compact representation of $\mathrm{AUTH}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}_{\mathsf{DM}}$, without all unnecessary steps. Therefore,

$$\mathsf{Adv}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}_{\mathsf{DM}}(\mathcal{A}) = \Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] \;. \tag{6.2}$$

Moreover, note that the game G_0, whenever a *valid* query $msg = r_2 \,||\, \tau_1$ is made to $\mathbf{P}$ in the second round, it sets the flag BAD if τ_1 is a valid tag for $\mathsf{state}[sid] = r_1$ sent in the first round, i.e., $\mathsf{VRFY}_{K_1}(\mathsf{state}[sid], \tau_1) = 1$. The second game, Game G_1, is identical to G_0, with the sole difference that no query (sid, msg) with $msg \neq \mathsf{start}$

procedure main: //G_0, G_1

$K_1, K_2 \xleftarrow{\$} \mathsf{KGen}$
For all $sid \in \mathbb{N}$ do
  $\mathsf{state}[sid] = \varepsilon$; $\mathsf{decision}[sid] = \bot$
  $\mathsf{done}[sid] = \mathtt{false}$
$\sigma_1 \xleftarrow{\$} \mathcal{A}_1^{\mathbf{P},\mathbf{T}}$ // Phase 1
$\mathcal{A}_2^{\mathbf{V}}(\sigma_1)$ // Phase 2
Ret $(\exists sid \in SID_{\mathcal{V}} : \mathsf{decision}[sid] = \mathsf{A})$

oracle V(sid, msg): // G_0, G_1

If $(sid \notin SID_{\mathcal{V}}) \vee \mathsf{done}[sid]$ then
  Ret $\bot$
If $\mathsf{state}[sid] = \epsilon$ then // 2nd round
  $\tau_1 \xleftarrow{\$} \mathsf{TAG}_{K_1}(msg)$
  $\mathsf{state}[sid] \xleftarrow{\$} \mathcal{M}$
  Ret $\mathsf{state}[sid] \,||\, \tau_1$
Else // decision
  $\mathsf{done}[sid] \leftarrow \mathtt{true}$
  If $\mathsf{VRFY}_{K_2}(\mathsf{state}[sid], msg) = 1$ then
    $\mathsf{decision}[sid] \leftarrow \mathsf{A}$
  Else
    $\mathsf{decision}[sid] \leftarrow \mathsf{R}$
  Ret $\mathsf{decision}[sid]$

oracle T(): // G_0, G_1

$r_1 \xleftarrow{\$} \mathcal{M}$; $\tau_1 \xleftarrow{\$} \mathsf{TAG}_{K_1}(r_1)$
$r_2 \xleftarrow{\$} \mathcal{M}$; $\tau_2 \xleftarrow{\$} \mathsf{TAG}_{K_2}(r_2)$
Ret $(r_1, (\tau_1, r_2), \tau_2)$

oracle P(id, msg): // G_0, $\boxed{G_1}$

If $(sid \notin SID_{\mathcal{P}}) \vee \mathsf{done}[sid]$ then
  Ret $\bot$
If $\mathsf{state}[sid] = \varepsilon$ then // 1st round
  If $msg \neq \mathsf{start}$ then
    Ret $\bot$
  $\mathsf{state}[sid] \xleftarrow{\$} \mathcal{M}$
  Ret $\mathsf{state}[sid]$
Else // 3rd round
  $\mathsf{done}[sid] \leftarrow \mathtt{true}$
  $r_2 \,||\, \tau_1 \leftarrow msg$
  If $\mathsf{VRFY}_{K_1}(\mathsf{state}[sid], \tau_1) = 1$ then
    $\mathsf{BAD} \leftarrow \mathtt{true}$
    $\tau_2 \xleftarrow{\$} \mathsf{TAG}_{K_2}(r_2)$
    $\boxed{\tau_2 \leftarrow \bot}$
    Ret τ_2
  Ret $\bot$.

Figure 6.4: Games G_0 and G_1 used in the proof of Theorem 6.4.1. Above, $a \,||\, b \leftarrow msg$ denotes the operation of parsing the string msg as the concatenation of the strings a and b of understood lengths.

made to **P** is accepted, i.e., they are all replied with $\bot$. The following claim bounds the difference between the probabilities $\mathcal{A}$ wins games G_0 and G_1, respectively.

Claim 6.4.2. *There exists an adversary $\mathcal{B}$ such that*

$$\Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] - \Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] \leq \mathsf{Adv}^{\mathsf{uf\text{-}rmrc}}_{\mathsf{MAC}}(\mathcal{B}) \,. \tag{6.3}$$

In particular, $\mathcal{B}$ makes q_{T} queries to **Tag** *and q_{P} queries to* **Chal** *and* **Vrfy**, *and runs in time $t' = t + \mathcal{O}(q_{\mathsf{T}} \cdot t_{\mathsf{TAG}})$, where t_{TAG} is the time needed to evaluate* TAG.

Proof. First note that G_0 and G_1 are equivalent-until-bad. By the fundamental lemma of game playing,

$$\Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \texttt{true} \,\right] - \Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \texttt{true} \,\right] \leq \Pr\left[\, G_1^{\mathcal{A}} \text{ sets } \mathsf{BAD} \,\right] \ .$$

We now construct the adversary $\mathcal{B}$ for the UF-RMRC$_{\mathsf{MAC}}$ game. The crucial observation here is that as long as we are only concerned about the probability of BAD being set, we only need to look at the first phase of the game, which will in particular avoid the reduction simulating the second phase. The adversary $\mathcal{B}$ simulates the interaction of the adversary $\mathcal{A}_1$ in the *first* phase of the game $\text{AUTH}_{\mathsf{DM}}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})}$ as follows: First, it selects $K_2 \xleftarrow{\$} \mathsf{KGen}$. Upon receiving a $\mathbf{T}$ query from $\mathcal{A}_1$, $\mathcal{B}$ makes a query to its **Tag** oracle to get a pair r_1, τ_1 and also computes (r_2, τ_2) by sampling $r_2 \xleftarrow{\$} \mathcal{M}$ and setting $\tau_2 \xleftarrow{\$} \mathsf{TAG}_{K_2}(r_2)$ (recall that $\mathcal{B}$ can compute $\mathsf{TAG}_{K_2}(\cdot)$ using the K_2 it has chosen for the simulation). It then returns $(r_1, (\tau_1, r_2), \tau_2)$ as a transcript to $\mathcal{A}_1$. On every $\mathbf{P}$ query (sid, start), $\mathcal{B}$ makes a query to its **Chal** oracle, which returns a message message r_1. Then, $\mathcal{B}$ sends r_1 to $\mathcal{A}_1$ and upon receiving (sid, τ_1, r_2), for the same sid, $\mathcal{B}$ sends τ_1 as a forgery for r_1 (that is, $\mathcal{B}$ makes a query (r_1, τ_1) to its **Vrfy** oracle), but returns $\perp$ to $\mathcal{A}_1$. It is straightforward to verify that $\mathcal{B}$ simulates perfectly the first phase of G_1 to $\mathcal{A}_1$, and that the probability that BAD is set is exactly the probability that $\mathcal{B}$ forges. Finally, for the simulation, $\mathcal{B}$ calls its **Tag** oracle q_{T} times, and its **Chal** and **Vrfy** oracles, each, q_{P} times, and also needs to compute q_{T} tags by itself. □

To conclude the proof, we reduce the problem of $\mathcal{A}$ winning the game G_1 to forging MAC in the game UF-RMRC$_{\mathsf{MAC}}$. Specifically, we build an adversary $\mathcal{C}$ such that

$$\Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \texttt{true} \,\right] = \Pr\left[\, \text{UF-RMRC}_{\mathsf{MAC}}^{\mathcal{C}} \Rightarrow \texttt{true} \,\right] \ . \tag{6.4}$$

The adversary $\mathcal{C}$ simulates an interaction of $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with the game G_1 as follows: It first chooses $K_1 \xleftarrow{\$} \mathsf{KGen}$. When $\mathcal{A}_1$ makes a query to $\mathbf{T}$, $\mathcal{C}$ first generates $r_1 \xleftarrow{\$} \mathcal{M}$ and $\tau_1 \xleftarrow{\$} \mathsf{TAG}_{K_1}(r_1)$, then samples a pair (r_2, τ_2) by querying its own **Tag** oracle and finally returns $(r_1, (r_2, \tau_1), \tau_2)$ to $\mathcal{A}$. Moreover, every query (sid, msg) to $\mathbf{P}$ is replied as follows: If $msg = \mathsf{start}$, then $\mathcal{C}$ simply samples $r \xleftarrow{\$} \mathcal{M}$ and returns it to $\mathcal{A}_1$. If $msg = r' \,||\, \tau'$, $\mathcal{C}$ replies with $\perp$. Finally, whenever $\mathcal{A}_2$

Game $\mathrm{LPN}_{n,\eta}$	**oracle Sample**():
procedure main:	$\mathbf{a} \xleftarrow{\$} \mathbb{Z}_2^n$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$	$e \leftarrow \mathsf{Ber}_\eta$
$d \leftarrow \mathcal{A}^{\mathbf{Sample}}$	Ret $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e)$

Figure 6.5: Game $\mathrm{LPN}_{n,\eta}$.

makes a query (sid, r_1) to $\mathbf{V}$, $\mathcal{C}$ queries its **Chal** oracle, obtaining a value r_2. It then samples $\tau_1 \xleftarrow{\$} \mathsf{TAG}_{K_1}(r_1)$, and returns $r_2 \,||\, \tau_1$. If $\mathcal{A}_2$ queries $\mathbf{V}$ again for the same *sid*, with a value τ_2, then $\mathcal{C}$ submits (r_2, τ_2) to **Vrfy**, and returns the outcome to $\mathcal{A}_2$. It is not hard to see that the probability that $\mathcal{C}$ forges is exactly the probability that $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ wins the game G_1. Also, $\mathcal{C}$ has running time $t_2 = t + \mathcal{O}((t_{\mathsf{T}} + t_{\mathsf{V}}) \cdot t_{\mathsf{TAG}})$, and makes q_{T} queries to **Tag** and q_{V} queries to **Chal** and **Vrfy**. □

It is worth mentioning that the security of DM is based on a very weak assumption. Previous generic constructions require either a much stronger MAC allowing for chosen-message queries, and giving a challenge-response protocol directly, or a weak PRF [48], which is a strictly stronger assumption, as a weak PRF yields a (deterministic) uf-rmrc-secure MAC. Also, in contrast to the weak-PRF based protocol of [48], our proof avoids rewinding, hence yielding an essentially tight reduction.

6.5 Efficient Instantiation from Learning Parity with Noise

In this section, we instantiate DM using the Learning Parity with Noise (LPN) assumption. In the game $\mathrm{LPN}_{n,\eta}$ (shown in Figure 6.5), the **Sample** oracle, given a secret $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$, returns pairs $(\mathbf{a}, \mathbf{as} + e)$ for a random $\mathbf{a} \in \mathbb{Z}_2^n$ and $e \xleftarrow{\$} \mathsf{Ber}_\eta$ upon each invocation, where **as** denotes scalar product in $\mathbb{Z}_2$. The (decisional) LPN is the problem of distinguishing $\mathrm{LPN}_{n,\eta}$ from $\mathrm{LPN}_{n,1/2}$. For $t, q > 0$, we

define the lpn advantage function as

$$\mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(t,q) = \max\left\{\Pr\left[\,\mathrm{LPN}^{\mathcal{A}}_{n,\eta} \Rightarrow 1\,\right] - \Pr\left[\,\mathrm{LPN}^{\mathcal{A}}_{n,1/2} \Rightarrow 1\,\right]\right\} \tag{6.5}$$

where the maximum is taken over all adversaries $\mathcal{A}$ running in time t and making q queries to the **Sample** oracle.

We define $\mathsf{MAC}_{\mathrm{LPN}} = (\mathsf{KGen}, \mathsf{TAG}, \mathsf{VRFY})$ in Figure 6.6. $\mathsf{MAC}_{\mathrm{LPN}}$ has keyspace $\mathcal{K} = \mathbb{Z}_2^n$, message space $\mathcal{M} = \mathbb{Z}_2^{m\times n}$ and tag space $\mathcal{T} = \mathbb{Z}_2^m$, and is parametrized by constants η, η' such that $0 < \eta < \eta' < 1/2$.

$\underline{\mathsf{KGen}:}$	$\underline{\mathsf{TAG}(\mathbf{s},\mathbf{A}):}$	$\underline{\mathsf{VRFY}(\mathbf{s},\mathbf{A},\mathbf{t}):}$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$; Ret $\mathbf{s}$	$\mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\eta^m$; Ret $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$.	If $\mathsf{hw}(\mathbf{t} - \mathbf{A}\mathbf{s}) < \eta' \cdot m$ then Ret 1 else Ret 0

Figure 6.6: A $\mathsf{uf\text{-}rmrc}$-secure MAC based on LPN.

The expected hamming weight of $\mathbf{t} - \mathbf{A}\mathbf{s}$ is $\eta \cdot m$. Therefore the completeness error of $\mathsf{MAC}_{\mathrm{LPN}}$ can be upper bounded by the Chernoff bound (2.1) as

$$\epsilon_c = \Pr\left[\,\mathsf{hw}(\mathbf{e}) > \eta' \cdot m\,\right] \leq 2^{-\mathrm{D}(\eta' \,||\, \eta)m}\ . \tag{6.6}$$

The following lemma states that $\mathsf{MAC}_{\mathrm{LPN}}$ is $\mathsf{uf\text{-}rmrc}$-secure assuming LPN is hard. Its proof uses ideas similar to the ones used in the proof that the HB protocol is secure against passive attacks [90]. Similar ideas are also implicit in the LPN-based randomized weak PRF construction by Applebaum *et al* [8].

Lemma 6.5.1. [**Security of** $\mathsf{MAC}_{\mathrm{LPN}}$] *Let* $\bar{\eta} = \eta + \eta' - 2\eta\eta'$ *and* η'' *such that*[3] $0 < \bar{\eta} < \eta'' < 1/2$. *Then, for all* $t, q_{\mathsf{T}}, q_{\mathsf{C}}, q_{\mathsf{V}} > 0$,

$$\mathsf{Adv}^{\mathsf{uf\text{-}rmrc}}_{\mathsf{MAC}_{\mathrm{LPN}}}(t, q_{\mathsf{T}}, q_{\mathsf{C}}, q_{\mathsf{V}}) \leq \mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(t', q) + q_{\mathsf{V}} \cdot \left(2^{-\mathrm{D}(\eta'' \,||\, \bar{\eta})m} + 2^{-(1-\mathrm{H}_2(\eta''))m}\right)\ ,$$

where $t' = t + \mathcal{O}(q_{\mathsf{C}})$ *and* $q = (q_{\mathsf{T}} + q_{\mathsf{C}}) \cdot m$.

Proof. We use the sequence of games whose main procedure and oracles are shown in Figure 6.7. Throughout the proof, we fix an adversary $\mathcal{A}$ making q_{T}, q_{C} and q_{V} queries to **Tag**, **Chal** and **Vrfy** respectively.

[3] For constants $\eta, \eta' \in (0, 1/2)$, a constant η'' within that range always exists.

procedure main:
$//G_0$-G_4
Let $\bar{\eta} = \eta + \eta' - 2\eta\eta'$
$\eta'' \in (\bar{\eta}, 1/2)$
$\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$
$\mathsf{Forge} \leftarrow \mathtt{false}$
$C \leftarrow \emptyset$
Run $\mathcal{A}^{\mathbf{Tag},\mathbf{Chal},\mathbf{Vrfy}}$
Ret Forge

Chal(): $//G_0$-G_4
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{m\times n}$
$C \leftarrow C \cup \{\mathbf{A}\}$
Ret $\mathbf{A}$

Tag(): $//G_0$-G_3
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{m\times n}$
$\mathbf{e} \leftarrow \mathsf{Ber}_\eta^m$
$\mathbf{t} \leftarrow \mathbf{As} + \mathbf{e}$
Ret $(\mathbf{A}, \mathbf{t})$

Vrfy$(\mathbf{A}^*, \mathbf{t}^*)$: $//G_0$
If $\mathbf{A}^* \notin C$ Ret $\perp$
$C \leftarrow C \setminus \{\mathbf{A}^*\}$
$\mathbf{e}^* \leftarrow \mathsf{Ber}_\eta^m$
If $\mathsf{hw}(\mathbf{t}^* - \mathbf{A}^*\mathbf{s}) \le \eta' m$
  $\mathsf{Forge} \leftarrow \mathtt{true}$
  If $\mathsf{hw}(\mathbf{t}^*\text{-}\mathbf{A}^*\mathbf{s}\text{-}\mathbf{e}^*) > \eta'' m$
    $\mathsf{BAD} \leftarrow \mathtt{true}$
  Ret 1
Ret 0

Vrfy$(\mathbf{A}^*, \mathbf{t}^*)$: $//G_1$, $\boxed{G_2}$
If $\mathbf{A}^* \notin C$ Ret $\perp$
$C \leftarrow C \setminus \{\mathbf{A}^*\}$
$\mathbf{e}^* \leftarrow \mathsf{Ber}_\eta^m$
If $\mathsf{hw}(\mathbf{t}^* - \mathbf{A}^*\mathbf{s}) \le \eta' m$
  $\mathsf{Forge} \leftarrow \mathtt{true}$
  If $\mathsf{hw}(\mathbf{t}^*\text{-}\mathbf{A}^*\mathbf{s}\text{-}\mathbf{e}^*) > \eta'' m$
    $\mathsf{BAD} \leftarrow \mathtt{true}$
    $\boxed{\mathsf{Forge} \leftarrow \mathtt{false}}$
    $\boxed{\text{Ret } 0}$
  Ret 1
If $\mathsf{hw}(\mathbf{t}^*\text{-}\mathbf{A}^*\mathbf{s}\text{-}\mathbf{e}^*) \le \eta'' m$
  $\mathsf{Forge} \leftarrow \mathtt{true}$
  Ret 1
Ret 0

Vrfy$(\mathbf{A}^*, \mathbf{t}^*)$: $//G_3$
If $\mathbf{A}^* \notin C$
  Ret $\perp$
$C \leftarrow C \setminus \{\mathbf{A}^*\}$
$\mathbf{e}^* \leftarrow \mathsf{Ber}_\eta^m$
If $\mathsf{hw}(\mathbf{t}^*\text{-}\mathbf{A}^*\mathbf{s}\text{-}\mathbf{e}^*) \le \eta'' m$
  $\mathsf{Forge} \leftarrow \mathtt{true}$
  Ret 1
Ret 0

Tag(): $//G_4$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{m\times n}$
$\mathbf{t} \xleftarrow{\$} \mathbb{Z}_2^m$
Ret $(\mathbf{A}, \mathbf{t})$

Vrfy$(\mathbf{A}^*, \mathbf{t}^*)$: $//G_4$
If $\mathbf{A} \notin C$
  Ret $\perp$
$C \leftarrow C \setminus \{\mathbf{A}^*\}$
$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^m$
if $\mathsf{hw}(\mathbf{t}^* - \mathbf{r}) \le \eta'' \cdot m$
  $\mathsf{Forge} \leftarrow \mathtt{true}$
  Ret 1
Ret 0

Figure 6.7: Sequence of games for the proof of Lemma 6.5.1.

Game G_0 is equivalent to UF-RMRC$_{\mathsf{MAC}_{\mathrm{LPN}}}$. All extra commands in the code of the **Vrfy** oracle serve as internal bookkeeping and do not affect adversary's view. Therefore

$$\mathsf{Adv}^{\mathsf{uf\text{-}rmrc}}_{\mathsf{MAC}_{\mathrm{LPN}}}(\mathcal{A}) = \Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] . \tag{6.7}$$

Game G_1 is identical to G_0 except in the way queries to **Vrfy** are answered. However, whenever Forge is set to $\mathtt{true}$ in G_0, so is in G_1 and thus

$$\Pr\left[\, G_0^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] \le \Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] . \tag{6.8}$$

Games G_2, G_1 are clearly equivalent-until-bad. Below, we compute the probabilty that the $\mathsf{BAD} \leftarrow \mathtt{true}$ happens in G_2. Let $\mathbf{y} = \mathbf{t}^* - \mathbf{A}^*\mathbf{s}$ and $\mathbf{y}' = \mathbf{t}^* - \mathbf{A}^*\mathbf{s} - \mathbf{e}^*$. Consider a single query $(\mathbf{A}^*, \mathbf{t}^*)$ to **Vrfy**. Then

$$\begin{aligned} \Pr\left[\, \mathsf{BAD} \leftarrow \mathtt{true} \,\right] &= \Pr\left[\, (\mathsf{hw}(\mathbf{y}') > \eta'' m) \wedge (\mathsf{hw}(\mathbf{y}) \leq \eta' m) \,\right] \\ &\leq \Pr\left[\, \mathsf{hw}(\mathbf{y}') > \eta'' m \;\middle|\; \mathsf{hw}(\mathbf{y}) \leq \eta' m \,\right] . \end{aligned}$$

Each coordinate y'_i of $\mathbf{y}'$ is independent with $\mathop{\mathbb{E}}_{e_i^*}[\, y'_i \,] = (1-2\eta)y_i + \eta$. Therefore, (since $\mathsf{hw}(\mathbf{y}) \leq \eta' m$)

$$\mathop{\mathbb{E}}_{\mathbf{e}^*}[\, \mathsf{hw}(\mathbf{y}') \,] = (1-2\eta)\mathsf{hw}(\mathbf{y}) + \eta \cdot m \leq (\eta + \eta' - 2\eta\eta') \cdot m = \bar{\eta} \cdot m \; .$$

Since $\eta'' > \bar{\eta}$, by applying Chernoff bound, we get $\Pr[\, \mathsf{hw}(\mathbf{y}') > \eta'' m \,] \leq 2^{-\mathrm{D}(\eta'' \,||\, \bar{\eta})m}$. Using the fundamental lemma of game playing and the union bound across q_{V} queries to **Vrfy** we get that for every $\mathcal{A}$ (even unbounded)

$$\begin{aligned} \Pr\left[\, G_1^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] - \Pr\left[\, G_2^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] &\leq \Pr\left[\, \mathsf{BAD} \leftarrow \mathtt{true} \,\right] \\ &\leq q_{\mathsf{V}} \cdot 2^{-\mathrm{D}(\eta'' \,||\, \bar{\eta})m} \; . \end{aligned} \tag{6.9}$$

G_3 is essentially a compact rewriting of G_2. The view of any adversary $\mathcal{A}$ (even unbounded) is exactly the same in both games. Therefore

$$\Pr\left[\, G_2^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] = \Pr\left[\, G_3^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] \; . \tag{6.10}$$

G_4 differs from G_3 with respect to both **Tag** and **Vrfy** oracles. We claim that there exists an adversary $\mathcal{B}$ against LPN such that

$$\Pr\left[\, G_3^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] - \Pr\left[\, G_4^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] = \mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(\mathcal{B}) \; . \tag{6.11}$$

$\mathcal{B}$ maintains a set C (initialized to $\emptyset$) that contains all messages $\mathbf{A}$ for which $\mathcal{A}$ is allowed to query the oracle **Vrfy** and replies to $\mathcal{A}$'s queries as follows: For each query to **Tag** by $\mathcal{A}$, $\mathcal{B}$ makes m queries to its **Sample** oracle. Let $\{(\mathbf{a}_i, z_i)\}_{i \in [m]}$ be the samples returned. $\mathcal{B}$ then sends $(\mathbf{A}, \mathbf{z})$ to $\mathcal{A}$ where $\mathbf{A}$ is an $m \times n$ matrix with the i-th row being $\mathbf{a}_i$ and $\mathbf{z} = (z_1, \ldots, z_m)^T$. On each **Chal** query by $\mathcal{A}$, $\mathcal{B}$ gets m more samples $(\mathbf{A}^*, \mathbf{z}^*)$ and returns $\mathbf{A}^*$ to $\mathcal{A}$. At the same time, $\mathcal{B}$ adds $\mathbf{A}^*$ to C along with $\mathbf{z}^*$. On a $(\mathbf{A}^*, \mathbf{t}^*)$ query to **Vrfy**, $\mathcal{B}$ first checks that $\mathbf{A}^* \in C$

and if so, recovers the vector $\mathbf{z}^*$ that corresponds to $\mathbf{A}^*$. It then checks whether $\mathsf{hw}(\mathbf{t}^* - \mathbf{z}^*) \leq \eta'' m$ and if so, it outputs 1 and terminates. Otherwise it returns 0 to $\mathcal{A}$, removes $\mathbf{A}^*$ and $\mathbf{z}^*$ from C and resumes the simulation. Clearly, if $\mathcal{B}$ is playing in game $\mathrm{LPN}_{n,\eta}$, then it simulates G_3 perfectly to $\mathcal{A}$ whereas if it is playing in game $\mathrm{LPN}_{n,1/2}$, then it simulates G_4 perfectly to $\mathcal{A}$. Thus

$$\begin{aligned}\Pr\left[\, G_3^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] - \Pr\left[\, G_4^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] &= \Pr\left[\, \mathrm{LPN}_{n,\eta}^{\mathcal{B}} \Rightarrow 1 \,\right] - \Pr\left[\, \mathrm{LPN}_{n,1/2}^{\mathcal{B}} \Rightarrow 1 \,\right] \\ &= \mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(\mathcal{B})\ .\end{aligned}$$

Moreover, for each **Tag** and each **Chal** query by $\mathcal{A}$, $\mathcal{B}$ makes m queries to its **Sample** oracle. Hence, $\mathcal{B}$ makes $(q_{\mathsf{T}} + q_{\mathsf{C}})m$ queries in total to its oracle and runs in time $t + \mathcal{O}(q_{\mathsf{C}})$.

Finally, the view of $\mathcal{A}$ in G_4 is completely independent of $\mathbf{s}$. Consider again a single query $(\mathbf{A}^*, \mathbf{t}^*)$ to **Vrfy**. It is straightforward to verify that $\mathbf{t}^* - \mathbf{r}$ is uniform and random over $\mathbb{Z}_2^m$. Therefore the probabilty a verification query causes **Vrfy** to return 1 is

$$2^{-m} \sum_{i=0}^{\lfloor \eta'' m \rfloor} \binom{m}{i} \leq 2^{-(1-\mathrm{H}_2(\eta''))m}\ .$$

Using the union bound,

$$\Pr\left[\, G_4^{\mathcal{A}} \Rightarrow \mathtt{true} \,\right] \leq q_{\mathsf{V}} \cdot 2^{-(1-\mathrm{H}_2(\eta''))m}\ . \tag{6.12}$$

Proof then follows combining (6.7), (6.8), (6.9), (6.10), (6.11) and (6.12). □

Instantiating DM with $\mathsf{MAC}_{\mathrm{LPN}}$ yields a protocol $\mathsf{DM}_{\mathrm{LPN}}$ whose secret key consists of two vectors $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}_2^n$. The prover first selects a random matrix $\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_2^{m \times n}$ and sends it to the verifier. The verifier then selects another matrix $\mathbf{A}_2 \xleftarrow{\$} \mathbb{Z}_2^{m \times n}$ and a noise vector $\mathbf{e}_1 \xleftarrow{\$} \mathsf{Ber}_\eta^m$, and sends $(\mathbf{A}_2, \mathbf{A}_1\mathbf{s}_1 + \mathbf{e}_1)$ to the prover. Upon receiving a pair $(\mathbf{A}_2, \mathbf{z}_1)$, the prover checks whether $\mathsf{hw}(\mathbf{z}_1 - \mathbf{A}_1\mathbf{s}_1) \leq \eta' \cdot m$, and if so, samples $\mathbf{e}_2 \xleftarrow{\$} \mathsf{Ber}_\eta^m$, and sends $\mathbf{A}_2\mathbf{s}_2 + \mathbf{e}_2$ back to the verifier. Finally, the verifier, on input $\mathbf{z}_2$, accepts iff $\mathsf{hw}(\mathbf{z}_2 - \mathbf{A}_2\mathbf{s}_2) < \eta' \cdot m$.

The overall advantage of our $\mathsf{DM}_{\mathrm{LPN}}$ protocol can be computed, combining (6.1) and Lemma 6.5.1, as

$$\begin{aligned}\mathsf{Adv}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}_{\mathsf{DM}_{\mathrm{LPN}}}(t, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}}) &\leq (q_{\mathsf{P}} + q_{\mathsf{V}}) \left[2^{-\mathrm{D}(\eta'' \,||\, \bar{\eta})m} + 2^{-(1-\mathrm{H}_2(\eta''))m} \right] \\ &+ \mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(t_1, q_1) + \mathsf{Adv}^{\mathsf{lpn}}_{n,\eta}(t_2, q_2)\end{aligned} \tag{6.13}$$

Table 6.1: (Asymptotic) comparison of known LPN-based active secure protocols: Here, n is the secret-size for the underlying LPN problem and ϵ is the assumed hardness of LPN given $q = (q_{\mathsf{P}} + q_{\mathsf{V}} + q_{\mathsf{T}})m$ samples.

Protocol	rounds	**Complexity**			Security
		keysize	Communication	Computation	
HB^+ [87]	3	$2n$	$2nm + n$	$\Theta(n \cdot m)$	$q_{\mathsf{V}} \cdot \sqrt{\epsilon}$
KP^+ [94]	2	$\geq 4.2n$	$\geq 2.1nm$	$\Theta(n \cdot m)$	$q_{\mathsf{V}} \cdot \epsilon$
This work	3	$2n$	$2nm + 2n$	$\Theta(n \cdot m)$	ϵ

where $q_1 = (q_{\mathsf{T}} + q_{\mathsf{P}})m$, $q_2 = (q_{\mathsf{T}} + q_{\mathsf{V}})m$, $t_1 = t + \mathcal{O}(q_{\mathsf{T}} + q_{\mathsf{P}}) \cdot t_{\mathsf{TAG}}$, $t_2 = t + \mathcal{O}(q_{\mathsf{T}} + q_{\mathsf{V}}) \cdot t_{\mathsf{TAG}}$ and t_{TAG} is the time to compute a single LPN mac. It is easy to see that this bound is superior to the one of HB^+ [87, 90], due to their use of rewinding, which results in a loose security reduction. Comparing with KP^+ [94] is more complicated. For that, we use the bound provided in their security reduction [94, Thm. 1]. Moreover, we need to adapt their security bound to the case where both transcript and *multiple* verification queries are allowed. When the keysize of KP^+ is 2ℓ, then the overall bound can be computed as

$$\begin{aligned} \mathsf{Adv}^{(\{\mathsf{T},\mathsf{P}\},\{\mathsf{V}\})\text{-auth}}_{\mathrm{KP}^+}(t, q_{\mathsf{T}}, q_{\mathsf{P}}, q_{\mathsf{V}}) \;\leq\; & q_{\mathsf{V}} \left[\frac{(q_{\mathsf{P}} + q_{\mathsf{T}})m}{2^{g+1}} + (q_{\mathsf{P}} + q_{\mathsf{T}})2^{-c_1 \cdot \ell} + 2^{-c_2 \cdot m} \right] \\ & + q_{\mathsf{V}} \cdot \mathsf{Adv}^{\mathsf{lpn}}_{d,\eta}(t', q) \end{aligned} \tag{6.14}$$

where $t' = t + \mathcal{O}(q_{\mathsf{P}} + q_{\mathsf{T}})$, $q = (q_{\mathsf{P}} + q_{\mathsf{T}})m$, c_1, c_2 are constants, and d, g are parameters such that $d + g \leq \ell/2.1$. Also, for keysize 2ℓ, KP^+ has communication complexity $2\ell + m\ell + m$. Notice that the security of KP^+ (with keysize 2ℓ) is based on the hardness of LPN with secret size $d < \ell/2.1$. In contrast, $\mathsf{DM}_{\mathrm{LPN}}$ with keysize $2n$ relies on the hardness of LPN with secret size n. Moreover, too small values for g affect negatively the security of KP^+ and in practice one might have to choose $g = d < \ell/4.2$. This means that, even in the most optimistic case ($\ell = 2.1d$), for the same security level, i.e. LPN with the same secret size, $\mathsf{DM}_{\mathrm{LPN}}$ requires a substantially smaller key than KP^+ and incurs lower communication complexity.

The comparison with both HB^+ and KP^+ is even more in our favor when multiple verification queries are considered. Indeed, the bounds for HB^+ and KP^+

increase linearly with the number of verification queries.

Based on the above analysis, Table 6.1 provides an a symptotic comparison of our LPN-based authentication protocol with HB^+ and KP^+. In the table, we have used LPN with fixed secret size n as the underlying hardness assumption across all three protocols.

Chapter 6 is, in part, a reprint of the paper "Secret-Key Authentication Beyond the Challenge-Response Paradigm: Denitional Issues and New Protocols" [119] co-authored with Stefano Tessaro. The dissertation author was the primary investigator and author of the relevant part of the paper.

Bibliography

[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.

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

[6] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugere, Robert Fitzpatrick, and Ludovic Perret. On the Complexity of the BKW Algorithm on LWE. Cryptology ePrint Archive, Report 2012/636, 2012. Avaialable at http://eprint.iacr.org/2012.

[7] Michael Alekhnovich. More on Average Case vs Approximation Complexity. In *FOCS*, pages 298–307, 2003.

[8] 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.

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

[10] 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/.

[11] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A Family of Fast Syndrome Based Cryptographic Hash Functions. In *Mycrypt*, pages 64–83, 2005.

[12] Gildas Avoine. Adversarial Model for Radio Frequency Identification. *IACR Cryptology ePrint Archive*, 2005:49, 2005.

[13] Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing Time Complexity in RFID Systems. In *Selected Areas in Cryptography*, pages 291–306, 2005.

[14] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 =$ improves information set decoding. In *EUROCRYPT*, 2012.

[15] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged Public-Key Encryption: How to Protect against Bad Randomness. In *ASIACRYPT*, pages 232–249, 2009.

[16] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Advances in Cryptology – EUROCRYPT 2009*, number 5479 in Lecture Notes in Computer Science, pages 1–35. Springer, 2009.

[17] Mihir Bellare and Phillip Rogaway. Entity Authentication and Key Distribution. In *CRYPTO*, pages 232–249, 1993.

[18] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In *EUROCRYPT*, pages 409–426, 2006.

[19] Mihir Bellare, Stefano Tessaro, and Alexander Vardy. Semantic Security for the Wiretap Channel. In *CRYPTO*, pages 294–311, 2012.

[20] Elwyn Berlekamp, Robert J. McEliece, and Henk van Tilborg. On the Inherent Intractability of Certain Coding Problems. *Transactions on Information Theory*, 24(3):384–386, May 1978.

[21] Daniel J. Bernstein and Tanja Lange. Never Trust a Bunny. Cryptology ePrint Archive, Report 2012/355, 2012.

[22] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller Decoding Exponents: Ball-Collision Decoding. In *CRYPTO*, pages 743–760, 2011.

[23] Daniel Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. In *CRYPTO*, pages 1–12, 1998.

[24] 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.

[25] 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.

[26] Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract). In *STOC*, pages 103–112, 1988.

[27] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *Advances in Cryptology – CRYPTO 2008*, number 5157 in Lecture Notes in Computer Science, pages 335–359. Springer, 2008.

[28] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[29] Elette Boyle, Gil Segev, and Daniel Wichs. Fully Leakage-Resilient Signatures. In *EUROCRYPT*, pages 89–108, 2011.

[30] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVPs. In *CRYPTO*, 2012.

[31] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehle. Classical Hardness of Learning with Errors. In *STOC*. ACM, 2013 (to appear).

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

[33] Ernest F. Brickell. Some Ideal Secret Sharing Schemes. In *EUROCRYPT*, pages 468–475, 1989.

[34] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In *SecPerU*, pages 28–33, 2006.

[35] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally Private Information Retrieval with Polylogarithmic Communication. In *EUROCRYPT*, pages 402–414, 1999.

[36] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT*, pages 207–222, 2004.

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

[38] David Cash, Eike Kiltz, and Victor Shoup. The Twin Diffie-Hellman Problem and Applications. *J. Cryptology*, 22(4):470–504, 2009.

[39] Hao Chen and Ronald Cramer. Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields. In *CRYPTO*, pages 521–536, 2006.

[40] Hao Chen, Ronald Cramer, Shafi Goldwasser, Robbert de Haan, and Vinod Vaikuntanathan. Secure Computation from Random Error Correcting Codes. In *EUROCRYPT*, pages 291–310, 2007.

[41] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme. In *ASIACRYPT*, pages 157–174, 2001.

[42] Ronald Cramer and Victor Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO*, pages 13–25, 1998.

[43] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology – EUROCRYPT 2002*, volume 2332, pages 45–64. Springer, 2002.

[44] Ivan Damgård and Michael Østergaard Pedersen. RFID Security: Tradeoffs between Security and Efficiency. In *CT-RSA*, pages 318–332, 2008.

[45] George I. Davida, Richard A. DeMillo, and Richard J. Lipton. Protecting Shared Cryptographic Keys. In *IEEE Symposium on Security and Privacy*, pages 100–102, 1980.

[46] Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. In *SecureComm*, pages 59–66, 2005.

[47] 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.

[48] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message Authentication, Revisited. In *EUROCRYPT*, 2012.

[49] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.*, 38(1):97–139, 2008. Preliminary Version in EUROCRYPT 2004.

[50] Yevgeniy Dodis and Adam Smith. Correcting Errors Without Leaking Partial Information. In *STOC*, pages 654–663, 2005.

[51] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography (Extended Abstract). In *STOC*, pages 542–552, 1991.

[52] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. IND-CCA Secure Cryptography Based on a Variant of the LPN Problem. In *ASIACRYPT*, pages 485–503, 2012.

[53] Rafael Dowsley, Jörn Müller-Quade, and Anderson C. A. Nascimento. A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. In *CT-RSA*, pages 240–251, 2009.

[54] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade, and Anderson C. A. Nascimento. Oblivious Transfer Based on the McEliece Assumptions. In *ICITS*, pages 107–117, 2008.

[55] Dang N. Duc and Kwangjo Kim. Securing HB^+ Against GRS Man-in-the-Middle Attack. In *SCIS*, 2007.

[56] Peter Elias. List Decoding for Noisy Channels, September 1957, Technical Reports 335.

[57] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-Knowledge Proofs of Identity. *J. Cryptology*, 1(2):77–94, 1988.

[58] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. 263:186–194, 1986.

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

[60] David Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. Number-theoretic constructions of lossy and correlation-secure trapdoor functions. Manuscript.

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

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

[63] 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.

[64] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good Variants of HB^+ Are Hard to Find. In *Financial Cryptography*, pages 156–170, 2008.

[65] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. $HB^\#$: Increasing the Security and Efficiency of HB^+. In *EUROCRYPT*, pages 361–378, 2008.

[66] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. How to encrypt with the lpn problem. In *ICALP (2)*, pages 679–690, 2008.

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

[68] 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.

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

[70] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *STOC*, pages 365–377, 1982.

[71] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[72] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[73] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In *Advances in Cryptology - CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231, 1988.

[74] Richard W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950.

[75] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security. In *ASIACRYPT*, pages 70–88, 2011.

[76] Brett Hemenway and Rafail Ostrovsky. Public-Key Locally-Decodable Codes. In *CRYPTO*, pages 126–143, 2008.

[77] Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public Key Locally Decodable Codes with Short Keys. In *APPROX-RANDOM*, pages 605–615, 2011.

[78] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An Efficient Authentication Protocol Based on Ring-LPN. In *FSE*, 2012.

[79] Dennis Hofheinz and Eike Kiltz. Practical Chosen Ciphertext Secure Encryption from Factoring. In *Advances in Cryptology – EUROCRYPT 2009*, pages 313–332. Springer, 2009.

[80] Susan Hohenberger, Allison B. Lewko, and Brent Waters. Detecting Dangerous Queries: A New Approach for Chosen Ciphertext Security. In *EUROCRYPT*, pages 663–681, 2012.

[81] Nicholas J. Hopper and Manuel Blum. Secure Human Identification Protocols. In *ASIACRYPT*, pages 52–66, 2001.

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

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

[84] A. Kh. Al Jabri. A Statistical Decoding Algorithm for General Linear Block Codes. In *IMA Int. Conf.*, pages 1–8, 2001.

[85] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. In *ASIACRYPT*, pages 663–680, 2012.

[86] Ari Juels and Martin Wattenberg. A Fuzzy Commitment Scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[87] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols. In *CRYPTO*, pages 293–308, 2005.

[88] Ari Juels and Stephen A. Weis. Defining Strong Privacy for RFID. *IACR Cryptology ePrint Archive*, 2006:137, 2006.

[89] Ehud D. Karnin, J. W. Greene, and Martin E. Hellman. On Secret Sharing Systems. *IEEE Transactions on Information Theory*, 29(1):35–41, 1983.

[90] 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.

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

[92] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive Trapdoor Functions and Chosen-Ciphertext Security. In *EUROCRYPT*, pages 673–692, 2010.

[93] Eike Kiltz, Adam O'Neill, and Adam Smith. Instantiability of RSA-OAEP under Chosen-Plaintext Attack. In *CRYPTO*, pages 295–313, 2010.

[94] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient Authentication from Hard Learning Problems. In *EUROCRYPT*, pages 7–26, 2011.

[95] Kazukuni Kobara and Hideki Imai. Semantically Secure McEliece Public-Key Cryptosystems-Conversions for McEliece PKC. In *Public Key Cryptography*, pages 19–35, 2001.

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

[97] Adeline Langlois and Damien Stehlé. Hardness of Decision (R)LWE for Any Modulus. *IACR Cryptology ePrint Archive*, 2012:91, 2012.

[98] Pil Joong Lee and Ernest F. Brickell. An Observation on the Security of McEliece's Public-Key Cryptosystem. In *EUROCRYPT*, pages 275–280, 1988.

[99] Jeffrey S. Leon. A Probabilistic Algorithm for Computing Minimum Weights of Large Error-Correcting Codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.

[100] Yehuda Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. *J. Cryptology*, 19(3):359–377, 2006.

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

[102] Richard J. Lipton. A New Approach To Information Theory. In *STACS*, pages 699–708, 1994.

[103] Mingjie Liu and Phong Q. Nguyen. Solving BDD by Enumeration: An Update. In *CT-RSA*, pages 293–309, 2013.

[104] Vadim Lyubashevsky and Daniel Masny. Man-in-the-Middle Secure Authentication Schemes from LPN and Weak PRFs. Cryptology ePrint Archive, Report 2013/092, 2013. `http://eprint.iacr.org/`.

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

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

[107] F.J. Macwilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North Holland, January 1983.

[108] J. L. Massey. Some Applications of Coding Theory in Cryptography. In *Codes and Ciphers: Cryptography and Coding IV*, pages 33–47, 1995.

[109] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In *ASIACRYPT*, pages 107–124, 2011.

[110] Robert J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *DSN Progress report*, pages 114–116, 1978.

[111] Robert J. McEliece and Dilip V. Sarwate. On Sharing Secrets and Reed-Solomon Codes. *Commun. ACM*, 24(9):583–584, 1981.

[112] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal Error Correction Against Computationally Bounded Noise. In *TCC*, pages 1–16, 2005.

[113] 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.

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

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

[116] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, pages 700–718, 2012.

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

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

[119] Petros Mol and Stefano Tessaro. Secret-Key Authentication Beyond the Challenge-Response Paradigm: Definitional Issues and New Protocols, 2013. Available at http://cseweb.ucsd.edu/~pmol/Publications/auth.pdf.

[120] Petros Mol and Scott Yilek. Chosen-Ciphertext Security from Slightly Lossy Trapdoor Functions. In *Public Key Cryptography*, pages 296–311, 2010.

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

[122] Jorge Munilla and Alberto Peinado. HB-MP: A Further Step in the HB-Family of Lightweight Authentication Protocols. *Computer Networks*, 51(9):2262–2267, 2007.

[123] Steven Myers and Abhi Shelat. Bit Encryption Is Complete. In *FOCS*, pages 607–616, 2009.

[124] Moni Naor and Moti Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC*, pages 427–437, 1990.

[125] Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *CRYPTO*, pages 31–53, 1992.

[126] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private Locally Decodable Codes. In *ICALP*, pages 387–398, 2007.

[127] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB# Against a Man-in-the-Middle Attack. In *ASIACRYPT*, pages 108–124, 2008.

[128] Raphael Overbeck. Statistical Decoding Revisited. In *ACISP*, pages 283–294, 2006.

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

[130] 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.

[131] Chris Peikert and Brent Waters. Lossy Trapdoor Functions and Their Applications. Latest Version, Available from `http://www.cc.gatech.edu/~cpeikert/`, October 5, 2009.

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

[133] Chris Peikert and Brent Waters. Lossy Trapdoor Functions and Their Applications. *SIAM J. Comput.*, 40(6):1803–1844, 2011.

[134] Krzysztof Pietrzak, Alon Rosen, and Gil Segev. Lossy Functions Do Not Amplify Well. In *TCC*, pages 458–475, 2012.

[135] Eugene Prange. The Use of Information Sets in Decoding Cyclic Codes. *IRE Transactions on Information Theory*, September 1962.

[136] M. O. Rabin. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. Technical report, Massachusetts Institute of Technology, 1979.

[137] Charles Rackoff and Daniel R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO*, pages 433–444, 1991.

[138] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *SIAM J. Comput.*, 8(2):300–304, 1960.

[139] 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.

[140] Oded Regev. On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. *J. ACM*, 56(6), 2009.

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

[142] Alon Rosen and Gil Segev. Chosen-Ciphertext Security via Correlated Products. IACR ePrint Archive, Report 2008/116.

[143] Alon Rosen and Gil Segev. Efficient lossy trapdoor functions based on the composite residuosity assumption. IACR ePrint Archive, Report 2008/134.

[144] Alon Rosen and Gil Segev. Chosen-Ciphertext Security via Correlated Products. In *Proceedings of the Sixth Theory of Cryptography Conference – TCC 2009*, pages 419–436. Springer, 2009.

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

[146] Amit Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS*, pages 543–553, 1999.

[147] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[148] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, November 1979.

[149] Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 1948.

[150] Richard C. Singleton. Maximum Distance q-nary Codes. *IEEE Transactions on Information Theory*, 10:116–118, April 1964.

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

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

[153] Jacques Stern. A Method for Finding Codewords of Small Weight. In *Coding Theory and Applications*, pages 106–113, 1988.

[154] Jacques Stern. A New Identification Scheme Based on Syndrome Decoding. In *CRYPTO*, pages 13–21, 1993.

[155] Jacques Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[156] Amnon Ta-Shma, David Zuckerman, and Shmuel Safra. Extractors from Reed-Muller Codes. *J. Comput. Syst. Sci.*, 72(5):786–812, 2006.

[157] Luca Trevisan. Extractors and Pseudorandom Generators. *J. ACM*, 48(4):860–879, 2001.

[158] Serge Vaudenay. On Privacy Models for RFID. In *ASIACRYPT*, pages 68–87, 2007.

[159] Hoeteck Wee. Efficient Chosen-Ciphertext Security via Extractable Hash Proofs. In *CRYPTO*, pages 314–332, 2010.

[160] Hoeteck Wee. Dual Projective Hashing and Its Applications - Lossy Trapdoor Functions and More. In *EUROCRYPT*, pages 246–262, 2012.

[161] Aaron D. Wyner. The Wire-tap Channel. *Bell System Technical Journal*, 54:1355–1387, 1975.