Karlsruhe Institute of Technology

Cryptography based on the Hardness of Decoding

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Nico Marcel Döttling

aus Heilbronn

Tag der mündlichen Prüfung:	08.05.2014
Erster Gutachter:	Prof. Dr. Jörn Müller-Quade
Zweiter Gutachter:	Prof. Daniel Wichs, PhD

Acknowledgement

First and foremost, I would like to thank my advisor Jörn for his guidance, his optimism and kindness. I am grateful for many good times we had over the years. We had inspiring discussions, from which eventually the ideas that solidified into this thesis emerged. He sparked my curiosity in coding based cryptography and helped me find interesting problems in this field. Moreover, I'd like to thank Anderson Nascimento for sharing his ideas and collaborating with us, which lead to a significant part of this thesis. I would like to address special thanks to Daniel Wichs for taking interest in my work and accepting to co-referee this thesis.

I was glad to work with some colleagues who made my time at work memorable, enjoyable and entertaining. In particular, I enjoyed the time working with my former colleague Daniel Kraschewski, to whom I owe a good deal of insights and technical tricks. Moreover, I enjoyed working with my long time friend Matthias Huber and my *roommate* Tobias Nilges, who provided me with help, support and friendship.

I want to thank IBM R&D Germany, in particular Reinhard Bündgen, Ulrich Mayer and Ingrid Schwarz-Hartmann for three years of rewarding cooperation.

Finally and most importantly, my biggest thanks goes to my wife Sónia and my son Friedrich. For their love and support during times of hardship.

The plots in this thesis were created using the Python Scipy toolbox [JOP+01] and the Matplotlib environment [Hun07].

Abstract

Public key cryptography is like magic. It allows two people who have never met before to communicate privately over any public channel. Since its conception in the 1970s, public key cryptography has become indispensable for the modern day connected world. Public key cryptography is essential for the `https` protocol, e-commerce, online auctions, online elections and many more. Computational hardness makes public key cryptography work. However, there is only a handful of hardness assumptions known which suffice for the construction of public key cryptosystems.

Early public key cryptosystems like RSA and ElGamal are based on number-theoretic problems, such as the factoring problem and discrete logarithm problem. By their nature, these problems are highly *structured*. While it is the structure of these problems that enables the construction of public key cryptosystems in the first place, this structure also gives rise to highly non-trivial attacks. As a consequence, all public key cryptosystems based on number-theoretic assumptions can be efficiently broken by quantum computers and there are very few candidates that resist subexponential-time classical attacks. This fact raised concerns about the hardness of these problems.

As a promising alternative to number-theoretic hardness assumptions, coding and lattice-based hardness-assumptions have emerged. Most prominent among these are the *learning parity with noise* (LPN) and the *learning with errors* (LWE) problem. Such problems appear naturally in coding-theory and have resisted more than 50 years of algorithmic/cryptanalytic efforts, both classically and quantumly.

This thesis provides progress for both LPN- and LWE-based cryptography. As main contribution of this thesis, we provide two constructions of adaptively secure public key cryptosystems. Adaptive chosen-ciphertext (IND-CCA2) security is the gold-standard of security definitions for public key cryptography. IND-CCA2 secure cryptosystems must withstand attacks by an adversary having access to a decryption-oracle that decrypts every ciphertext except for a special challenge-ciphertext, for which the adversary is tasked with guessing its corresponding plaintext. Our first proposal is based on the McEliece assumption and the LPN problem. The second one is based solely based on the hardness of a *low noise* LPN problem. This construction was the first of its kind and answered a problem that has been open for 9 years.

The second contribution of this thesis regards the LWE problem. The most important feature of the LWE problem is its worst-case hardness guarantee. Simply put, this means that almost all instances of the problem are as hard as the hardest instance of the problem. Such a feature is highly desirable in cryptography, as it guarantees that a cryptosystem based on this problem has essentially no weak keys or weak ciphertexts. This worst case guarantee however is established using

a special gaussian error-distribution. It is thus a natural question to ask whether gaussians are necessary for security or just a proof-artifact. In this thesis we show that the latter is true: The hardness guarantees of the LWE problem are preserved if a uniform error-distribution on a small interval is used instead of a gaussian distribution. As gaussian sampling usually is the computationally heaviest step in LWE-based cryptosystems, this contribution can tremendously improve the performance of LWE-based cryptosystems while preserving their security-guarantees.

Zusammenfassung

Public Key Kryptographie ist wie Magie. Sie erlaubt zwei Unbekannten über öffentliche Kanäle vertraulich miteinander zu kommunizieren. Seit der Entwicklung der ersten Public Key Verschlüsselungsverfahren in den 1970er Jahren hat sich Public Key Kryptographie zu einem essentiellen Bestandteil der vernetzten Welt entwickelt. Public Key Kryptographie ist die Grundlage für das `https` Protokoll, E-Commerce, Onlineauktionen, Onlinewahlen und viele weitere Technologien.

Die Sicherheit kryptographischer Verfahren beruht auf schwierigen Berechnungsproblemen. Es sind aber nur wenige (vermutlich) schwierige Probleme bekannt mit welchen die Konstruktion von Public Key Verfahren gelungen ist. Frühe Public Key Verfahren wie das RSA oder das ElGamal Verfahren basieren auf zahlentheoretischen Problemen, wie beispielsweise dem Faktorisierungsproblem oder dem diskreten Logarithmen-Problem. Diese Probleme sind naturgemäß hoch strukturiert. Einerseits erlaubt es diese Struktur erst Public Key Verfahren zu konstruieren. Andererseits aber führt *zu viel* Struktur aber auch zu hochgradig nicht-trivialen Angriffen auf derartige Verfahren. Als negative Konsequenz dieses *Zuviels* an Struktur sind alle auf zahlentheoretischen Problemen basierenden Public Key Verfahren effizient durch Quantencomputer brechbar und für nur wenige von ihnen existieren keine klassischen Algorithmen mit sub-exponentieller Komplexität. Daher gibt es Zweifel an der tatsächlichen Schwierigkeit dieser Probleme.

Als eine vielversprechende Alternative zu zahlentheoretischen Problemen haben sich Probleme aus der Codierungs- und Gittertheorie etabliert. Die bekanntesten dieser Probleme sind das *Learning Parity with Noise* (LPN) Problem und das *Learning With Errors* (LWE) Problem. Diese Probleme leiten sich aus praktischen Problemen der Codierungstheorie ab und auch mehr als 50 Jahre Forschung haben bisher keine effizienten Algorithmen für diese Probleme hervorgebracht, weder klassische noch quantische Algorithmen.

Diese Dissertation stellt Ergebnisse aus dem Bereich der LPN und LWE basierten Kryptographie vor. Als Hauptbestandteil dieser Arbeit werden zwei Public Key Verschlüsselungsverfahren vorgestellt welche sicher gegen adaptive Chosen-Ciphertext Angriffe sicher sind. Adaptive Chosen-Ciphertext Sicherheit (IND-CCA2) ist der *Gold-Standard* der Sicherheitsbegriffe für Public Key Verschlüsselung. IND-CCA2 sichere Verfahren müssen Angriffen widerstehen, bei welchen sich ein Angreifer jedes beliebige Chiffrat, entschlüsseln lassen kann, bis auf ein Ziel-Chiffrat, dessen Inhalt er erraten soll. Das erste vorgestellte Verfahren beruht auf der sogenannten McEliece Annahme und dem LPN Problem. Das zweite Verfahren beruht lediglich auf einer Variante des LPN Problems mit *wenigen Fehlern*. Die zweite Konstruktion war die erste ihrer Art und löste ein Problem welches für mindestens 9 Jahre offen war.

Der zweite Beitrag dieser Arbeit beschäftigt sich mit dem LWE Problem. Das vielleicht wichtigste Merkmal des LWE Problems ist seine garantierte *Worst-Case*

Schwierigkeit. Einfach ausgedrückt bedeutet das dass fast alle Instanzen des Problems so schwer wie die schwierigste Instanz sind. Eine derartige Eigenschaft ist hochgradig erwünschenswert bei kryptographischen Problemen, denn sie schließt die Existenz von schwachen Schlüsseln und schwachen Chiffraten aus. Damit diese Worst-Case Garantie für das LWE Problem beweisbar gilt, muss es allerdings mit *normalverteilten* Fehlern instanziert werden. Daher stellt sich die natürliche Frage ob Normalverteilungen in diesem Zusammenhang wirklich notwendig für die Schwierigkeit des Problems sind, oder ob es sich bei Ihnen um ein *Beweisartefakt* handelt. Diese Arbeit zeigt dass der zweite Fall gilt: Die Schwierigkeit des LWE Problems bleibt erhalten falls einfachere Fehlerverteilungen als die Normalverteilung verwendet werden, namentlich eine Gleichverteilung auf einem kurzen Intervall. Da das Ziehen normalverteilter Fehler den berechnungsintensivsten Schritt von LWE basierten Kryptoverfahren darstellt, hat dieser Beitrag das Potential die Effizienz von LWE basierten Kryptosystemen enorm zu steigern während gleichzeitig die Worst-Case Sicherheitsgarantien erhalten bleiben.

Contents

1. Preamble

> A large part of mathematics which becomes useful developed with absolutely no desire to be useful, and in a situation where nobody could possibly know in what area it would become useful; and there were no general indications that it ever would be so. By and large it is uniformly true in mathematics that there is a time lapse between a mathematical discovery and the moment when it is useful; and that this lapse of time can be anything from 30 to 100 years, in some cases even more; and that the whole system seems to function without any direction, without any reference to usefulness, and without any desire to do things which are useful.
>
> John von Neumann

1.1. Motivation

Computational hardness is the foundation of modern cryptography. Since public key cryptography has been conceived more than 30 years ago, the study of candidate sources of computational hardness has been a central purpose of cryptographic research. A major source of hardness in cryptography are computational problems located in the realm of *number theory*. These include problems like the well known *factoring* problem or *discrete logarithm* problems in various cyclic groups. Number theoretic problems arise genuinely from pure math. They have accompanied public key cryptography from its very beginning and have highly contributed to the real world success of public key cryptography. Basically all *asymmetric* cryptographic schemes in practical use today, like the RSA cryptosystem [RSA78] or variations

of the Diffie-Hellman key exchange protocol [DH76] are based on the hardness of number theoretic problems.

However, public key cryptography suffered a foundational crisis in 1994, when Peter Shor [Sho94] discovered that every number theoretic problem used in cryptography can be efficiently solved by *quantum computers*. Conversely, this means that every cryptographic scheme based on number theoretic problems will be rendered insecure once *scalable quantum computers* become reality. This discovery sparked interest in cryptographic schemes that are based on problems that are not known to be tractable by quantum computers. The oldest proposal of a public key cryptosystem based on a problem not known to be tractable by quantum computers is the McEliece cryptosystem [McE78]. Unlike number theoretic problems, which arise from centuries old questions in pure math, McEliece's proposal is based on a problem of highly practical relevance: Recovering signals from noisy observations. In discrete math, such problems are called decoding problems and are generally posed as follows.

> Given an efficiently computable encoding function and an encoded message that is corrupted by random noise, recover the uncorrupted encoding.

The theory of linear codes studies encoding functions f with the algebraic property of linearity. This theory and its applications are of fundamental importance for every problem concerned with the reliable transmission of messages through unreliable channels. Coding theory has found applications way beyond its original purpose and is maybe the most useful tool in theoretical computer science [BFL90, BFLS91, AS92, FGL+96, Din06]. So how is it possible that coding theory is relevant for cryptography?

There has been a paradox in coding theory dating back to the earliest works in the field [Sha48, Gil52, Var57]. It has been known from the very beginning that almost every code is good in the sense that it allows the recovery of uncorrupted messages in principal. However, for arbitrary codes the best strategy known to recover the message is *brute forcing* through all the possibilities. Efficient decoding algorithms are only known for very limited classes of highly structured codes.

And this is where where McEliece's idea ties in with. Take an efficiently decodable code and make it appear as if it was an arbitrary or random code. Anyone who knows which code really conceals itself behind this random looking code will be able to decode. Anyone else will be faced by an intractable problem.

Linear codes are encoding schemes for discrete channels. Discrete channels are assumed to inflict errors that destroy certain parts of a message entirely but leave others intact. The error rate of such a channel bounds the number of locations at which the error will strike.

Continuous channels, such as radio channels, usually have a different error model. For continuous channels, the error is usually assumed to be limited in its *total power* [Sha48]. The power of an error naturally corresponds to an euclidean metric on the errors. Linear encoding schemes for continuous channels are called lattices. Thus, lattices can be seen as the euclidean analogue of linear codes. Interestingly, the theoretical study of lattices started long before linear codes were conceived. Lattices date back Minkowski's *geometry of numbers* [Min10] which studies convex bodies in high dimensional euclidean spaces.

Soon, decoding problems in lattices were also discovered as a source of computational hardness for cryptography. Decoding problems in lattices have a similar structure to their analogues in linear codes. However, their algebraically richer structure provides a much stronger leverage for cryptographic constructions. A unique feature of (certain) lattice based hardness assumptions is their worst-to-average case connection. For most cryptographic assumptions, including coding based assumptions, the hardness of an average case problem must be explicitly assumed. For certain lattice problems however, the hardness of average case problems can be based on the hardness of natural worst case problems. Among lattice problems with worst case hardness guarantee, the *learning with errors* (LWE) problem [Reg05] stands out.

Theoretical cryptography, or *provably secure* cryptography, claims that a win-win situation lies at its core. A cryptographic security proof basically states that one of the following two possibilities must be the case. The first possibility is that a cryptographic scheme based on a natural computational problem really is secure. The second possibility is that the scheme is insecure, which however would imply an efficient algorithm solving the said computational problem. For most problems used in cryptography this would constitute a major algorithmic breakthrough. In the case of number theoretic problems such a breakthrough would indeed be highly surprising.

But otherwise practically irrelevant[1]. Number theoretic problems basically have no practical purpose outside of cryptography. Coding based cryptography, on the other hand, was motivated by practical problems that apparently have no efficient solution. Thus, successful structural attacks against (provably secure) coding or lattice based cryptosystems will have an actual repercussion on the practical problems they were motivated by.

Besides number theoretic and lattice/coding based problems, various other problems have been suggested as sources of hardness for public key cryptography. Mentionable proposals were conjugation problems in braid groups [WM84, KLC+00] and multivariate polynomial equation systems [MI88, Pat96]. However, with very few exceptions all of these were eventually broken [RST07, KS99]. All remaining proposals are more or less variations of broken schemes.

From a practical point of view, lattice and coding based cryptography is very favorable with respect to implementation. Implementations of number theory based cryptographic schemes, such as the classical example RSA, usually require expensive long number arithmetic operations, like modular exponentiations. To make things worse, these operations are mostly resilient to parallelization[2]. Attempts to speed up expensive arithmetic operations may even introduce unexpected weaknesses into implementations. This was, for instance the case with CRT-RSA, where Chinese remaindering was used to speed up RSA decryption or signing on smart cards. This infamous performance tweak gave rise to the well known *fault induction attacks* [BDL97]. A fault induction attack disrupts the computation of a smartcard. With

[1] This opinion was not shared by Carl Friedrich Gauß [Gau66, Knu97]: "The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic... The dignity of science seems to demand that every aid to the solution of such an elegant and celebrated problem be zealously cultivated."

[2] Somewhat ironically, in the case of modular exponentiations this fact has even been turned into a hardness assumption by its own right. A time lock puzzles [RSW96] is a cryptographic bottle post into the future. Its security is based on the assumption that modular exponentiations cannot be sped up by parallelization

some luck, the flawed output of the smartcard then discloses the full secret key.

Coding based cryptography, on the other hand, relies mostly on simple binary arithmetic. All operations necessary are highly parallelizable. While the implementations of lattice based cryptosystems do require integer arithmetic and are thus not as efficient as coding based schemes, they usually do not require long number arithmetic. An oft mentioned drawback of lattice and coding based cryptography are the rather large key sizes. Due to the nature of the underlying decoding problems, the public and secret keys are matrices and their size grows quadratically. While this was a serious drawback still a few years ago, current technology may very well handle key sizes of several megabytes.

Chosen ciphertext security is the gold standard of security for public key encryption. In a nutshell, this security notion guarantees that no adversary may learn the contents of a specific ciphertext (for a given key), even if he is allowed to learn the contents of any other ciphertext (for the same key) of his choice. Obtaining efficient chosen ciphertext secure public key encryption schemes from standard hardness assumptions proved to be a tough nut for cryptographic research. While it was shown in the late 80s and early 90s that chosen ciphertext secure cryptosystems can be constructed from standard assumptions [NY90, RS91, DDN91], all these constructions involved very heavy theoretical machinery such as *non-interactive zero knowledge proofs* [BFM88b, BFM88a]. These results are nowadays considered mostly feasibility results, without relevance for practical applications. Nonetheless, these early constructions pointed the way ahead to efficient constructions and proved to be a valuable source of inspiration.

1.2. Contribution and Structure of this Thesis

While coding based cryptography has had major significance in the realm of symmetric cryptography, this work exclusively concerned with public key cryptography. Research in public key cryptography usually proceeds in two (mostly orthogonal) directions. The first direction seeks to explore new concepts, add new features to existing concepts and generally provide first instantiations of such concepts. Very often, new hardness assumptions are conjectured to reach this goal. The second direction tries to find more efficient instantiations of existing concepts and base them on a wider range of and possibly stronger hardness assumptions. This thesis is in line with the latter direction.

The better part of this thesis deals with the construction of chosen ciphertext secure public key encryption schemes. We will develop techniques to construct efficient chosen ciphertext secure public cryptosystems from coding based hardness assumptions. The centerpiece of of this thesis is Chapter 6, where we develop a chosen ciphertext secure public key cryptosystem based on the *learning parity with noise* assumption. We will now provide a brief summary of the chapters of this thesis.

- In Chapter 2 we will provide the technical background for the following Chapters. In particular, we will gather tools and notions from probability, public key cryptography, coding theory and lattices.

- In Chapter 3 we provide an introduction to coding and lattice based cryptography. We will introduce the main objects of study in this thesis, the *learning*

parity with noise (LPN) and *learning with errors* (LWE) problems and provide basic reductions between different flavors of the problems. As a minor contribution in this thesis, we present a result in Section 3.5 that provides a reduction from LPN with unbounded samples to LPN with bounded samples. Specifically, the result shows that if LPN with a square-root fraction of noise and a linear number of samples is hard, then LPN with a constant fraction of noise and unbounded samples is also hard. To the best of our knowledge this result is novel.

- In Chapter 4, we provide the framework for our results in Chapters 5 and 6. We discuss the notion of *tag-based encryption* and its purpose for the construction of chosen ciphertext secure public key encryption. This chapter contains no original contribution by the author and serves mostly the purpose of self-containedness.

- In Chapter 5 we present a public key cryptosystem based on McEliece's cryptosystem. The cryptosystem we propose is secure against adaptive chosen ciphertext attacks in the standard model. The scheme is an improvement of the scheme of Dowsley, Müller-Quade and Nascimento [DMQN09] and was originally published in the IEEE Transactions on Information Theory [DDMQN12].

- In Chapter 6 we present a public key cryptosystem based on a low noise learning parity with noise assumption. The cryptosystem is secure against adaptive chosen ciphertext attacks in the standard model. This scheme was the first that achieved this from this assumption and was originally published in Asiacrypt 2012 [DMQN12]. The scheme presented here is an improved version achieving better efficiency.

- In Chapter 7 we present a hardness reduction that bases the learning with errors problem with uniform errors on a standard problem in worst-case lattices. Prior hardness reductions for this problem needed to assume the hardness of non-standard worst case lattice assumptions. The result was originally published in Eurocrypt 2013 [DMQ13].

2. Prerequisites

If I have seen further it is by
standing on ye sholders of Giants.

Sir Issac Newton

2.1. Pseudocode Notation

We will generally present algorithms and experiments in pseudocode notation. We will give a brief overview of the notation to be used.

- **Assignents:** To denote the assignment of a value b to a variable a we will use the notation $a \leftarrow b$. Moreover, if $\mathsf{B}()$ is an algorithm, we will use $a \leftarrow \mathsf{B}()$ to indicate that a is being assigned the output of $\mathsf{B}()$.

- **Random Choices:** If $\mathcal{B}$ is a distribution or an algorithm that samples a distribution, we will use the notation $a \leftarrow_{\$} \mathcal{B}$ to denote that a is being assigned a sample of $\mathcal{B}$. If S is a finite set, we will use the notation $a \leftarrow_{\$} S$ to indicate that the value assigned to a was drawn uniformly from S.

- **Loops and Conditional Statements** We will use *For* loops and *If* statements in the standard way. We will allow *For* loops to run over sets, e.g. we will use *For* loops of the form *For* $x \in S$ to denote that x runs through all the elements in the set S.

- **Object-oriented Notation:** To denote that a certain algorithm A belongs to a scheme S (e.g. an encryption or signature scheme), we will use the notation $\mathsf{S.A}()$ for invocations of A.

- **Oracle-Access:** If $\mathsf{A}()$ is an algorithm that requires access to an oracle and $\mathcal{O}()$ is an implementation of such an oracle, we will write $\mathsf{A}^{\mathcal{O}(\cdot)}()$ to denote that A can query $\mathcal{O}$ with inputs of its choice during execution. If $\mathcal{O}$ takes two (or more) arguments, we will use the notation $\mathsf{A}^{\mathcal{O}(a_1,\cdot)}()$ to denote that A has access to $\mathcal{O}$ with the first argument hardwired to a_1.

- **Parsing:** To denote that a certain object is semantically split into several components we will use the keyword *Parse*. For instance, to denote that A is tuple of elements a, b, c we will write *Parse* $A = (a, b, c)$.

2.2. Notation and Basic Math

We will denote the finite field with q elements by $\mathbb{F}_q$. Vectors $\mathbf{x}$ and matrices $\mathbf{A}$ will be written boldface while scalars y will be written regular. We will use $\|\mathbf{x}\|_p$ to denote the L_p norm of a vector $\mathbf{x}$ and $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote the inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$. We will generally assume that elements of the residue class ring $\mathbb{Z}_q$ are given in the central residue-class representation, i.e. if $x' \in \mathbb{Z}_q$, we will identify $x' = x \mod q$ with an integer x in $\{-\lfloor q/2 \rfloor, \ldots, \lceil q/2 \rceil - 1\}$. We can thus generically lift x' from $\mathbb{Z}_q$ to $\mathbb{Z}$. Moreover, with this we can define a meaningful *norm* on $\mathbb{Z}_q$ by $\|\mathbf{x} \mod q\| = \|\mathbf{x}\|$. Horizontal concatenations two matrices $\mathbf{A}$ and $\mathbf{B}$ will be denoted by $(\mathbf{A}\|\mathbf{B})$. To denote the vertical concatenation of $\mathbf{A}$ and $\mathbf{B}$ we will write $(\mathbf{A}^T\|\mathbf{B}^T)^T$.

The following simple (and commonly known) lemma will provide useful estimates in many situations.

Lemma 2.1. *It holds for all $x \geq 0$ that*

$$1 - x \leq e^{-x}.$$

Moreover, if $x \in [0, \frac{1}{2}]$, then

$$e^{-2x} \leq 1 - x$$

Proof. By the mean value theorem (see e.g. [Wei]), for every $x \geq 0$ there exists an $x' \in [0, x]$ with

$$e^{-x} = 1 - e^{-x'}x \geq 1 - x$$

as $e^{-x'}$ assumes its maximum on $[0, x]$ at $x' = 0$. For the second statement, apply the mean value theorem to $\ln(1 - x)$ and we get that for every $x \in [0, \frac{1}{2}]$ there exists an $x' \in [0, x]$ such that

$$\ln(1 - x) = -\frac{1}{1 - x'}x \geq -2x,$$

as $-\frac{1}{1-x'}$ assumes its minimum on $[0, \frac{1}{2}]$ at $x' = \frac{1}{2}$. Consequently,

$$1 - x \geq e^{-2x}$$

for $x \in [0, \frac{1}{2}]$. □

2.3. Probability, Combinatorics and Information

In this section we introduce the stochastic notions needed in this thesis. We will assume the underlying probability spaces for the random variables we use to be implicitly given. For an event E we will denote the probability of E as $\Pr[E]$. Discrete random variables X will be defined by the probability mass function $\Pr[X = x]$ corresponding to their distribution. Continuous random variables X will be given by their probability density function $p_X(x)$ of their distribution. For a discrete random variable X defined over a set $\mathcal{X} \subseteq \mathbb{R}$, the expectation of X is defined by

$$\mathsf{E}[X] = \sum_{x \in \mathcal{X}} \Pr[X = x] \cdot x.$$

If X is a continuous random variable with density-function p_X, then $\mathsf{E}[X]$ is defined as

$$\mathsf{E}[X] = \int_{x \in \mathbb{R}} p_X(x) \cdot x \, \mathrm{d}x,$$

if this integral converges. An important property of the expectation is its linearity, i.e. it holds for arbitrary random variables X and Y and constants $\alpha, \beta \in \mathbb{R}$ that $\mathsf{E}[\alpha X + \beta Y] = \alpha \mathsf{E}[X] + \beta \mathsf{E}[Y]$. The statistical distance of two discrete random variables X and Y defined over a common domain $\mathcal{X}$ is defined by

$$\Delta(X, Y) = \frac{1}{2} \sum_{t \in \mathcal{X}} |\Pr[X = t] - \Pr[Y = t]|.$$

We will identify the uniform distribution on sets S with S itself, i.e. we will write

$$x \leftarrow_{\$} S$$

to denote that x is drawn uniformly from S or that the random variable x follows the uniform distribution on S.

We will now introduce several probability distributions needed in this thesis.

Definition 2.1. *Let $\rho \in [0, 1]$. Let X be a random variable with*

$$\Pr[X = x] = \begin{cases} 1 - \rho & \text{if } x = 0 \\ \rho & \text{if } x = 1. \end{cases}$$

We say that X follows the Bernoulli distribution $\mathsf{Ber}(\rho)$. If $0 \leq \delta \leq \frac{1}{2}$ and X follows $\mathsf{Ber}(\frac{1}{2} - \delta)$, we also say that X is a δ-biased coin. If $\mathbf{x} = (x_1, \ldots, x_n)$ where the x_i are independent an each x_i follows $\mathsf{Ber}(\rho)$, then we say that $\mathbf{x}$ follows $\mathsf{Ber}(m, \rho)$. Moreover, we denote the component-wise independent Bernoulli distribution on $m \times n$ matrices by $\mathsf{Ber}(m \times n, \rho)$.

The Bernoulli distribution with parameter ρ models a biased cointoss which takes outcome 0 with probability $1 - \rho$ and outcome 1 with probability ρ. The following elementary lemma describes the distribution of the sum of Bernoulli distributed random variables modulo 2.

Lemma 2.2. *Let $x, y \in \mathbb{F}_2$ be independently distributed where x follows $\mathsf{Ber}(\frac{1}{2} - \delta_1)$ and y follows $\mathsf{Ber}(\frac{1}{2} - \delta_2)$. Then $x + y$ follows $\mathsf{Ber}(\frac{1}{2} - 2\delta_1\delta_2)$. Moreover, if $x_1, \ldots, x_n \in \mathbb{F}_2$ are independently distributed according to $\mathsf{Ber}(\frac{1}{2} - \delta)$, then the sum $\sum_{i=1}^{n} x_i$ follows $\mathsf{Ber}(\frac{1}{2} - \frac{1}{2}(2\delta)^n)$.*

Proof. Let $x \leftarrow_{\$} \mathsf{Ber}(\frac{1}{2} - \delta_1)$ and $y \leftarrow_{\$} \mathsf{Ber}(\frac{1}{2} - \delta_2)$ be drawn independently and interpreted as elements of $\mathbb{F}_2$. It holds that

$$\begin{aligned} \Pr[x + y = 1] &= \Pr[x = 1 \text{ and } y = 0] + \Pr[x = 0 \text{ and } y = 1] \\ &= \Pr[x = 1] \Pr[y = 0] + \Pr[x = 0] \Pr[y = 1] \\ &= \left(\frac{1}{2} - \delta_1\right)\left(\frac{1}{2} + \delta_2\right) + \left(\frac{1}{2} + \delta_1\right)\left(\frac{1}{2} - \delta_2\right) \\ &= \frac{1}{2} - 2\delta_1\delta_2. \end{aligned}$$

Thus $x + y$ follows $\mathsf{Ber}(\frac{1}{2} - 2\delta_1\delta_2)$. If $x_1, \ldots, x_n$ are independently distributed by $\mathsf{Ber}(\frac{1}{2} - \delta)$, then applying the first statement inductively yields

$$\Pr[\sum_{i=1}^{n} x_i = 1] = \frac{1}{2} - \frac{1}{2}(2\delta)^n$$

and thus $\sum_{i=1}^{n} x_i$ follows $\mathsf{Ber}(\frac{1}{2} - \frac{1}{2}(2\delta)^n)$. □

Binomial distributions model the sum (in $\mathbb{Z}$) of independently distributed Bernoulli trials.

Definition 2.2. *Let $\rho \in [0, 1]$. Let $X_1, \ldots, X_n$ be independent Bernoulli trials. Then $X = \sum_{i=1}^{n} X_i$ follows the binomial distribution $\mathsf{Bin}(m, \rho)$. X has the probability mass-function*

$$\Pr[X = x] = \sum_{i=0}^{x} \binom{n}{i} \rho^i (1 - \rho)^{n-i}.$$

for $x \in \{0, \ldots, n\}$ and $\Pr[X = x] = 0$ otherwise.

2.3.1. Concentration of Measure

An important tool to bound binomial distributions is the Chernoff-Hoeffding bound.

Theorem 2.1 (Multiplicative Chernoff-Hoeffding bound [Hoe63])**.** *Let $\rho \in [0, 1]$. Let $X_1, \ldots, X_n$ be independent Bernoulli trials distributed according to $\mathsf{Ber}(\rho)$ and let $X = \sum_{i=1}^{n} X_i$. Then it holds for any $\beta > 0$ that*

$$\Pr[X \geq (1 + \beta)\rho n] \leq e^{-\frac{\beta^2}{3}\rho n}$$

and

$$\Pr[X \leq (1 - \beta)\rho n] \leq e^{-\frac{\beta^2}{2}\rho n}.$$

For an elementary proof of Theorem 2.1 see e.g. [Hoe63, MU05].

2.3.2. Combinatorics

Definition 2.3. *For $x \in (0, 1)$ the binary entropy function is defined by*

$$H(x) = -x \log x - (1 - x) \log(1 - x).$$

We set $H(0) = 0$ and $H(1) = 0$. We define the q-ary entropy function by

$$H_q(x) = \frac{x \log(q - 1) + H(x)}{\log(q)}.$$

On the interval $[0, 1 - 1/q]$ the q-ary entropy function is injective. Thus, the inverse q-ary entropy function $H_q^{-1} : [0, 1] \to [0, 1 - 1/q]$ is well defined.

When x is very small (i.e. asymptotically converging to 0) it can be useful to apply the following estimate for $H(x)$, which follows from Lemma 2.1.

Corollary 2.3. *For $x \in [0, \frac{1}{2}]$ it holds that*

$$x \log \frac{1}{x} \leq H(x) \leq x \left(\log \frac{1}{x} + \frac{2}{\ln 2} \right)$$

Lemma 2.4. *For every $\epsilon > 0$ and every $x \in (0, 1/2)$ it holds that*

$$H((1+\epsilon)x) \leq (1+\epsilon)H(x).$$

Proof. Basic calculation shows that

$$H'(x) = -\log(x) + \log(1-x).$$

By the mean value theorem ([Wei]) it holds that

$$H((1+\epsilon)x) = H(x+\epsilon x) = H(x) + H'(x_0)\epsilon x$$

for some $x_0 \in (x, (1+\epsilon)x)$. $H'(x_0)$ assumes its maximum at $x_0 = x$ in this interval. Thus it holds that

$$\begin{aligned} H((1+\epsilon)x) &= H(x) + H'(x_0)\epsilon x \\ &\leq H(x) + H'(x) \cdot x \cdot \epsilon \\ &= H(x) + (-x\log(x) + x\log(1-x))\epsilon \\ &= H(x) + (H(x) + \log(1-x))\epsilon \\ &= (1+\epsilon)H(x) + \epsilon\log(1-x) \\ &\leq (1+\epsilon)H(x), \end{aligned}$$

as $\epsilon \log(1-x) \leq 0$. □

The following lemma shows how binomial coefficients can be approximated using the binary entropy function.

Lemma 2.5. *(see e.g. [MU05]) Let n, k be integers with $0 \leq k \leq n$. It holds that*

$$\frac{2^{n \cdot H(\frac{k}{n})}}{n+1} \leq \binom{n}{k} \leq 2^{n \cdot H(\frac{k}{n})}$$

We will need to draw binary matrices uniformly at random and bound the probability that they have full rank. The following lemma provides a useful estimate.

Lemma 2.6. *Let m, n be positive integers with $m \geq n$. Let $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m \times n}$ be chosen uniformly at random. Then it holds that*

$$\Pr[\mathbf{A} \textit{ has full rank}] = \prod_{i=0}^{n-1} \left(1 - 2^{i-m}\right) \geq 1 - 2^{n-m+1}.$$

Proof. We will count the number of matrices $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ with full rank. For the first column $\mathbf{a}_1$ of $\mathbf{A}$, we can choose any vector in $\mathbb{F}_2^m \backslash \{0\}$, For the second column $\mathbf{a}_2$ of $\mathbf{A}$, we can choose any vector in $\mathbb{F}_2^m \backslash \mathsf{span}(\mathbf{a}_1)$. Inductively, for the i-th column $\mathbf{a}_i$ of $\mathbf{A}$, we can choose any vector in $\mathbb{F}_2^m \backslash \mathsf{span}(\mathbf{a}_1, \ldots, \mathbf{a}_{i-1})$. As $\mathsf{span}(\mathbf{a}_1, \ldots, \mathbf{a}_{i-1})$ is a vector space of dimension $i-1$, it holds that $|\mathbb{F}_2^m \backslash \mathsf{span}(\mathbf{a}_1, \ldots, \mathbf{a}_{i-1})| = 2^m - 2^{i-1}$. Consequently, there are

$$\prod_{i=1}^{n} \left(2^m - 2^{i-1}\right) = 2^{mn} \prod_{i=0}^{n-1} \left(1 - 2^{i-m}\right)$$

full rank matrices in $\mathbb{F}_2^{m\times n}$. Therefore, the probability that a uniformly chosen matrix $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m\times n}$ has full rank is

$$\Pr[\mathbf{A} \text{ has full rank}] = \frac{2^{mn}\prod_{i=0}^{n-1}(1-2^{i-m})}{2^{mn}} = \prod_{i=0}^{n-1}\left(1-2^{i-m}\right),$$

as $|\mathbb{F}_2^{m\times n}| = 2^{mn}$. We will now provide a lower bound for $\Pr[\mathbf{A} \text{ has full rank}]$. For $i \leq n-1$, we have that $2^{i-m} \leq \frac{1}{2}$, as $n \leq m$. Consequently, by Lemma 2.1 we can bound

$$1 - 2^{i-m} \geq e^{-2\cdot 2^{i-m}}.$$

This yields

$$\begin{aligned}
\Pr[\mathbf{A} \text{ has full rank}] &= \prod_{i=0}^{n-1}\left(1-2^{i-m}\right) \\
&\geq \prod_{i=0}^{n-1} e^{-2\cdot 2^{i-m}} \\
&= e^{-2\sum_{i=0}^{n-1} 2^{i-m}} \\
&= e^{-2\cdot 2^{-m}\cdot(2^n-1)} \\
&\geq 1 - 2\cdot 2^{-m}\cdot(2^n-1) \\
&\geq 1 - 2^{n-m+1}
\end{aligned}$$

□

2.3.3. Min-Entropy

The min-entropy of a random variable X measures the amount of *worst case* randomness of X. For random variables with high min-entropy, no single outcome has too high probability, i.e. the probability mass function of such variables is not pointy.

Definition 2.4. *Let χ be a probability distribution with finite support and let X be distributed according to χ. Define the* min-entropy $H_\infty(X)$ *by*

$$H_\infty(X) = -\log(\max_\xi(\Pr[X=\xi])).$$

Let Y be random-variable (possibly correlated with X) and let $\tilde{y}$ be a measurement or outcome of Y. The conditional min-entropy $H_\infty(X|Y=\tilde{y})$ *is defined by*

$$H_\infty(X|Y=\tilde{y}) = -\log(\max_\xi(\Pr[X=\xi|Y=\tilde{y}])).$$

2.4. Cryptographic Notions

In this section we will provide the cryptographic notions and fundamental results required in this thesis.

2.4.1. General

Results in complexity theory and theoretical cryptography are usually stated in an asymptotic manner. The main parameter that controls the asymptotic security of cryptographic constructions is called the security parameter. Throughout this thesis, we will reserve the variable λ to denote a security parameter. We will use the following standard asymptotic notations to denote the asymptotic behavior of functions. Let f and g be functions $\mathbb{N} \to \mathbb{R}$. Denote

- $f(\lambda) = O(g(\lambda))$ if there exists a constant $c > 0$ and an $n_0 \in \mathbb{N}$ such that it holds for all $n > n_0$ that $f(n) \leq c \cdot g(n)$.
- $f(\lambda) = \tilde{O}(g(\lambda))$ if there exist a constants $c_1 > 0$, $c_2 \geq 0$ and an $n_0 \in \mathbb{N}$ such that it holds for all $n > n_0$ that $f(n) \leq c_1(\log(n))^{c_2} \cdot g(n)$.
- $f(\lambda) = o(g(\lambda))$ if for every constant $c > 0$ there exists an $n_0 \in \mathbb{N}$ such that it holds for all $n > n_0$ that $f(n) < c \cdot g(n)$.
- $f(\lambda) = \Omega(g(\lambda))$ if there exists a constant $c > 0$ and an $n_0 \in \mathbb{N}$ such that it holds for all $n > n_0$ that $f(n) \geq c \cdot g(n)$.
- $f(\lambda) = \omega(g(\lambda))$ if for every constant $c > 0$ there exists an $n_0 \in \mathbb{N}$ such that it holds for all $n > n_0$ that $f(n) > c \cdot g(n)$.
- $f(\lambda) = \mathsf{poly}(\lambda)$ if there exists a constant $c > 0$ such that $f(\lambda) = O(\lambda^c)$.

The notion of negligible functions is essential for the definition of the security of cryptographic schemes. Informally speaking, a function is called negligible if it vanishes faster than the inverse of any polynomial.

Definition 2.5. *We say a function $f : \mathbb{N} \to \mathbb{R}$ is negligible, if for every constant $c > 0$ there exists an $n_0 \in \mathbb{N}$ such that for all $n > n_0$ it holds that $f(n) < \frac{1}{n^c}$. This condition can be equivalently expressed as $f(\lambda) = \lambda^{-\omega(1)}$. We will write shorthand $\mathsf{negl}(\lambda)$ to denote an unspecified negligible function. We will call functions of the form $1 - \mathsf{negl}(\lambda)$ overwhelming. Moreover. we will call functions of the form $\Omega(\lambda^{-c})$ (for a constant $c > 0$) substantial or noticeable.*

The notions of negligible and noticeable functions are robust under polynomial changes. More precisely, for any constant $c > 0$ if $f(\lambda)$ is negligible, then so is $\mathsf{poly}(\lambda) \cdot (f(\lambda))^c$ and if $f(\lambda)$ is noticeable, then so is $(f(\lambda))^c/\mathsf{poly}(\lambda)$.

As is usual in theoretical computer science, we consider machines/algorithms efficient if they run in probabilistic polynomial time (PPT).

A standard way of defining security for cryptographic schemes are *game-based* security definitions [Nao03, Pas11]. Game-based security definitions are modeled as an interaction between a challenger C and an adversary $\mathcal{A}$. We will represent $\mathcal{A}$ as a multi-stage algorithm, which outputs a state st after each stage and receives this state as input in its next stage. Both the challenger and the adversary receive as part of their first input an *unary* encoding 1^λ of the security parameter. Then several rounds of interaction between the challenger and the adversary may follow, in which the challenger may provide access to additional resources to the adversary. After the end of the interaction the challenger computes an output which is either 1 or 0. If the output is 1, we say the adversary wins the game/experiment, if it is 0 he loses. Let $\langle C(1^\lambda), \mathcal{A}(1^\lambda)\rangle$ denote C's output after interacting with $\mathcal{A}$. We consider a scheme secure if the success probability

$$\mathsf{Succ}_C(\mathcal{A}) = \Pr[\langle C(1^\lambda), \mathcal{A}(1^\lambda)\rangle = 1]$$

of any PPT adversary $\mathcal{A}$ is at most negligibly better than what can be achieved by a trivial strategy (e.g. guessing the result). In such security definitions, the adversary is usually given more resources than one would expect a real live adversary to have.

This is primarily done to model certain parts of the context in which the protocol is designed to be used as part of the adversary.

A special case of security experiments are indistinguishability experiments. We say that two distributions A and B depending on the security parameter λ are *statistically close*, if

$$\Delta(A, B) \leq \mathsf{negl}(\lambda).$$

Moreover, we say that A and B are computationally indistinguishable, if for every PPT algorithm $\mathcal{D}$ the *advantage*

$$\mathsf{Adv}(\mathcal{D}) = |\Pr[\mathcal{D}(A) = 1] - \Pr[\mathcal{D}(B) = 1]|$$

is at most negligible, i.e. $\mathsf{Adv}(\mathcal{D}) \leq \mathsf{negl}(\lambda)$. We usually call the algorithm $\mathcal{D}$ distinguisher. We say that a distribution A is *pseudorandom*, if it is computationally indistinguishable from a uniform distribution U.

Unconditional security proofs for most cryptographic tasks would immediately imply $\mathsf{P} \neq \mathsf{NP}$. Thus, security of cryptographic schemes is usually proven relative to certain computational hardness assumptions. Such a security proof provides a reduction from the security property to be proven to an established computational problem which is conjectured to be hard. In such reductions we show how a successful adversary against the security of the cryptographic scheme (defined via a security experiment) can be used to construct an efficient algorithm solving the problem assumed to be hard. This leads to the desired contradiction and we conclude that the scheme fulfills the security property.

For more complex proofs, we will use a technique called game transform. The idea of this technique is to slightly modify the security game on a step-by-step approach, where we show for each step that the success probability of the adversary differs only by a negligible amount for successive games. The goal of such a transformation is usually to transform a complex security game into a simple security game which has a more or less elementary security proof. To signify differences in successive games we will highlight the affected parts of the game. In the subsections to follow we will provide game based security notions for the cryptographic tasks relevant in this thesis.

2.4.2. The Goldreich Levin Hardcore Predicate

An important tool to establish the pseudorandomness of certain distributions is the Goldreich Levin [GL89] hardcore predicate.

Theorem 2.2 (Goldreich Levin [GL89])**.** *Let λ be a security parameter. Let $m, n, k = \mathsf{poly}(\lambda)$. Let $f : \mathbb{F}_2^n \times \mathbb{F}_2^k \to \mathbb{F}_2^m$ be an efficiently computable injective function. Let X be a distribution on $\mathbb{F}_2^n$ and Y be a distribution on $\mathbb{F}_2^m$. Let $\mathbf{x} \leftarrow_\$ X$, $\mathbf{y} \leftarrow_\$ Y$ and $\mathbf{z} = f(\mathbf{x}, \mathbf{y})$. Let further $\mathbf{r} \leftarrow_\$ F_2^n$ and $b = \langle \mathbf{r}, \mathbf{x} \rangle$. Assume there exists an efficient algorithm $\mathcal{A}$ that computes b given $\mathbf{r}$ and $\mathbf{z}$ with probability non-negligibly better than $\frac{1}{2}$. Then there exists an efficient algorithm $\mathcal{A}'$ that computes $\mathbf{x}$ given $\mathbf{z}$ with non-negligible probability.*

2.4.3. Public Key Encryption

Public Key Cryptography started with the seminal works of Diffie and Hellman [DH76] and Rivest, Shamir and Adleman [RSA78]. In a public key encryption scheme, a receiver Bob generates a pair of private and public keys and publishes

the public key. A sender Alice who wants to securely transmit a message to Bob can encrypt it using Bob's public key key. After receiving the encrypted message, Bob will be able to decrypt the message using his secret key. Before discussing the security requirements for public key encryption, we will provide the syntactical definition for public key encryption schemes.

Definition 2.6. *A public key encryption scheme* PKE *consists of three PPT-algorithms* $\mathsf{PKE.KeyGen}$, $\mathsf{PKE.Enc}$ *and* $\mathsf{PKE.Dec}$, *such that the following syntactical requirements are met.*

- $\mathsf{PKE.KeyGen}(1^\lambda)$ *is a PPT-algorithm that takes a security-parameter* λ *and outputs a pair of public and private keys* (pk, sk).
- $\mathsf{PKE.Enc}(pk, \mathfrak{m})$ *is a PPT-algorithm that takes a public key* pk, *a message* $\mathfrak{m}$ *and outputs a ciphertext* $\mathfrak{c}$.
- $\mathsf{PKE.Dec}(sk, \mathfrak{c})$ *is a PPT-algorithm taking as input a secret key* sk *and a ciphertext* $\mathfrak{c}$ *and outputs a plaintext* $\mathfrak{m}$.

A standard-requirement for any public key encryption scheme is completeness, i.e. the receiver Bob will be able to decrypt messages encrypted by a sender Alice.

Definition 2.7. *We say that* $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *is complete, if it holds for all plaintexts* $\mathfrak{m}$ *that*

$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, \mathfrak{m})) \neq \mathfrak{m} : (pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)] < \mathsf{negl}(\lambda).$$

The early candidates for public key encryption schemes (such as [RSA78, McE78]) were all constructed to (implicitly) meet the notion of one-way CPA security. This notion basically requires that encryption is one-way, i.e. it is infeasible to recover the randomly chosen plaintext entirely given only the ciphertext and public key. This notion has obvious shortcomings. First, no guarantee is given if the ciphertext is not chosen from a uniform or almost uniform distribution. Second, an encryption scheme which does not encrypt parts of the plaintext at all can be considered secure under this security notion, as this notion only guarantees that no adversary can recover the *entire* plaintext message.

Goldwasser and Micali [GM82] were the first authors to come up with a security definition for public key encryption which captures the intuition that a ciphertext should basically reveal nothing about the plaintext it encrypts in a rigorous manner. They introduced the notion of semantic security, called ciphertext indistinguishability under chosen plaintext attacks (IND-CPA security) in later works [MRS86, BDPR98]. Semantic security is modeled as the following game between an experiment and the adversary. The experiment first generates a pair of public and private keys (pk, sk). After that, the public key pk is given to the adversary $\mathcal{A}$. The adversary now chooses two messages $\mathfrak{m}_0$ and $\mathfrak{m}_1$ and provides them to the experiment. The experiment flips a coin $b \leftarrow_\$ \{0, 1\}$ and encrypts $\mathfrak{m}_b$. This *challenge ciphertext* is now given to the adversary, who has to guess whether the challenge ciphertext contains $\mathfrak{m}_0$ or $\mathfrak{m}_1$. The adversary wins if it guesses correctly. We consider a public encryption scheme secure (under this notion), if the adversary's chance of winning is not noticeably better than blind guessing. The rationale behind this notion is that messages sent through a network may be influenced by

certain adversarial choice, so the adversary might have a-priori information about the plaintext. If a scheme however fulfills the semantic security notion, then such a-priori information is of no use to the adversary. Another way to view this is that encryptions of such a scheme hide all partial information about the plaintext message. Notice that no encryption scheme with deterministic encryption algorithm can fulfill this security notion, as in this case the adversary could just encrypt $\mathfrak{m}_0$ and $\mathfrak{m}_1$ by itself and check which one yields the challenge ciphertext. Thus, IND-CPA secure encryption schemes must necessarily be probabilistic, i.e. the encryption algorithm must produce a randomized output.

Definition 2.8. *We say a public key encryption-scheme* PKE *is ciphertext indistinguishable under chosen message attacks (IND-CPA), if every PPT-adversary* $\mathcal{A}$ *has success-probability at most negligibly better than* $1/2$ *in the experiment* $\mathit{IND\text{-}CPA}$*, i.e.* $\Pr[\mathit{IND\text{-}CPA}(\mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

Experiment *IND-CPA*
$(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{Dec}}(sk,\tau^*,\cdot,\cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}^* \leftarrow \mathsf{PKE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

While IND-CPA security guarantees security against passive or eavesdropping adversaries, this notion falls short against adversaries that actively manipulate messages sent through a network. In such a scenario, an adversary may intercept ciphertexts, manipulate them, and observe whether the receiver accepts this ciphertext or rejects. Using this feedback, the adversary might learn certain parts of the encrypted message, breaking the security of the scheme. Therefore, to deal with active adversaries, a stronger security notion is needed. Naor and Yung [NY90] defined chosen ciphertext attacks (CCA1) to model the capabilities of such stronger adversaries. The basic idea is to keep the basic IND-CPA experiment, but to provide the adversary with a decryption oracle, i.e. a black box that decrypts ciphertexts of the adversary's choice *before* he receives the challenge ciphertext. Thereafter, the experiment continues like the IND-CPA experiment. A common metaphor for this kind of attack are socalled *lunchtime attacks*. We imagine that in this kind of attack the adversary gains control of a decryption resource (e.g. a smartcard) while its legitimate owner is in lunch break. The adversary may now use the decryption resource to gain as much knowledge as possible about the secret key, but has to return the decryption resource and must guess the message encrypted in the challenge ciphertext without it.

Definition 2.9. *We say a public key encryption-scheme* PKE *is ciphertext indistinguishable under non-adaptively chosen ciphertext attacks (IND-CCA1), if every PPT-adversary* $\mathcal{A}$ *has success-probability at most negligibly better than* $1/2$ *in the experiment* $\mathit{IND\text{-}CCA1}$*, i.e.* $\Pr[\mathit{IND\text{-}CCA1}(\mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

While the notion of IND-CCA1 security comes close to what we expect from a public key encryption scheme in terms of security, this is still not sufficient for certain applications. Rackoff and Simon [RS91] proposed the notion of adaptive chosen

Experiment IND-CCA1
$(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk,\cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}^* \leftarrow \mathsf{PKE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(sk, \mathfrak{c})$
$\mathfrak{m} \leftarrow \mathsf{PKE.Dec}(sk, \tau, \mathfrak{c})$
Return $\mathfrak{m}$

ciphertext security. For this notion the adversary also gets access to a decryption oracle in its guessing phase, however with the restriction that he cannot use it to decrypt the challenge ciphertext, i.e. the decryption oracle refuses to decrypt the challenge ciphertext. This restriction is necessary to make this security notion not trivially unfulfillable. For several years this stronger security notion was mostly of theoretical interest, but had no practical impact. This certainly changed when Bleichenbacher discovered such an adaptive chosen ciphertext attack against the RSA-based encryption standard PKCS#1 [JK03, Ble98].

Definition 2.10. *We say a public key encryption-scheme* PKE *is ciphertext indistinguishable under adaptively chosen ciphertext attacks (IND-CCA2), if every PPT-adversary $\mathcal{A}$ has success-probability at most negligibly better than* $1/2$ *in the experiment* IND-CCA2*, i.e.* $\Pr[\mathit{IND\text{-}CCA2}(\mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

Experiment IND-CCA2
$(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk,\cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}^* \leftarrow \mathsf{PKE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec1}}(sk, \mathfrak{c})$
$\mathfrak{m} \leftarrow \mathsf{PKE.Dec}(sk, \tau, \mathfrak{c})$
Return $\mathfrak{m}$
$\mathcal{O}_{\mathsf{Dec2}}(sk, \mathfrak{c}^*, \mathfrak{c})$
If $\mathfrak{c} = \mathfrak{c}^*$
Return $\perp$
Otherwise
$\mathfrak{m} \leftarrow \mathsf{PKE.Dec}(sk, \tau, \mathfrak{c})$
Return $\mathfrak{m}$

2.4.4. One-Time Signature Schemes

Since one-time signatures play an important role in the construction of IND-CCA2 secure encryption schemes, we will give a brief overview over this cryptographic primitive. One-time signatures were first defined and constructed by Lamport [Lam79]. The key difference between standard signature schemes and one-time signature schemes is the following. While a standard signature scheme remains secure if a signing key is used to sign an arbitrary number of messages, one-time signature schemes may become insecure if more than one message is signed. We first provide the syntactical definition of a one-time signature scheme.

Definition 2.11. *A one-time signature scheme* OTS *consists of three algorithms* OTS.Gen*,* OTS.Sign *and* OTS.Verify*, such that*

- $\mathsf{OTS.Gen}(1^\lambda)$ *is a PPT-algorithm that takes a security-parameter* λ *and outputs a pair of verification and signature keys* (vk, sgk).

- $\mathsf{OTS.Sign}_{sgk}(\mathfrak{m})$ *is a PPT-algorithm that takes a signature key* sgk*, a message* $\mathfrak{m}$ *and outputs a signature* σ.

- $\mathsf{OTS.Verify}(vk, \mathfrak{m}, \sigma)$ *is a PPT-algorithm taking as input a verification key* vk*, a message* $\mathfrak{m}$ *and a signature* σ *and outputs a bit* $b \in \{0, 1\}$.

As with public key encryption, a standard requirement of one-time signatures is completeness. Completeness basically requires that messages that were legitimately signed by the owner of the secret key will verify, except with negligible probability.

Definition 2.12. *We say that* $\mathsf{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS}, \mathsf{Verify})$ *is complete, if it holds for all messages* $\mathfrak{m}$ *that*

$$\Pr[\mathsf{OTS.Verify}(vk, \mathfrak{m}, \mathsf{OTS.Sign}(sgk, \mathfrak{m}) = 1 : (vk, sgk) = \mathsf{OTS.Gen}(1^\lambda)] > 1 - \mathsf{negl}(\lambda).$$

We will now define security for one-time signature schemes. The security property we need for our constructions is called strong existential unforgeability under one-time chosen message attacks. As usual, we will define a security experiment. The experiment first generates a pair of verification and signing keys. The adversary $\mathcal{A}$ then receives the verification key and gets access to a one-time signing oracle that signs at most one message of his choice and then refuses to sign further messages. Call the message signature pair $\mathcal{A}$ obtains from the oracle $(\mathfrak{m}', \sigma')$. The adversary eventually outputs a pair of message and signature $(\mathfrak{m}^*, \sigma^*)$. The experiment now checks if the σ^* is a valid signature for $\mathfrak{m}^*$ *and* if $\mathfrak{m}^* \neq \mathfrak{m}'$ or $\sigma^* \neq \sigma'$. If this check is passed, the adversary wins the experiment, if not he loses. We consider a one-time signature scheme secure under this notion if every PPT-adversary $\mathcal{A}$ has most negligible chance of winning this experiment. Notice that it is a perfectly legitimate strategy for the adversary to output an $\mathfrak{m}^*$ with $\mathfrak{m}^* = \mathfrak{m}'$ but $\sigma^* \neq \sigma'$. Thus, the adversary can win the experiment by coming up with a new signature for a message for which he has already seen a signature. This is what distinguishes the *strong* unforgeability property from the standard unforgeability property. The standard unforgeability property for signature schemes only requires that the adversary fails to forge a signature for a new message, for which he has not yet seen a valid signature. In the context of constructing IND-CCA2 secure encryption schemes, the strong unforgeability property is very important.

Definition 2.13. *We say a one-time signature scheme* OTS *is strongly existentially unforgeable under chosen one-time chosen message attacks, if every PPT-adversary* $\mathcal{A}$ *has most negligibly success probability in the experiment* $\mathsf{sEUF\text{-}OTCMA}$*, i.e.* $\Pr[\mathsf{sEUF\text{-}OTCMA}(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda)$.

Standard EUF-OTCMA secure one-time signature schemes can be constructed from any one-way function [Lam79]. Using the same construction, sEUF-OTCMA secure one-time signature schemes can be constructed using universal one-way hash functions (UOWHF) [NY89, Gol04]. Since UOWHFs in turn can be constructed from any one-way function [Rom90], this yields a construction of sEUF-OTCMA secure signatures from any one-way function.

Experiment sEUF-OTCMA
$(vk, sgk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$(\mathfrak{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}(sgk,\cdot)}(\mathtt{find}, vk)$
Parse $\mathsf{r} = (\mathfrak{m}', \sigma')$
If $\mathsf{OTS.Verify}(vk, \mathfrak{m}^*, \sigma^*) = 1$
and $(\mathfrak{m}^*, \sigma^*) \neq (\mathfrak{m}', \sigma')$
Return 1
Otherwise
Return 0

Initialize $\mathsf{r} \leftarrow (\perp, \perp)$
$\mathcal{O}_{\mathsf{Sign}}(sgk, \mathfrak{m})$
If $\mathsf{r} = (\perp, \perp)$
$\sigma \leftarrow \mathsf{OTS.Sign}(sgk, \mathfrak{m})$
$r \leftarrow (\mathfrak{m}, \sigma)$
Return σ
Otherwise
Return $\perp$

2.5. Coding Theory

In this section, we provide the coding-theoretic tools and techniques required in this thesis. Error correcting codes are a central tool in theoretical computer science. The interesting property of error correcting codes is their geometry. All distinct codewords of a linear code are well-separated by a minimum distance. This *geometric* separation introduces the redundancy necessary to deal with errors inflicted by a noisy channel or an adversary. Thus, error correcting codes can be equivalently seen as encodings the *amplify* distance of messages.

2.5.1. Error Models

As their name suggests, the main purpose error-correcting codes is to encode messages in a way that is resilient against data loss incurred by corruption or loss of parts of the encoded message. Classically, two different models of how a channel introduces errors are distinguished. In the first model, due to Shannon [Sha48], a memoryless channel introduces errors at random. This model is a reasonable approximation for most physical channels, where the most prominent sources of errors are thermal noise, multi-path interference or glitches. In the second model, which is due to Hamming [Ham50], no such assumptions about a more or less benign or average case behavior of channels are made. In Hamming's model, errors are chosen adversarially, i.e. in a worst case manner. In this model channels can be considered as malicious entities that try to corrupt messages using a certain number of errors. The goal of the adversarial channel is to inflict a decoding error on the receivers side. Restricting the number of errors the channel is allowed to introduce is necessary in order to exclude trivial strategies of the channel, such as erasing the whole message. Notice that this perspective of a channel fits very well with the cryptographic view of adversarial behavior detailed in the last section. We will now discuss certain types of errors. The most prominent types of errors discussed in literature are symbol errors and erasures. Let $\mathbb{F}$ be a finite alphabet over which messages are defined. We will always assume that $\mathbb{F}$ is a finite field, i.e. there are well defined addition, multiplication and division operations in $\mathbb{F}$.

2.5.1.1. Bit and Symbol Errors

A symbol error is an arbitrary alteration of a symbol or component of a message $\mathbf{x} \in \mathbb{F}^n$ in an *unknown* location. We will usually treat symbol errors as additive offsets of a message, i.e. if $\mathbf{x} \in \mathbb{F}^n$ is the uncorrupted message then $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$ is a corrupted version of $\mathbf{x}$ which differs from $\mathbf{x}$ in the locations/components i for which

$\mathbf{e}_i \neq 0$. We call $\mathbf{e} \in \mathbb{F}^n$ the error vector. In case the code-alphabet is binary, we refer to symbol errors as bit errors.

2.5.1.2. Erasure Errors

Erasure errors model the *detectable* loss of information in a transmitted message. But in contrast to symbol errors, the locations of erasure errors are known. To indicate that an erasure has occurred, we replace the erased component by an erasure symbol $\perp$. Erasures can be viewed as a puncturing of a codeword, i.e. the effect of an erasure is the projection of the codeword onto a shorter code.

2.5.1.3. Simultaneous Block Erasure and Bit Errors

Of particular importance in Chapter 6 is an error model in which bit errors and erasures occur simultaneously. Assume that a message $\mathbf{x} \in \mathbb{F}^{n_1 \cdot n_2}$ is partitioned in n_1 blocks of size n_2, i.e. $\mathbf{x} = (\mathbf{x}_1^T \| \ldots \| \mathbf{x}_{n_1}^T)^T$ where all $\mathbf{x}_i$ are elements of $\mathbb{F}_2^{n_2}$. We allow bit errors to occur in each bit of each block $\mathbf{x}_i$, but erasure errors only to occur block-wise, i.e whole blocks $\mathbf{x}_i$ are erased instead of just single bits of those blocks. Looking ahead, this error model will occur in a security reduction in which symbol errors are inflicted by an adversary and erasure errors appear due to incomplete knowledge of a secret key (by the reduction).

2.5.2. Linear Error Correcting Codes

The most important class of error correcting codes studied in literature are linear error correcting codes. Linearity allows efficient representation of codes and facilitates the analysis of their properties. Basically, linear codes are subspaces of finite vector spaces.

Definition 2.14 (Linear Codes)**.** *Let $\mathbb{F}_q$ be the finite field of size q. A q-ary linear code C of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_q^n$. We also say C is a q-ary $[n, k]$ code.*

The interesting aspect of linear codes is their *geometry*. This geometry is defined in terms of the Hamming metric. The Hamming metric measures the how different two given vectors are.

Definition 2.15 (Hamming Metric)**.** *Let $\mathbb{F}$ be a finite field. The Hamming metric or Hamming weight of a vector $\mathbf{x} \in \mathbb{F}^n$ is defined by*

$$\mathsf{wgt}(\mathbf{x}) = |\{i \mid x_i \neq 0\}|,$$

i.e. $\mathsf{wgt}(\mathbf{x})$ counts the number of nonzero components of $\mathbf{x}$. The Hamming distance or Hamming metric $\mathsf{d}(\mathbf{x}, \mathbf{y})$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$ is defined by $\mathsf{d}(\mathbf{x}, \mathbf{y}) = \mathsf{wgt}(\mathbf{x} - \mathbf{y})$.

The following Lemma shows that the Hamming weight fulfills the triangle inequality and thus establishes that the Hamming metric actually is a metric.

Lemma 2.7. *Let $\mathbb{F}$ be a finite field. It holds for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$ that $\mathsf{wgt}(\mathbf{x} + \mathbf{y}) \leq \mathsf{wgt}(\mathbf{x}) + \mathsf{wgt}(\mathbf{y})$. Furthermore $\mathsf{wgt}(\mathbf{x}) = 0$ if and only if and only if $\mathbf{x} = \mathbf{0}$.*

Proof. It holds that

$$\begin{aligned}\mathsf{wgt}(\mathbf{x}+\mathbf{y}) &= |\{i \mid x_i + y_i \neq 0\}| \\ &\leq |\{i \mid x_i \neq 0 \text{ or } y_i \neq 0\}| \\ &\leq |\{i \mid x_i \neq 0\}| + |\{i \mid y_i \neq 0\}| \\ &= \mathsf{wgt}(\mathbf{x}) + \mathsf{wgt}(\mathbf{y}).\end{aligned}$$

Moreover, if $\mathbf{x} \neq \mathbf{0}$ then there exists an index i such that $x_i \neq 0$ and therefore $\mathsf{wgt}(\mathbf{x}) \neq 0$. □

The minimum distance of a linear code C provides information how different distinct codewords of C are.

Definition 2.16. *The minimum distance* $\mathsf{d}(\mathsf{C})$ *of a linear code* C *is defined by*

$$\mathsf{d}(\mathsf{C}) = \min_{\mathbf{x} \in \mathsf{C} \setminus \{\mathbf{0}\}} \mathsf{wgt}(\mathbf{x})$$

$\mathsf{d}(\mathsf{C})$ is also the minimum distance between to distinct codewords $\mathbf{x}, \mathbf{y} \in \mathsf{C}$, as

$$\mathsf{d}(\mathbf{x}, \mathbf{y}) = \mathsf{wgt}(\underbrace{\mathbf{x} - \mathbf{y}}_{\in \mathsf{C} \setminus \{\mathbf{0}\}}) \geq \mathsf{d}(\mathsf{C}).$$

If C is a q-ary linear $[n, k]$ code and $\mathsf{d}(\mathsf{C}) = d$, then we say C is a q-ary $[n, k, d]$ code. We will call $R = \frac{k}{n}$ the rate of C and $\delta = \frac{d}{n}$ the relative minimum distance of C.

Definition 2.17 (Hamming Sphere and Hamming Ball)**.** *Let* $r \leq n$*. We call*

$$\mathsf{S}_n(r, q) = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathsf{wgt}(\mathbf{x}) = r\}$$

the q*-ary Hamming sphere of radius* r*. Moreover, we call*

$$\mathsf{B}_n(r, q) = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathsf{wgt}(\mathbf{x}) \leq r\}$$

the q*-ary Hamming ball of radius* r*. For* $q = 2$ *we will set* $\mathsf{S}_n(r) = \mathsf{S}_n(r, 2)$ *and* $\mathsf{B}_n(r) = \mathsf{B}_n(r, 2)$*.*

Lemma 2.8. *It holds that*

$$|\mathsf{B}_n(r, q)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i \leq |\mathsf{S}_n(r, q)| = \binom{n}{r} (q-1)^r \leq q^{H_q(r/n) \cdot n}.$$

Proof. There are $\binom{n}{r}$ possibilities of choosing the non-zero locations for an $\mathbf{x} \in \mathbb{F}_q^n$ with $\mathsf{wgt}(\mathbf{x}) = r$. Each of these locations can be set with one of the $q - 1$ non-zero elements of $\mathbb{F}_q$. Thus it holds

$$|\mathsf{S}_n(r, q)| = \binom{n}{r} (q-1)^r.$$

By Lemma 2.5 it holds that $\binom{n}{r} \leq 2^{H(r/n) \cdot n}$ and thus

$$|\mathsf{S}_n(r, q)| = \binom{n}{r} (q-1)^r \leq 2^{H(r/n) \cdot n} \cdot 2^{r \cdot \log(q-1)} = q^{H_q(r/n) \cdot n}.$$

Clearly, it holds that

$$\mathsf{B}_n(r,q) = \bigcup_{i=0}^{r} \mathsf{S}_n(i,q)$$

and therefore

$$|\mathsf{B}_n(r,q)| = \sum_{i=0}^{r} |\mathsf{S}_n(i,q)| \leq |\mathsf{S}_n(r,q)| \leq q^{H_q(r/n)\cdot n}.$$

□

As linear codes are vectors spaces, they can either be expressed in terms of a basis matrix or by a homogeneous linear equation system.

Definition 2.18 (Generator and Parity Check Matrices, Error Syndromes)**.** *Let* C *be a* q*-ary linear* $[n,k]$ *code. If* $\mathbf{G} \in \mathbb{F}_q^{n\times k}$ *is a basis matrix of* C *as a vector space, we also call* $\mathbf{G}$ *a generator matrix of* C*, i.e. it holds that*

$$\mathsf{C} = \mathsf{C}(\mathbf{G}) = \{\mathbf{G}\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x} \in \mathbb{F}_q^k\}.$$

If C *is the kernel of a matrix* $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times k}$*, we call* $\mathbf{H}$ *a parity check matrix of* C*, i.e.*

$$\mathsf{C} = \mathsf{C}^{\perp}(\mathbf{H}) = \mathsf{Ker}(\mathbf{H}) = \{\mathbf{y} \in \mathbb{F}_q^n \mid \mathbf{H}\mathbf{y} = \mathbf{0}\}.$$

For an $\tilde{\mathbf{x}} \in \mathbb{F}_q^n$ *we call* $\mathbf{s} = \mathbf{H}\cdot\tilde{\mathbf{x}}$ *the error syndrome or just syndrome of* $\tilde{\mathbf{x}}$*.*

Clearly, if $\mathbf{G}$ is a generator matrix of C and $\mathbf{H}$ is a parity check matrix of C, then $\mathbf{H}\cdot\mathbf{G} = \mathbf{0}$.

Definition 2.19 (Dual Codes)**.** *Let* C *be a* q*-ary linear* $[n,k]$ *code. We call*

$$\mathsf{C}^{\perp} = \{\mathbf{y} \in \mathbb{F}_q^n \mid \forall \mathbf{x} \in \mathsf{C}:\ \langle \mathbf{y}, \mathbf{x}\rangle = 0\}$$

the dual code of C*.*

Clearly, if $\mathbf{G}$ is a generator matrix of C and $\mathbf{H}$ is a parity check matrix of C, then $\mathbf{H}^T$ is a generator matrix of $\mathsf{C}^{\perp}$ and $\mathbf{G}^T$ is a parity check matrix of $\mathsf{C}^{\perp}$, i.e. it holds that

$$\mathsf{C}^{\perp} = \mathsf{C}(\mathbf{H}^T) = \mathsf{C}^{\perp}(\mathbf{G}^T).$$

Two codes are called equivalent if they are *structurally* the same. This is formalized as follows.

Definition 2.20. *We say that two* $[n,k]$ *codes* C_1 *and* C_2 *are equivalent, if there exists a permutation matrix* $\mathbf{P} \in \mathbb{F}_q^{n\times n}$ *such that* $\mathsf{C}_2 = \mathbf{P}\mathsf{C}_1$*, i.e. for all* $\mathbf{x} \in \mathsf{c}_1$

$$\mathbf{P}\mathbf{x} \in \mathsf{C}_2.$$

If $\mathbf{G}_1$ is a generator matrix of a $[n,k]$ code C_1, $\mathbf{G}_2$ is a generator matrix of a $[n,k]$ code C_2 and C_1 and C_2 are equivalent, then there exists a permutation matrix $\mathbf{P}$ and an invertible matrix $\mathbf{T}$ such that

$$\mathbf{G}_2 = \mathbf{P}\mathbf{G}_1\mathbf{T}.$$

There are no efficient algorithms known that decide whether the codes generated by two matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ are equivalent. The McEliece problem (c.f. Section 5.2) is (in part) based on the intractability of this problem.

We will usually assume that a linear code C is equipped with a standard generator matrix $\mathbf{G}$. We will use the syntax

$$\mathsf{C.Encode}(\mathbf{x}) = \mathbf{G}\mathbf{x}$$

to denote that $\mathbf{x}$ is encoded using the generator matrix $\mathbf{G}$. Moreover, if C has an efficient decoding algorithm for the generator matrix $\mathbf{G}$, we will denote this algorithm by $\mathsf{C.Decode}(\cdot)$. In case there exists an efficient list decoding algorithm for the basis $\mathbf{G}$, we will denote this algorithm by $\mathsf{C.ListDecode}(\cdot)$.

2.5.3. Random Codes and Best Known Codes

An important goal of coding theory is finding codes with optimal parameters, i.e. with maximum possible rate and relative minimum distance. The Singleton bound gives an upper bound for the dimension of a code with a desired minimum distance.

Lemma 2.9 (Singleton Bound [Sin64])**.** *Let C be a q-ary $[n, k, d]$ code. Then it holds that*

$$k \leq n - d + 1.$$

Proof. Let $\mathsf{C}' \subseteq \mathbb{F}_q^{(n-1)\times k}$ be the code obtained by puncturing C, e.g. by dropping the last symbol of all codewords in C. C' has minimum distance at least $d-1$, as if $\mathbf{x} \in \mathsf{C}$ is a vector of minimum weight d in C and $\mathbf{x}' \in \mathsf{C}'$ is obtained by dropping the last symbol of $\mathbf{x}$, then $\mathsf{wgt}(\mathbf{x}') \geq d-1$. Thus, the number of codewords of C' is the same as of C as long as $d > 1$. Therefore, we can repeat this procedure at least $d-1$ times without decreasing the number of codewords. It follows that

$$k \leq n - d - 1.$$

□

Randomly chosen codes are among the best known codes. We will first provide a version of Shannon's noisy channel coding theorem for binary symmetric channels. A binary symmetric channel inflicts errors chosen from a Bernoulli distribution $\mathsf{Ber}(\rho)$.

Theorem 2.3 (Noisy Channel Coding Theorem for Binary Symmetric Channels [Sha48])**.** *Let $\rho \in [0, 1/2)$ and let $k \leq (1 - H(\rho) - \epsilon)n$ for any constant $\epsilon > 0$. Let $\mathbf{G} \leftarrow_\$ \mathbb{F}_2^{n\times}$ be chosen uniformly at random, let $\mathbf{x} \leftarrow_\$ \mathbb{F}_2^n$ be chosen uniformly at random and let $\mathbf{e} \leftarrow_\$ \mathsf{Ber}(n, \rho)$. Let $\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{e}$. Then $\mathbf{x}$ can be uniquely recovered from $\mathbf{G}$ and $\mathbf{y}$, except with negligible probability over the choice of $\mathbf{G}$ and $\mathbf{e}$.*

Proof. Since $\mathbf{e}$ is chosen by the Bernoulli distribution $\mathsf{Ber}(n, \rho)$ it holds by the Chernoff inequality (Theorem 2.1) that

$$\mathsf{wgt}(\mathbf{e}) \leq (1 + \epsilon/2)\rho n,$$

except with probability $e^{-\frac{\epsilon^2}{12}\rho n}$. Fix $\mathbf{x}$ and an $\mathbf{e}$ with $\mathsf{wgt}(\mathbf{e}) \leq (1 + \epsilon/2)\rho n$. Clearly, it holds that

$$\mathsf{wgt}(\mathbf{y} - \mathbf{G}\mathbf{x}) = \mathsf{wgt}(\mathbf{e}) \leq (1 + \epsilon/2)\rho n.$$

We will consider the probability that there exists an $\mathbf{x}' \neq \mathbf{x}$ such that

$$\mathsf{wgt}(\mathbf{y} - \mathbf{G}\mathbf{x}') \leq (1 + \epsilon/2)\rho n$$

holds. Fix any $\mathbf{x}' \neq \mathbf{x}$. It holds that

$$\mathbf{y} - \mathbf{G}\mathbf{x}' = \mathbf{G}(\mathbf{x} - \mathbf{x}') + \mathbf{e}.$$

Thus if

$$\mathsf{wgt}(\mathbf{y} - \mathbf{G}\mathbf{x}') \leq (1 + \epsilon/2)\rho n$$

then

$$\mathbf{G}(\mathbf{x} - \mathbf{x}') + \mathbf{e} \in \mathsf{B}_n((1 + \epsilon/2)\rho n).$$

As $\mathbf{x} - \mathbf{x}' \neq \mathbf{0}$ and $\mathbf{G}$ is chosen uniformly at random, it holds that $\mathbf{G}(\mathbf{x} - \mathbf{x}')$ is distributed uniformly at random. Thus it holds that

$$\begin{aligned}
\Pr[\mathsf{wgt}(\mathbf{y} - \mathbf{G}\mathbf{x}') \leq (1 + \epsilon/2)\rho n] &= \Pr[\mathbf{G}(\mathbf{x} - \mathbf{x}') + \mathbf{e} \in \mathsf{B}_n((1 + \epsilon/2)\rho n)] \\
&\leq \frac{|\mathsf{B}_n((1 + \epsilon/2)\rho n)|}{2^n} \\
&\leq 2^{-n+H((1+\epsilon/2)\rho))n} \\
&\leq 2^{-n+(1+\epsilon/2)H(\rho)n}
\end{aligned}$$

where the last inequality follows by Lemma 2.4. By a union bound, it holds that

$$\begin{aligned}
\Pr[\exists \mathbf{x}' \neq \mathbf{x} : \mathsf{wgt}(\mathbf{y} - \mathbf{G}\mathbf{x}') \leq (1 + \epsilon/2)\rho n] &\leq 2^k \cdot 2^{-n+(1+\epsilon/2)H(\rho)n} \\
&= 2^{k-n+(1+\epsilon/2)H(\rho)n} \\
&\leq 2^{(-1+H(\rho)/2)\epsilon n} \\
&\leq 2^{-\epsilon n/2}
\end{aligned}$$

where we have used that $k \leq (1 - H(\rho) - \epsilon)n$ and $H(\rho) \leq 1$ for $x \in (0, 1/2)$. All together, this yields that $\mathbf{x}$ can be recovered from $\mathbf{G}$ and $\mathbf{y}$ with probability at least

$$1 - e^{\epsilon^2 \rho n/12} - e^{-\epsilon n/2},$$

which is overwhelming in n. □

The noisy channel coding theorem assumes that the error $\mathbf{e}$ is chosen at random. If the error is chosen adversarially, then the following bounds provide the best known codes.

Theorem 2.4 (Gilbert [Gil52], Varshamov [Var57] (GV))**.** *Let $\epsilon > 0$ be an arbitrary constant, and q be a prime power. Let n, k and δ be such that*

$$k \leq (1 - H_q(\delta) - \epsilon)n.$$

Let $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ be chosen uniformly at random. Then $\mathsf{C} = \mathsf{C}(\mathbf{G})$ has minimum distance at least δn, except with negligible probability in n.

Proof. We will bound the probability that there is a non-zero vector in C with weight smaller than δn. Fix an $\mathbf{x} \neq \mathbf{0}$. Then $\mathbf{Gx}$ is distributed uniformly at random in $\mathbb{F}_q^n$. Thus it holds that

$$\Pr[\mathsf{wgt}(\mathbf{Gx}) < \delta n] = \Pr[\mathbf{Gx} \in \mathsf{B}_n(\delta n)] \leq \frac{|\mathsf{B}_n(\delta n)|}{q^n} = q^{-n+H_q(\delta)n}.$$

Thus, the probability that there exists an $\mathbf{x} \in \mathbb{F}_q^k \backslash \{\mathbf{0}\}$ such that $\mathsf{wgt}(\mathbf{Gx}) < \delta n$ can be bounded by

$$\Pr[\exists \mathbf{x} : \mathsf{wgt}(\mathbf{Gx}) < \delta n] \leq q^k q^{-n+H_q(\delta)n} \leq q^{-\epsilon n}.$$

Thus, all non-zero vectors in $\mathsf{C}(\mathbf{G})$ have Hamming weight at least δn, except with negligible probability. □

If $q \geq 49$, then there exists a construction of *non-random codes* that beats the Gilbert Varshamov bound.

Theorem 2.5 (Tsfasman Vlăduţ Zink [TVZ82] (TVZ))**.** *Let q be a square. For every rate R there exists an asymptotically good sequence of codes with rate R and minimum distance δ given that*

$$R + \delta \geq 1 - \frac{1}{\sqrt{q} - 1}.$$

The proof of Theorem 2.5 is rather involved and requires algebraic geometry. See for instance [HB98].

2.5.4. Efficient Decoding

We will now discuss several constructions of efficiently decodable codes. There are two decoding goals for linear codes discussed in literature: minimum distance decoding and list decoding. The goal of minimum distance decoding is, given a message $\mathbf{x}$, to find the codeword closest to $\mathbf{x}$ in terms of the Hamming distance, i.e. to find a $\mathbf{c} \in \mathsf{C}$ such that

$$\mathsf{wgt}(\mathbf{c} - \mathsf{wgt}(\mathbf{x})) = \min_{\mathbf{y} \in \mathsf{C}}(\mathsf{wgt}(\mathbf{y} - \mathbf{x})).$$

Using minimum distance decoding, one can correct at most $(d-1)/2$ errors or $d-1$ erasures. If more errors are present, then a decoding error may occur. The notion of minimum distance decoding was relaxed to the notion of list decoding (implicitly defined in [GL89]). The goal of list decoding is not to compute the closest codeword for a given $\mathbf{x}$, but a *short* list of *nearby* codewords. In a situation where more than $(d-1)/2$ errors are present, one may still be able to recover the right codeword $\mathbf{c}$ uniquely given some additional information about $\mathbf{c}$.

Definition 2.21. *Let C be an $[n, k, d]_q$ code.*

- *We say that an efficient algorithm* $\mathsf{C.Decode}$ *decodes t errors and l erasures, if given an $\mathbf{x} \in \mathbb{F}_q^n$ with distance at most t from C of which at most l components are erased, then* $\mathsf{C.Decode}(\mathbf{x})$ *outputs $\mathbf{c}$.*

- *We say that an efficient algorithm* $\mathsf{C.ListDecode}$ *list-decodes t errors and l erasures, if given an $\mathbf{x} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} \in \mathsf{C}$ is a codeword and $\mathbf{e}$ is an error of weight at most t, and $\mathbf{x}$ contains at most l erasures, then* $\mathsf{C.ListDecode}(\mathbf{x})$ *outputs a list L of codewords that has at most polynomial size in n and contains $\mathbf{c}$.*

2.5.5. Reed Solomon Codes

Reed Solomon codes [RS60] are the most well known class of so-called evaluation codes. Information words are interpreted as low degree polynomials $f(X)$ and codewords *function tables* of $f(X)$.

Definition 2.22 (Reed Solomon [RS60])**.** *Let $n \geq k$ and q be a prime power with $q \geq n$. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$. The q-ary $[n, k]$ Reed Solomon code $\mathsf{RS}_{\boldsymbol{\alpha}}$ is consists of all codewords $\mathbf{c} = (c_1, \ldots, c_n)$ for which there exists an $f \in \mathbb{F}_q[X]$ of degree $k-1$ such that*

$$\begin{aligned} c_1 &= f(\alpha_1) \\ &\vdots \\ c_n &= f(\alpha_n). \end{aligned}$$

We will usually omit explicitly stating $\boldsymbol{\alpha}$.

Theorem 2.6. *Let RS be a q-ary $[n, k]$ Reed Solomon code. Then RS has dimension k and minimum distance $d = n-k+1$. Furthermore, there exists an efficient decoder $\mathsf{RS.Decode}$ for RS that can decode t errors and l erasures given that*

$$2t + l \leq d - 1 = n - k.$$

A proof of Theorem 2.6 can be found in any coding theory textbook, e.g. [HB98, vL99]. The following theorem establishes that Reed Solomon codes are list-decodable.

Theorem 2.7 (Guruswami Sudan [GS99])**.** *Let RS be a q-ary $[n, k]$ Reed Solomon code. There exists an efficient list decoder $\mathsf{RS.ListDecode}$ for RS that list-decodes t errors and l erasures given that*

$$t + l < n - \sqrt{(k+1)(n-l)}.$$

2.5.6. Binary Goppa Codes

Goppa codes are a class of subfield codes that are defined by a parity check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ over an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. Goppa codes [Ber73] can be defined as subfield codes of BCH codes [Hoc59, BRC60].

Definition 2.23. *Let $\mathbb{F}_{2^m}$ be the degree m extension field of $\mathbb{F}_2$. Let $g(X) \in \mathbb{F}_{2^m}[X]$ be a polynomial of degree t and let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in (\mathbb{F}_{2^m})^n$, where the α_i pairwise distinct non-zeros of $g(X)$, i.e. for all i $g(\alpha_i) \neq 0$. The binary Goppa-code $\Gamma(\boldsymbol{\alpha}, g)$ is the set of all vectors $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_2^n$ for which*

$$\sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} = 0 \pmod{g(X)}$$

holds.

Theorem 2.8. *Assume that $g(X) \in \mathbb{F}_2^m[X]$ has no multiple zeros. Then the binary Goppa code $\Gamma(\boldsymbol{\alpha}, g)$ has dimension $n - m \cdot t$ and minimum distance at least $2t + 1$. Moreover, $\Gamma(\boldsymbol{\alpha}, g)$ has an efficient decoder $\Gamma(\boldsymbol{\alpha}, g).\mathsf{Decode}(\cdot)$ that efficiently decodes up to t errors.*

For a proof of Theorem 2.8 refer to [Ber73]. Since the $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{2^m}$ need to be pairwise distinct non-zeros of $g(X)$ (which has degree t), we need to choose $2^m > n + t$.

2.5.7. Concatenated Codes

Concatenated codes are among the most powerful constructions of asymptotically good codes. Concatenated codes were introduced by Forney [For66] and Justesen [Jus72]. Binary concatenated codes encode messages in two steps. First a message is split into large blocks of size $\log(q)$ and encoded using an outer q-ary code. In the second step each component or symbol of this codeword encoded using an inner binary code.

Definition 2.24. *Let $n = n_1 n_2$, $k = k_1 k_2$ and $q = 2^{k_2}$. Let C_{out} be a q-ary $[n_1, k_1]$ code and let C_{in} be a binary $[n_2, k_2]$ code. Then the concatenated code $\mathsf{C} = \mathsf{C}_{in} \circ \mathsf{C}_{out}$ is defined by the following encoding procedure* C.Encode.

C.Encode($\mathbf{x}$)
  Interpret $\mathbf{x}$ *as an element of* $\mathbb{F}_q^{k_1}$
  $\mathbf{c} \leftarrow \mathsf{C}_{out}.\mathsf{Encode}(\mathbf{x})$
  Parse $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_{n_1}) \in \mathbb{F}_q^{n_1}$
  For $i = 1, \ldots, n_1$
    Interpret $\mathbf{c}_i$ *as an element of* $\mathbb{F}_2^{k_2}$
    $\mathbf{c}'_i \leftarrow \mathsf{C}_{in}.\mathsf{Encode}(\mathbf{c}_i)$
  $\mathbf{c}' \leftarrow ({\mathbf{c}'_1}^T \| \ldots \| {\mathbf{c}'_{n_1}}^T)^T$
  Return $\mathbf{c}'$

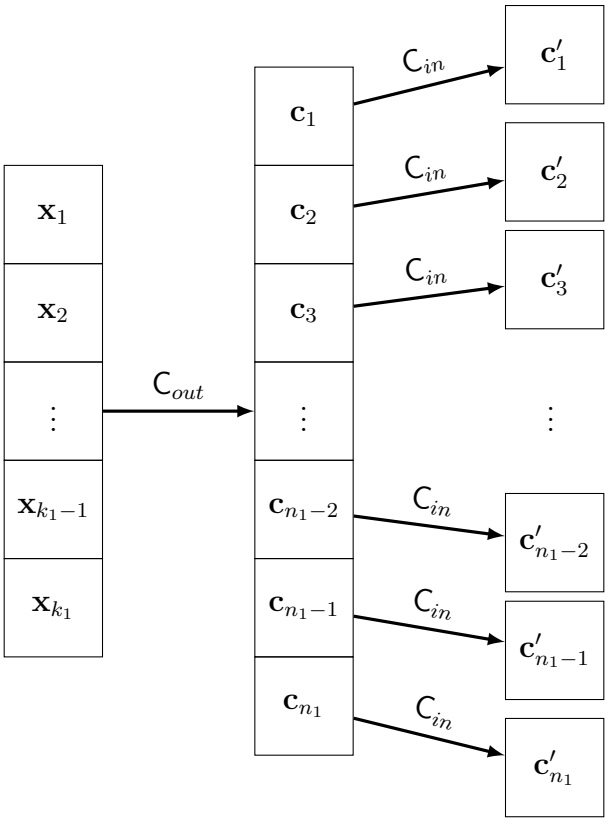

Figure 2.1.: Structure of concatenated codes

Concatenated codes are usually decoded using generalized minimum distance decoding (GMD) [For66]. GMD is a soft-decoding procedure which assigns decoding outputs of the inner decoder certain weights which represent the confidence in the

correctness of the output. These weights are then used by the decoder of the outer code to decide whether an erasure should be declared at the corresponding symbol.

Theorem 2.9 (Forney [For66]). *Let $q = 2^{k_2}$. Let C_{out} be a binary $[n_2, k_2]$ code with minimum distance at least d_2 and assume that there exists an efficient decoder $\mathsf{C}_{out}.\mathsf{Decode}$ that uniquely decodes up to $t_2 = d_2/2$ errors. Let C_{in} be a q-ary $[n_1, k_1]$ code that uniquely decodes t_1 errors and l_1 erasures given that $2t_1 + l_1 < d_1$. Then there exists an efficient decoder for $\mathsf{C}_{out} \circ \mathsf{C}_{in}$ that efficiently decodes up to $t_2 d_1$ errors.*

A modern proof of Theorem 2.9 can be found in [Gur01]. A particularly popular construction of concatenated codes uses as outer code a Reed Solomon code and as inner code an arbitrary good code. Given that the length of the inner code is short enough, it can be efficiently decoded by brute force. Thus, one usually aims to minimize the dimension of the inner code. Since we need $q \geq n_1$, we can achieve this by setting $k_2 = \lceil \log(n_1) \rceil$.

Concatenated codes carry a natural block structure, i.e. we can consider the codewords of the inner code as blocks of size n_2 of a codeword. It is therefore reasonable to consider erasures not just on the bit level, but also on the block level. In the rest of this Section, when we refer to block erasures we mean that entire codewords of the inner code are erased. Instantiating Theorem 2.9 with an outer Reed Solomon code yields the following corollary.

Corollary 2.10. *Let RS be an outer Reed Solomon code of relative distance δ_{RS} and let C_{in} be a short binary inner code with relative minimum distance δ_{in}. Then there exists an efficient decoder for $\mathsf{C} = \mathsf{C}_{in} \circ \mathsf{RS}$ that can efficiently correct an η fraction of bit errors and a σ fraction of block erasures given that*

$$\eta \leq \frac{1}{2}\delta_{in}(\delta_{\mathsf{RS}} - \sigma).$$

This bound can be improved asymptotically by turning to list decoding.

Theorem 2.10 (Guruswami [Gur01]). *Let RS be an outer Reed Solomon code of relative distance δ_{RS} and C_{in} be a binary inner code with relative minimum distance δ_{in}. Then there exists an efficient list decoder for $\mathsf{C}_{in} \circ \mathsf{RS}$ that list decodes an η fraction of bit errors and a σ fraction of block erasures given that*

$$\eta \leq \frac{1}{2}(1-\sigma) \cdot \left(1 - \sqrt{1 - 2\delta_{in}} - 2\sqrt{\frac{\delta_{in}(1-\delta_{\mathsf{RS}})}{1-\sigma}}\right).$$

If we consider bit erasures instead of block erasure, then we can use the following theorem.

Theorem 2.11 (Guruswami Sudan [GS99]). *Let $\epsilon > 0$ be an arbitrary constant. Let RS be an outer Reed Solomon code of relative distance δ_{RS} and C_{in} be a binary inner code with relative minimum distance δ_{in}. Then there exists an efficient list decoder for $\mathsf{C}_{in} \circ \mathsf{RS}$ that list decodes an η fraction of bit errors and a σ fraction of bit erasures given that*

$$\eta \leq \frac{1}{2} \cdot \left(1 - \sigma - \sqrt{(1+\epsilon)(1-2\delta_{in})} - \sqrt{\frac{(1-\sigma)(1-\delta_{\mathsf{RS}})}{\epsilon \cdot (1-2\delta_{in})}}\right).$$

We will briefly discuss concatenated codes with outer Reed Solomon code in more detail. For any given rate $R \in (0,1)$, we want to find such a concatenated code with maximum possible minimum distance. Let therefore RS be an outer Reed Solomon code with rate R_{out} and let C_{in} be an inner code with rate R_{in} and relative minimum distance δ_{in} that meets the Gilbert Varshamov bound, i.e. it holds that $R_{in} = 1 - H(\delta_{in})$. For any given rate R_{in} such a code C_{in} can be found efficiently by brute force search given that the length n_2 of C_{in} is at most logarithmic. We now want to determine the optimal choices for R_{out} and δ_{in} that lead to an optimal relative minimum distance δ for the concatenated code $\mathsf{RS} \circ \mathsf{C}_{in}$. The concatenated code $\mathsf{RS} \circ \mathsf{C}_{in}$ has rate

$$R = R_{out} \cdot R_{in} = R_{out} \cdot (1 - H(\delta_{in})) \tag{2.1}$$

and relative minimum distance at least

$$\delta = \delta_{in} \cdot (1 - R_{out}) = \delta_{in} \cdot (1 - R_{out}). \tag{2.2}$$

Solving equation 2.1 for R_{out} yields

$$R_{out} = \frac{R}{1 - H(\delta_{in})} \tag{2.3}$$

and the constraint $1 - H(\delta_{in}) > R$ as $R_{out} < 1$. This constraint can equivalently be expressed as $\delta_{in} < H^{-1}(1 - R)$. Substituting 2.3 into 2.2 yields

$$\delta = \delta_{in} \cdot \left(1 - \frac{R}{1 - H(\delta_{in})}\right).$$

The only undetermined variable on which δ depends on is δ_{in}. Maximizing δ as a function of δ_{in} under the constraint $\delta_{in} < H^{-1}(1 - R)$ cannot be solved analytically. Figure 2.5.7 shows a plot of a numerical solution of this optimization problem. The maximum possible δ is expressed as a function of R. In Section 6.4 we will formulate similar optimization problems to find optimal codes for the instantiation of our public key cryptosystems.

2.5.8. Further Constructions of Efficiently Decodable Codes

We will finally mention other constructions of efficiently decodable asymptotically good codes. Gallager [Gal63] introduced the notion of low density parity check (LDPC) codes. LDPC codes are defined by a parity check matrix which is the adjacency matrix of a sparse bipartite graph. Gallager showed that LDPC codes meet the Gilbert Varshamov bound. Later, Sipser and Spielman [SS94] showed that LDPC codes can be efficiently decoded up to a constant fraction of errors given that the bipartite graph used for the construction is a certain expander graph. More specifically, the decoding requires only linear time. Sipser and Spielman called such codes expander codes [SS94]. A large corpus of work has intensified the study in expander codes [Spi95, LMS$^+$97, LMSS98, RU01].

2.5.9. Additional Definitions

It will be convenient to bound the factor by which the multiplication with a binary matrix expands the Hamming weight of a binary vector. We will therefore define a matrix norm induced by the Hamming weight.

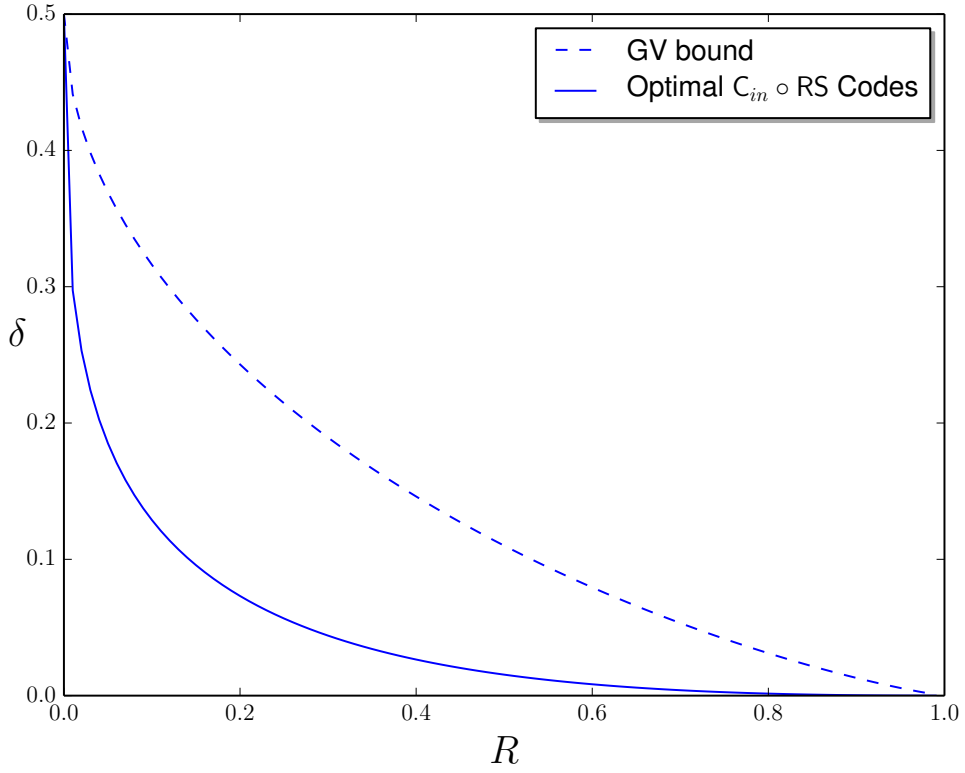

Figure 2.2.: Rate/distance trade-off for Best possible concatenated codes with outer Reed Solomon code

Definition 2.25. *The induced Hamming-norm* $\|\mathbf{A}\|_{\mathsf{wgt}}$ *of a matrix* $\mathbf{A} \in \mathbb{F}_2^{n \times k}$ *is defined by*

$$\|\mathbf{A}\|_{\mathsf{wgt}} = \max_{\mathbf{x} \neq 0} \frac{\mathsf{wgt}(\mathbf{A} \cdot \mathbf{x})}{\mathsf{wgt}(\mathbf{x})}$$

We will now show that matrices chosen from low noise Bernoulli distributions have a small induced Hamming-norm.

Lemma 2.11. *Let* m, n *be integers,* $\rho \in (0, 1)$ *and* $\beta > 0$ *be a constant. Let* $\mathbf{W}$ *be distributed according to* $\mathsf{Ber}(m \times n, \rho)$*. Then* $\|\mathbf{W}\|_{\mathsf{wgt}} \leq (1 + \beta)\rho m$*, except with probability* $n \cdot e^{-\frac{\beta^2}{3}\rho m}$*.*

Proof. Let $\mathbf{W} = (\mathbf{w}_1 \| \ldots \| \mathbf{w}_n)$ with $\mathbf{w}_i \in \mathbb{F}_2^m$. First observe that if the weight of all columns $\mathbf{w}_i$ of $\mathbf{W}$ is bounded by $(1 + \beta)\rho m$, then $\|\mathbf{W}\|_{\mathsf{wgt}} \leq (1 + \beta)\rho m$, as for all $\mathbf{x} \in \mathbb{F}_2^n$

$$\mathsf{wgt}(\mathbf{W}\mathbf{x}) = \mathsf{wgt}\Big(\sum_{i \in \mathsf{Support}(\mathbf{x})} \mathbf{w_i}\Big) \leq \sum_{i \in \mathsf{Support}(\mathbf{x})} \mathsf{wgt}(\mathbf{w}_i) \leq (1 + \beta)\rho m \cdot \mathsf{wgt}(\mathbf{x}).$$

Thus

$$\|\mathbf{W}\|_{\mathsf{wgt}} = \max_{\mathbf{x} \neq 0} \frac{\mathsf{wgt}(\mathbf{W} \cdot \mathbf{x})}{\mathsf{wgt}(\mathbf{x})} \leq (1 + \beta)\rho m.$$

By the Chernoff-bound, it holds for each $i \in \{1, \ldots, n\}$ that

$$\Pr[\mathsf{wgt}(\mathbf{w}_i) > (1+\beta)\rho m] \leq e^{-\frac{\beta^2}{3}\rho m},$$

thus a union-bound yields

$$\Pr[\exists i \in \{1, \ldots, n\} : \mathsf{wgt}(\mathbf{w}_i) > t] \leq n \cdot e^{-\frac{\beta^2}{3}\rho m}.$$

□

2.6. Lattices

A lattice Λ is a discrete subgroubs of $(\mathbb{R}^n, +)$. In this context, discrete means that all elements of Λ are well-separated, i.e. there exists a constant λ_1 such that it holds for all distinct $\mathbf{x}, \mathbf{x}' \in \Lambda$ that $\|\mathbf{x} - \mathbf{x}'\|_2 \geq \lambda_1$, where $\|\cdot\|_2$ is the euclidean norm defined over $\mathbb{R}^n$. Consequently, the definition of lattices is similar in spirit to the definition of linear codes. While all distinct codewords of linear $[n, k, d]$ code C are separated at least by Hamming distance d, all lattice points of a lattice Λ are separated by euclidean distance λ_1. Therefore, the defining property of both linear codes and lattices is their geometry.

Definition 2.26. *Let $n \geq k$. Let $\mathbf{B} \in \mathbb{R}^{n \times k}$ be a full rank matrix. We call*

$$\Lambda = \Lambda(\mathbf{B}) = \{\mathbf{Bz} \mid \mathbf{z} \in \mathbb{Z}^k\}$$

the lattice generated by $\mathbf{B}$.

For a proof that every discrete subgroup of $(\mathbb{R}^n, +)$ actually is a lattice in the sense of Definition 2.26 see Appendix A. The following Lemma provides the existence of a decomposition which is generally known as QR-decomposition or Gram Schmidt decomposition.

Lemma 2.12. *Let $n \geq k$ and let $\mathbf{A} \in \mathbb{R}^{n \times k}$ be a full rank matrix. Then there exists an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ (i.e. a $\mathbf{Q}^T \cdot \mathbf{Q} = \mathbf{I}$) and an upper triangular matrix $\mathbf{R} \in \mathbb{R}^{n \times k}$ such that*

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}.$$

For a proof see e.g. [HJ86]. Using Lemma 2.12 it can be established that a lattice actually is *well separated*.

Lemma 2.13. *Let $\Lambda = \Lambda(\mathbf{B})$ be a lattice. Then there exists a $\lambda_1 > 0$ such that it holds for every $\mathbf{x} \in \Lambda \backslash \{\mathbf{0}\}$ that $\|\mathbf{x}\| > \lambda_1$.*

Proof. Each $\mathbf{x} \in \Lambda \backslash \{\mathbf{0}\}$ can be written as $\mathbf{x} = \mathbf{B} \cdot \mathbf{z}$ for a $\mathbf{z} \in \mathbb{Z}^k \backslash \{\mathbf{0}\}$. By Lemma 2.12 there exists an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$ such that $\mathbf{B} = \mathbf{Q} \cdot \mathbf{R}$. Thus it holds that

$$\|\mathbf{x}\| = \|\mathbf{BZ}\| = \|\mathbf{Q} \cdot \mathbf{Rz}\| = \|\mathbf{Rz}\|,$$

as $\mathbf{Q}$ is an orthogonal matrix (and thus does not change the norm). Assume that

$$\mathbf{R} = \begin{pmatrix} r_{11} & * & \cdots & * \\ 0 & r_{11} & \cdots & * \\ 0 & 0 & \ddots & * \\ 0 & 0 & \cdots & r_{kk} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

If $z_k \neq 0$ then $\|\mathbf{Bz}\| \geq r_{kk}$. If $z_k = 0$ and $z_{k-1} \neq 0$ then $\|\mathbf{Bz}\| \geq r_{k-1,k-1}$ and so forth. Consequently, $\|\mathbf{Bz}\| \geq \min_{j=1,\ldots,k} r_{jj} > 0$ as $\mathbf{z} \in \mathbb{Z}^k \backslash \{\mathbf{0}\}$. Thus, setting $\lambda_1 = \min_{j=1,\ldots,k} r_{jj}$ the statement of the lemma follows. □

The successive minima $\lambda_1, \ldots, \lambda_k$ of a lattice Λ are defined as the lengths of the shortest independent vectors of a lattice in ascending order.

Definition 2.27 (Successive Minima)**.** *Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice of rank k. We say that $\lambda_1, \ldots, \lambda_k$ are successive minima of Λ if*

$$\lambda_i = \|\mathbf{x}_i\| = \min_{\mathbf{x} \in \Lambda \backslash \mathsf{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})} \|\mathbf{x}\|.$$

for some linearly independent $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \Lambda$.

Notice that there exist lattices with short independent vectors that do not possess short bases. For instance, consider the lattice generated by the matrix $\mathbf{B} \in \mathbb{Z}^{n \times n}$ with

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \frac{1}{2} \\ 0 & 1 & \cdots & 0 & \frac{1}{2} \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & \frac{1}{2} \\ 0 & 0 & \cdots & 0 & \frac{1}{2} \end{pmatrix}.$$

A moment of contemplation leads to the insight that this basis is shortest possible for Λ. However, the last column of $\mathbf{B}$ is a vector of length $\sqrt{n}/2$. In turn, as

$$\mathbf{e}_n = \mathbf{B} \cdot \begin{pmatrix} -1 \\ \vdots \\ -1 \\ 2 \end{pmatrix} \in \Lambda$$

it holds that $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\} \subseteq \Lambda$ is a full rank set of independent vectors, each with length 1.

The dual of a lattice is defined in a similar manner to the dual of a code.

Definition 2.28. *Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice of rank n. The dual lattice $\Lambda^\perp$ of Λ is defined by*

$$\Lambda^\perp = \{\mathbf{x} \in \mathbb{R}^n \mid \forall \mathbf{y} \in \Lambda : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\},$$

i.e. $\Lambda^\perp$ consists of all vectors that have scalar product in $\mathbb{Z}$ with all vectors in Λ.

First notice that $\Lambda^\perp$ actually is a lattice. If $\mathbf{B} \in \mathbb{R}^{n \times n}$ is a basis of Λ, then $\mathbf{B}$ has rank n, thus $\mathbf{B}$ has a inverse $\mathbf{B}^{-1}$. Clearly, if $\mathbf{x} \in \Lambda^\perp$, then $\mathbf{B}^T\mathbf{x} = \mathbf{z} \in \mathbb{Z}^k$. Therefore $\mathbf{x} = (\mathbf{B}^{-1})^T\mathbf{z}$ and therefore $\mathbf{x} \in \Lambda((\mathbf{B}^{-1})^T)$. Furthermore, it can easily be seen that $\Lambda((\mathbf{B}^{-1})^T) \subseteq \Lambda^\perp$, thus it holds $\Lambda((\mathbf{B}^{-1})^T) = \Lambda^\perp$ and we get that $\Lambda^\perp$ is a lattice.

We will consider two classes of lattices more closely that will appear in this work: Integer lattices and q-ary lattices. A lattice Λ is called an *integer lattice* if $\Lambda \subseteq \mathbb{Z}^n$. An integer lattice Λ is called a *q-ary lattice* if $\Lambda = \Lambda + q\mathbb{Z}^n$, i.e. Λ is invariant by shifts of integer multiples of q.

Definition 2.29. *A lattice Λ is called an integer lattice, if $\Lambda = \Lambda(\mathbf{B})$ for an integer matrix $\mathbf{B} \in \mathbb{Z}^{n \times k}$. Let q be an integer. A lattice Λ is called q-ary lattice, if there exists matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times k}$ such that $\Lambda = \Lambda_q(\mathbf{B}) = \{\mathbf{y} \in \mathbb{Z}^n \mid \exists \mathbf{x} \in \mathbb{Z}_q \text{ s.t. } \mathbf{y} = \mathbf{Bx} \bmod q\}$.*

First notice that q-ary lattice $\Lambda_q(\mathbf{B}) \subset \mathbb{Z}^n$ always have full rank. This can be seen as $q \cdot \mathbb{Z}^n \subset \Lambda_q(\mathbf{B})$, i.e. the q-multiples of the unit vectors in $\mathbb{Z}^n$ are always in $\Lambda_q(\mathbf{B})$. The dual of a q-ary lattice $\Lambda_q(\mathbf{B})$ has a particularly simple form. For every full rank $\mathbf{B} \in \mathbb{Z}_q^{n\times k}$ there exists a full rank $\mathbf{H} \in \mathbb{Z}_q^{(n-k)}$ such that $\mathbf{H} \cdot \mathbf{B} = \mathbf{0}$. Thus it holds for all $\mathbf{x} \in \Lambda_q(\mathbf{B})$ that $\mathbf{H} \cdot \mathbf{x} = \mathbf{0} \mod q$ and therefore $(\Lambda_q(\mathbf{B}))^\perp = \frac{1}{q} \cdot \Lambda_q(\mathbf{H}^T)$.

We will briefly proof an analogue of the Gilbert Varshamov bound for q-ary lattices.

Lemma 2.14. *Let $\epsilon > 0$ and let q be a modulus. Let n, k and r be such that*

$$k \leq (1 - \log_q(2r) - \epsilon)n.$$

Let $\mathbf{A}$ be chosen uniformly at random from $\mathbb{Z}_q^{m\times n}$. Then the shortest vector of the lattice $\Lambda_q(\mathbf{A})$ has length (in the $\|\cdot\|_2$-norm) greater than r, except with negligible probability.

Proof. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be chosen uniformly at random. Fix a vector $\mathbf{x} \neq 0 \in \mathbb{Z}_q^n$. Then the vector $\mathbf{A} \cdot \mathbf{x}$ is distributed uniformly at random in $\mathbb{Z}_q^m$. The ball

$$\mathsf{B}_n(r) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq r\}$$

contains at most $(2r)^n$ integer points. Thus it holds that

$$\Pr_{\mathbf{A}}[\|\mathbf{A} \cdot \mathbf{x}\| \leq r] \leq \Pr_{\mathbf{A}}[\mathbf{A} \cdot \mathbf{x} \in \mathsf{B}_n(r)] \leq \left(\frac{2r}{q}\right)^n.$$

A union-bound yields that

$$\Pr[\exists x \neq 0 \in \mathbb{Z}_q^n : \|\mathbf{A}\mathbf{x}\| \leq r] \leq \frac{(2r)^n}{q^{n-k}} = q^{k-(1-\log_q(2r))n} \leq q^{-\epsilon n},$$

as $k \leq (1 - \log_q(2r) - \epsilon)m$. This immediately yields

$$\Pr[\forall x \neq 0 \in \mathbb{Z}_q^n : \|\mathbf{A}\mathbf{x}\| \geq r] \geq 1 - q^{-\epsilon n},$$

which is overwhelming in n. □

2.6.1. Discrete Gaussians

Gaussian distributions play an important role in the study of lattices [MR04]. For a parameter $s > 0$, the n-dimensional gaussian function $\rho_s : \mathbb{R}^n \to (0, 1]$ is defined by

$$\rho_s(\mathbf{x}) = e^{-\frac{\pi}{s^2}\|\mathbf{x}\|^2}.$$

In order to obtain a probability density from the gaussian function, we need to normalize it on the desired support. For instance, to obtain a probabilty distribution on $\mathbb{R}^n$ we need to normalize $\rho_s(\mathbf{x})$ by

$$\int_{\mathbb{R}^n} \rho(\mathbf{x})\mathrm{d}\mathbf{x} = s^n.$$

A particularly useful feature of the gaussian function is its invariance under orthogonal operations, i.e. rotary reflections. If $\mathbf{T} \in \mathbb{R}^{n\times n}$ is such that $\mathbf{T}^T \cdot \mathbf{T} = \mathbf{I}$, then

$$\rho_s(\mathbf{T}\mathbf{x}) = e^{-\frac{\pi}{s^2}\|\mathbf{T}\mathbf{x}\|^2} = e^{-\frac{\pi}{s^2}(\mathbf{T}\mathbf{x})^T\mathbf{T}\mathbf{x}} = e^{-\frac{\pi}{s^2}\mathbf{x}^T\mathbf{T}^T\mathbf{T}\mathbf{x}} = e^{-\frac{\pi}{s^2}\|\mathbf{x}\|^2} = \rho_s(\mathbf{x})$$

The gaussian function ρ_s can also be used to define gaussian distributions on lattices $\Lambda \subseteq \mathbb{R}^n$. We define the discrete gaussian $D_{\Lambda,s}$ as follows.

Definition 2.30. *Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. The discrete Gaussian distribution $D_{\Lambda,s}$ on Λ with parameter s is given by the probility mass function*

$$\Pr[\mathbf{x} = \tilde{\mathbf{x}}] = \frac{1}{\rho_s(\Lambda)}\rho_s(\tilde{\mathbf{x}})$$

where $\mathbf{x}$ is distributed according to $D_{\Lambda,s}$ and $\tilde{\mathbf{x}} \in \Lambda$. $\rho_s(\Lambda)$ is defined by $\rho_s(\Lambda) = \sum_{\mathbf{z}\in\Lambda} \rho_s(\mathbf{z})$. For $\Lambda = \mathbb{Z}$ we set $D_s = D_{\mathbb{Z},s}$.

The following Lemma, due to Micciancio and Peikert [MP12], establishes a convenient tail bound for discrete Gaussians.

Lemma 2.15 (Micciancio Peikert [MP12] Lemma 2.8)**.** *Let λ be a security parameter. Let $n = \mathsf{poly}(\lambda)$, $\Lambda \subseteq \mathbb{R}^n$ be a lattice and $s > 0$. Let $\mathbf{x}$ be distributed according to $D_{\Lambda,s}$. Then it holds that*

$$\Pr[\|\mathbf{x}\| \geq t\sqrt{n}] \leq 2 \cdot n \cdot e^{-\frac{\pi}{s^2}t^2}$$

for any $t > 0$. In particular, if $t = \sqrt{\omega(\log(\lambda))}$, then $\mathbf{x}$ is $t \cdot s \cdot \sqrt{n}$ bounded, except with negligible probability.

Discrete gaussians can be sampled efficiently, given that the parameter s is sufficiently lower bounded. In this work we are only interested in the basic discrete gaussian distribution $D_{\mathbb{Z},s}$ on $\mathbb{Z}$. The following Lemma due to Gentry, Peikert and Vaikuntanathan [GPV08] shows that D_s can be sampled efficiently.

Lemma 2.16 (Gentry Peikert Vaikuntanathan [GPV08])**.** *Let λ be a security parameter. There exists an efficient algorithm sampling a distribution statistically close to $D_{\mathbb{Z},s}$, given that $s = \omega(\log(\lambda))$.*

2.6.2. Computational Problems in Lattices

The main reason why complexity theory and cryptography are interested in lattices are the natural computational problems that arise in lattices. Classical worst case problems in lattices are the shortest vector problem (SVP) and the shortest independent vectors problem (SIVP). The shortest vector problem is usually posed as a promise problem, for which a gap between the `YES` and `NO` instances exists.

Problem 2.1 (GapSVP promise problem)**.** *Let $\gamma > 0$. Instances of the gap shortest vector problem GapSVP_γ are pairs $(\mathbf{B}, r)$, where $\mathbf{B} \in \mathbb{Z}^{n\times k}$ is a basis matrix of a lattice $\Lambda = \Lambda(\mathbf{B})$ and $r > 0$ is a radius. Let λ_1 be the length of the shortest vector of Λ. $(\mathbf{B}, r)$ is a `YES` instance of GapSVP_γ if $\lambda_1 \leq r$ and a `NO` instance of GapSVP_γ if $\lambda_1 > \gamma \cdot r$.*

An algorithm that solves GapSVP_γ must classify `YES` instances and `NO` instances correctly, while all other instances (i.e. $(\mathbf{B}, r)$ with $r < \lambda_1(\mathbf{B}) \leq \gamma \cdot r$) may be classified arbitrarily. Given oracle access to an algorithm that solves GapSVP_γ, one can efficiently approximate the length of the shortest vector of a lattice $\Lambda = \Lambda(\mathbf{B})$ to within a factor of 2γ using binary search in r.

Currently, the best algorithms to solve the GapSVP problem solve the SVP problem by explicitly computing the shortest vector. These algorithms are usually based on the classical lattice reduction algorithm of Lenstra, Lenstra and Lovasz (LLL) [LLL82] and its improvements [NS00, GHGKN06, GN08].

The shortest independent vectors problem is a search problem. Given a basis $\mathbf{B}$ of a lattice $\Lambda = \Lambda(\mathbf{B})$, the task is to compute a full rank set of short vectors of Λ.

Definition 2.31 (Shortest Independent Vectors problem)**.** *Let $\gamma > 0$. Instances of the shortest independent vectors problem SIVP_γ are given by a basis matrix $\mathbf{B} \in \mathbb{Z}^{n \times k}$ of a lattice $\Lambda = \Lambda(\mathbf{B})$. The goal of the problem is to compute a set of linearly independent vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_k\} \subseteq \Lambda$ such that $\max_i \|\mathbf{a}_i\| \leq \gamma \cdot \lambda_k$, where λ_k is the k-th successive minimum of Λ.*

3. Decoding Problems for Cryptography

I have developed an encryption software package that I can best describe as a ONE-TIME-PAD GENERATOR.

Anthony Stephen Szopa posting to sci.crypt, August 8, 1997

Is it time for another one of these already? Oh, bother.

Bruce Schneier posting to sci.crypt, August 8, 1997 - in response to the Szopa quote

3.1. Introduction

In this Chapter we will provide a survey of the Learning Parity with Noise (LPN) and Learning with Errors (LWE) problems. We will provide basic reductions between the different flavors of the problems, briefly discuss known attacks and study parameter sets for which the problems can be conjectured to be hard. We will also provide a novel result that bases the hardness of an LPN problem with unbounded samples on LPN with bounded samples, trading samples for noise. The following outline is not meant to provide a comprehensive overview but a rather a brief introduction in the fields of coding and lattice based cryptography.

3.1.1. Coding Based Cryptography

To the best of our knowledge, the first use of decoding problems in cryptography is due to McEliece [McE78]. While public key cryptography was still in its early infant stage, McEliece proposed a public key encryption scheme which is based on the

assumption that for certain codes C it is hard to decode sufficiently noisy codewords, if the code is not given in a canonic form that allows efficient decoding. However, someone who knows how the given code can be transformed into its canonic form will be able to decode noisy codewords efficiently. McEliece's original proposal is best explained in terms of trapdoor functions. For a given code C, take a canonical representation in form of a generator matrix $\mathbf{G}$ and *scramble* $\mathbf{G}$, thereby obtaining a generator matrix $\mathbf{A}$ of a code C' equivalent to C. The public key of this scheme is the generator matrix $\mathbf{A}$, while the secret key is the transformation from $\mathbf{G}$ to $\mathbf{A}$. The evaluation of the trapdoor function $f_{\mathbf{A}}$ is basically the simulation of a noisy channel. Given a message $\mathbf{s}$ and a noise term $\mathbf{e}$, set

$$f_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}.$$

In terms of communications engineering, the function value $f_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is a message $\mathbf{s}$ encoded by using the generator matrix $\mathbf{A}$ and passed through a noisy channel that introduces the noise $\mathbf{e}$. Now, given $\mathbf{y} = f_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$, the McEliece assumption is that it is infeasible (for any efficient algorithm) to recover the message $\mathbf{s}$, even though $\mathbf{s}$ is information-theoretically uniquely defined by $\mathbf{A}$ and $\mathbf{y}$. Anyone who knows the secret key however, which is the transformation from the canonical generator matrix $\mathbf{G}$ to the scrambled generator matrix $\mathbf{A}$ will be able to recover $\mathbf{s}$ from $\mathbf{y}$, by restating the decoding problem in terms of $\mathbf{G}$, for which the unique solution can be found efficiently using the efficient decoder of C. McEliece's original proposal was to use binary Goppa codes (c.f. Section 2.5.6) for the code C, and scramble them by choosing an equivalent code at random. In Chapter 5 we will provide more details on the McEliece assumption and the construction of an IND-CCA2 secure public key encryption scheme from a somewhat stronger McEliece assumption.

From a contemporary point of view, it is straightforward to construct IND-CPA public key encryption from any (injective) trapdoor function using hardcore predicates (c.f. Section 2.4.2). The early proposals of the McEliece cryptosystem were insecure and it took considerable time until a provably secure variant of the McEliece encryption scheme was found that achieves the same efficiency as the original proposal [NIKM08]. Niederreiter [Nie86] proposed a *dual* version of the McEliece cryptosystem. In Niederreiter's proposal, the trapdoor function is formulated in terms of parity check matrices rather than generator matrices of codes. The secret key is a parity check matrix $\mathbf{H}$ of an efficiently decodable code C and the public key is a parity check matrix $\mathbf{B}$ of a code C' equivalent to C. The trapdoor function $f_{\mathbf{B}}$ is given by

$$f_{\mathbf{B}}(\mathbf{e}) = \mathbf{B}\mathbf{e},$$

i.e. $f_{\mathbf{B}}(\mathbf{e})$ is the syndrome of a random error $\mathbf{e}$. To invert $f_{\mathbf{B}}$ using the trapdoor $\mathbf{H}$, the syndrome decoding problem $\mathbf{y} = \mathbf{B}\mathbf{e}$ is restated in terms of $\mathbf{H}$. Niederreiter also proposed to use binary Goppa codes for this scheme. While many McEliece and Niederreiter variants proposed later were severely flawed [Nie85, Sid94, JM96, Gab05, BCGM07], the McEliece and Niederreiter trapdoor functions based on binary Goppa codes are considered secure to this day.

Both the schemes of McEliece [McE78] and Niederreiter [Nie86], as well as their refinements [NIKM08, MTSB13], assumed the hardness of decoding specific *structured* codes. These schemes rely on the existence of efficient decoding algorithms for the *unscrambled* codes in an essential way. In coding theory, it has long been known that random codes are among the best possible codes. Random linear codes achieve

both the Shannon limit for binary symmetric channels ([Sha48], also Theorem 2.3) and the Gilbert-Varshamov bound (Theorem 2.4). Random linear codes carry, besides their linearity, no further structure. However, all known efficient decoding algorithms for linear codes make use of the codes' structure in an essential way. It is generally conjectured that it is infeasible to decode noisy codewords of random linear codes significantly better than brute force search. While decoding linear codes is known to be NP-complete in the worst case [BMVT78], the average case hardness of the problem can, at the current state of knowledge, only be conjectured.

We will take a brief detour to illuminate the connection between cryptography and computational learning theory [Val84]. Computational learning theory, the theoretical branch of machine learning, is concerned with problems of the following kind: Given access to an oracle $\mathcal{O}_{samp}$ that generates (possibly noisy) *labeled samples*[1] of a function $f \in \mathcal{F}$, does there exist an efficient algorithm $\mathcal{A}$ computing a succinct description of f? $\mathcal{F}$ is a class of boolean functions and a labeled sample is a pair $(\mathbf{a}, f(\mathbf{a}) + e)$ where $\mathbf{a}$ is a randomly chosen for f input and e a noise term.

The first time the connection between hard learning problems and cryptography was made explicit was by Blum et al. [BFKL93]. Blum et al. show that certain hard learning problems imply fundamental cryptographic primitives like one-way functions, pseudorandom generators and private key encryption [2]. In particular, Blum et al. [BFKL93] introduced the so called learning parity with noise (LPN) problem. For this problem, the class $\mathcal{F}$ is the class of *parity functions* $\mathbb{F}_2^n \to \mathbb{F}$. Parity functions have a particularly simple structure: Given an input $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$, the function value of $f(\mathbf{a})$ is the sum modulo 2 (or parity) of a subset of the $a_1, \ldots, a_n$, i.e.

$$f(\mathbf{a}) = \sum_{i \in S}^{n} a_i,$$

for an $S \subseteq \{1, \ldots, n\}$. Using very little linear algebra, such functions can be more conveniently expressed as an inner product

$$f(\mathbf{a}) = \langle \mathbf{s}, \mathbf{a} \rangle,$$

where $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{F}_2^n$ is the characteristic vector of the set S, i.e. $s_i = 1$ iff $i \in S$ and $s_i = 0$ otherwise. Thus, the function f can be represented by the vector $\mathbf{s}$. We will henceforth refer to $\mathbf{s}$ as the secret, since the learning algorithms goal is to learn $\mathbf{s}$. Observe that a linear function f is easy to learn in the noise free case. A learning algorithm could learn the function f from n (or slightly more) samples, by computing $\mathbf{s}$ using gaussian elimination as soon as it gets a full rank system of labeled samples $\{(\mathbf{a}_i, f(\mathbf{a}_i))\}$. Thus, to make the task of learning f non-trivial, the LPN sample oracle $\mathcal{O}_{LPN}$ provides *noisy* samples to the learning algorithm. The noise terms e are chosen independently from a Bernoulli distribution $\mathsf{Ber}(\rho)$, for some $\rho \in (0, 1/2)$. This means that $e = 1$ with probability ρ and $e = 0$ with probability $1 - \rho$. Thus, $\mathcal{O}_{LPN}$ generates samples of the form

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e),$$

where $\mathbf{a}$ is chosen uniformly at random from $\mathbb{F}_2^n$ and e follows the Bernoulli distribution $\mathsf{Ber}(\rho)$. The LPN problem can be seen as a (discrete analogue of a) linear

[1] In computational learning theory literature these are usually called *examples*, while cryptographic literature usually refers to these as *samples*

[2] Though one-way functions are known to imply pseudorandom generators and private key encryption [ILL89, HILL99], the direct constructions given in [BFKL93] are arguably more efficient.

curve-fitting problem. Given arbitrarily many noisy observations $(\mathbf{a}_i, \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i)$, where the sample points $\mathbf{a}_i$ are randomly chosen from $\mathbb{F}_2^n$ and the noise terms e_i are chosen from a $\mathsf{Ber}(\rho)$, find the best explanation $\mathbf{s}$ for these observations.

There are two features of this problem, which were already observed by Blum et. al [BFKL93], that make this problem particularly interesting for cryptography. The first feature is *random self reducibility* of the secret $\mathbf{s}$. This property basically means that if there exists even one $\mathbf{s} \in \mathbb{F}_2^n$ (equivalently an $f \in \mathcal{F}$) such that $\mathbf{s}$ is *hard to learn* given samples from $\mathcal{O}_{LPN}$, then this must also be the case for all but a tiny (in fact negligible) fraction of the $\mathbf{s}' \in \mathbb{F}_2^n$. Conversely interpreted, this can be seen as a *weak* form of worst-to-average case reduction: If there exists a learning algorithm $\mathcal{L}$ that works for a non-negligible fraction of the $\mathbf{s} \in \mathbb{F}_2^n$, then there exists a learning algorithm $\mathcal{L}'$ that works for all $\mathbf{s} \in \mathbb{F}_2^n$. We call this a *weak* random self reduction property because we still rely on the fact that the sample points $\mathbf{a}$ are chosen randomly (i.e. average case) and not maliciously (i.e. worst case).

The second outstanding feature of the LPN problem is pseudorandomness. This basically means that if it is hard to learn a random secret $\mathbf{s}$ from samples of $\mathcal{O}_{LPN}$, then the samples generated by $\mathcal{O}_{LPN}$ must already look random, i.e. it is infeasible to distinguish $\mathcal{O}_{LPN}$ from an oracle $\mathcal{O}_{rand}$ that outputs samples $(\mathbf{a}, u)$ where u is chosen randomly and independent of $\mathbf{a}$. This feature of the LPN problem is particularly useful as yields direct constructions of pseudorandom generators and weakly random functions [PS08]. A reduction establishing the pseudorandomness of a problem is usually called search-to-decision reduction. In the case of the LPN problem, a series of works [BFKL93, HB01, KS06, KSS10, AIK07] have provided increasingly tighter search-to-decision reductions.

The LPN problem has found numerous applications in symmetric cryptography, ranging from efficient pseudorandom number generation, private key encryption to authentication [HB01, DKL09, ACPS09, KPC+11, JKPT12, HKL+12]. What makes LPN based schemes attractive from a practical viewpoint is their simple structure. All computations required can usually be performed by simple arithmetic gates such as AND and XOR. LPN based schemes were thus proposed for use in low power devices such as RFID tags [DLZW13].

We will now reconsider the learning problem LPN from a coding theoretic perspective. Assume that the learning algorithm may only learn a (polynomially) bounded number m of samples. Then all the samples can be provided in one shot. Given samples $\{(\mathbf{a}_i, y_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)\}$, we can arrange the $\mathbf{a}_i$ as the rows of a matrix $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ and the y_i and e_i as the components of vectors $\mathbf{y} \in \mathbb{F}_2^m$ and $\mathbf{e} \in \mathbb{F}_2^m$ and write the samples in the form

$$(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}).$$

Thus, the task of learning a secret $\mathbf{s}$ from noisy samples can be seen as decoding a noisy codeword $\mathbf{y}$ of a random linear code $\mathbf{A}$. Obviously, a learning algorithm that learns $\mathbf{s}$ from a bounded number of samples can also learn $\mathbf{s}$ from an unbounded number of samples. It is unknown if the converse also holds for arbitrary noise parameters $\rho < \frac{1}{2}$. In Section 3.5 we show that this is the case if one tolerates an increase in the noise of the unbounded sample instance.

Given this coding theoretic interpretation of the LPN problem, in hindsight the search-to-decision equivalence of the LPN problem seems natural. This comes as no surprise, as virtually all constructions of hardcore predicates (e.g. Theorem 2.2) for arbitrary one-way functions use coding theoretic techniques in an essential way.

This structural compatibility becomes most evident in the proof of Lemma 3.3.

We have thus far not explicitly discussed the choice of the noise rate ρ. All LPN applications mentioned thus far are located in the realm of symmetric cryptography, thus there is no need to embed trapdoors in LPN instances. Alekhnovich [Ale03] showed that if the noise rate is chosen sufficiently low, *asymptotically less than constant*, then one can construct public key encryption schemes based on the hardness of LPN. While McEliece [McE78] and Niederreiter [Nie86] needed the extra assumption that certain structured families of codes are hard to decode, this requirement is not needed in Alekhnovichs public key encryption scheme. However in order for Alekhnovich's trapdoor to work, the noise rate ρ needs to be chosen as low as $O(n^{-\frac{1}{2}})$, this means in particular that asymptotically the noise rate tends towards 0. While Theorem 3.3 shows that this low noise LPN variant implies LPN with a constant noise rate, the converse is believed to be false. In Chapter 6 we provide the construction of an IND-CCA2 secure encryption scheme based on such an *Alekhnovich-type* LPN assumption.

3.1.2. Lattice Based Cryptography

As described in Section 2.6, lattices can be seen as a euclidean analogue to linear codes. As decoding problems for linear codes turned out to be a valuable source of computational hardness, it was natural to extend this study to lattice problems.

The first (published) proposal of a lattice based public key encryption scheme is due to Goldreich, Goldwasser and Halevi [GGH97b]. The ideas underlying the GGH scheme follow the blueprint of the McEliece encryption scheme. For simplicity, we will also describe this scheme in terms of trapdoor functions. The private key of the scheme is a basis matrix $\mathbf{B} \in \mathbb{Z}^{n \times n}$ for a full-rank lattice $\Lambda \subseteq \mathbb{Z}^n$. The basis $\mathbf{B}$ is chosen in a way such that its columns are *nearly* orthogonal. The public key of the scheme is another basis matrix $\mathbf{A}$ of the lattice Λ which should not have the *nice* geometric properties of $\mathbf{B}$. In [GGH97b] several heuristics are discussed how such a basis $\mathbf{A}$ can be computed from $\mathbf{B}$ using simple randomization. The trapdoor function $f_{\mathbf{A}}$ has syntatically the same form as the McEliece trapdoor function,

$$f_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$$

where $\mathbf{s}$ is chosen from a distribution on $\mathbb{Z}^n$ with a sufficient amount of entropy and $\mathbf{e} \in \mathbb{Z}^n$ is chosen from a distribution of short vectors. Thus, similar to the McEliece cryptosystem, the function value $f_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ can be seen as the output of a noisy channel that received an encoded (or modulated) input $\mathbf{A}\mathbf{s}$. To invert $\mathbf{y} = f_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$, the decoding problem is expressed in terms of the basis $\mathbf{B}$. As the column vectors of $\mathbf{B}$ are nearly orthogonal, an efficient lattice decoding algorithm like Babai's nearest hyperplane algorithm [Bab85] can be used to find the lattice point closest to $\mathbf{y}$.

The generation of the public basis $\mathbf{A}$ in [GGH97b] was highly heuristic. Micciancio [Mic01] later showed that there is a *best possible way* of choosing the basis matrix $\mathbf{A}$, namely by choosing $\mathbf{A}$ to be the Hermite normal form (HNF) of the matrix $\mathbf{B}$. The HNF basis actually is a normal form, i.e. it only depends on the lattice $\Lambda = \Lambda(\mathbf{B})$ and is independent of the specific input basis. Moreover, the HNF of a matrix $\mathbf{B}$ can be computed efficiently. Thus, for $\mathbf{A} = \mathsf{HNF}(\mathbf{B})$ the decoding problem $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$ is as hard as possible for the lattice Λ, as if $\mathbf{s}$ could be computed efficiently from $\mathbf{A}$ and $\mathbf{y}$, then one could solve the decoding problem for *any* other basis $\mathbf{A}'$ of Λ, simply by computing the HNF $\mathbf{A}$ of $\mathbf{A}'$ (which is unique for Λ), and then solving the decoding problem in terms of $\mathbf{A}$. Micciancio [Mic01] also proposed a dual version

of the GGH cryptosystem, which might just be considered a lattice analogue of the (coding based) Niederreiter cryptosystem [Nie86] (described above). We will describe a slightly modified version of Micciancio's trapdoor function, to highlight the similarities to the Niederreiter trapdoor function. Given a full rank matrix $\mathbf{A} \in \mathbb{Z}^{n \times n}$, we can efficiently compute a matrix $\mathbf{H} \in \mathbb{Z}^{n \times n}$ such that $\mathbf{H} \cdot \mathbf{A} = \det(\mathbf{H})\mathbf{I}$. This basically means that $\frac{1}{\det(\mathbf{H})} \cdot \mathbf{H}$ is a basis matrix of the dual lattice $\Lambda^{\perp}(\mathbf{A})$. The trapdoor function $f_{\mathbf{H}}$ is then given by

$$f_{\mathbf{H}}(\mathbf{e}) = \mathbf{He} \mod \det(\mathbf{H}).$$

Similar to the Niederreiter trapdoor function, $f_{\mathbf{H}}(\mathbf{e})$ can be considered as the syndrome of the error vector $\mathbf{e}$. To invert $f_{\mathbf{H}}(\mathbf{e})$, we proceed similar to the GGH inversion procedure. Given $\mathbf{y} = f_{\mathbf{H}}(\mathbf{e})$, express the decoding problem in terms of the private basis $\mathbf{B}$ and recover the error vector $\mathbf{e}$. We've omitted all the optimizations that can be applied given that the matrix $\mathbf{A}$ is in Hermite normal form.

Around the same time as Goldreich, Goldwasser and Halevi [GGH97b] published their lattice based cryptosystem, Hoffstein, Pipher and Silverman [HPS98] published a public key encryption scheme called NTRU [3] based on similar ideas. From a modern perspective, NTRU is based on hard problems in *ideal* lattices. A complete description of the NTRU cryptosystem is beyond the scope of this outline, we only mention that recently that there has been a renewed interest in the techniques used in NTRU, as these can be used to construct multi-key fully homomorphic encryption [LATV12] and (approximate) multilinear maps [GGH13]. Moreover, recent works have shown that variants of NTRU can be based on the so called Ring-LWE problem [LPR10, SS11].

The lattice based trapdoor functions discussed so far all rely on average case assumptions, i.e. we assume that the decoding problem is hard for lattices generated in a certain (probabilistic) way. Lattice based cryptography received widespread attention when Ajtai [Ajt96] showed that there exists an efficiently samplable distribution $\mathcal{L}$ of q-ary lattices with guaranteed worst-case hardness. Specifically, any algorithm that finds short vectors in a lattice Λ chosen from $\mathcal{L}$ with non-negligible probability, can be turned into an algorithm finding short vectors in *any* integer lattice. The distribution $\mathcal{L}$ has a particularly simple structure. First fix appropriate m, n and q. To sample a lattice in $\mathcal{L}$, simply choose a parity-check matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times m}$ uniformly at random and set

$$\Lambda = \Lambda_q^{\perp}(\mathbf{H}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{Hx} = \mathbf{0} \mod q\}.$$

Ajtai [Ajt96] then showed how to construct one-way functions based on the hardness of finding short vectors in such lattices Λ. Goldreich, Goldwasser and Halevi [GGH96] showed that Ajtai's one-way function is actually collision resistant [4]. For a $\mathbf{H} \in \mathbb{Z}_q^{n \times m}$ the hash function $h_{\mathbf{H}} : \{0,1\}^m \to \mathbb{Z}_q^n$ is given by

$$h_{\mathbf{H}}(\mathbf{x}) = \mathbf{Hx}.$$

Notice that this hash function actually is compressing if $m > n \cdot \log(q)$ (which is the binary representation size of elements of $\mathbb{Z}_q^n$). Collision resistance of $h_{\mathbf{H}}$ can now be argued as follows. Assume there exists an efficient algorithm $\mathcal{A}$ finding

[3] An acronym for "n-th truncated polynomials"

[4] In a black box sense, collision resistance is a stronger property than one-wayness [Sim98]

with non-negligible probability collisions for $h_{\mathbf{H}}$, i.e. pairs $\mathbf{x}, \mathbf{y} \in \{0,1\}^m$ such that $h_{\mathbf{H}} = h_{\mathbf{H}}(\mathbf{y})$. Since $h_{\mathbf{H}}$ is linear, such a collision immediately corresponds to a short vector $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \Lambda_q^{\perp}(\mathbf{H})$, as

$$\mathbf{Hz} = \mathbf{H}(\mathbf{x} - \mathbf{y}) = \mathbf{Hx} - \mathbf{Hy} = \mathbf{0}.$$

As $\mathbf{x}$ and $\mathbf{y}$ are both in $\{0,1\}^m$, $\mathbf{z}$ is in $\{-1,0,1\}^m$. Thus $\mathbf{z}$ is short as $\|\mathbf{z}\|_2 \leq \sqrt{m}$. Therefore, $\mathcal{A}$ can be used to find short vectors in $\Lambda_q^{\perp}(\mathbf{H})$. But such an algorithm implies an algorithm that finds short vectors in any lattice. Thus, finding collisions for $h_{\mathbf{H}}$ is as hard as finding short vectors in worst-case lattices. In modern terminology, the problem of inverting Ajtai's one-way function [Ajt96, GGH96] is called *short integer solution* (SIS) problem and has been studied extensively [yCN97, Mic04, MR04, MP13].

However, one-way and collision resistant hash functions both belong to the realm of symmetric cryptography, i.e. they are insufficient for public key cryptography. Ajtai and Dwork [AD97] proposed the first public encryption scheme based on the worst case hardness of a lattice problem. Namely, the cryptosystem of [AD97] is based on the unique shortest vector problem. In the unique shortest vector problem, it is guaranteed that all vectors that are linearly independent of the shortest vector are at least by a certain polynomial factor longer. Thus, this problem is only defined on a restricted class of lattices, while the problem of approximating the shortest vector can be defined on any lattice. Subsequently, the original constructions were improved [GGH97a] and stronger worst-to-average case connection were established for lattice problems [Mic98, Ajt99, Reg04, MR04, PR07, LM09]. This effort culminated in the the work of Regev [Reg05, Reg09], in which the Learning with Errors (LWE) problem was introduced. The LWE problem can be seen as a generalization of the LPN problem to larger moduli $q \geq 2$ and more general error distributions. Specifically, in the LWE problem an adversary/learning algorithm $\mathcal{A}$ is given access to an oracle $\mathcal{O}_{LWE}$ that generates samples of the form

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e),$$

where $\mathbf{s} \in \mathbb{Z}_q^n$ is a fixed secret, the vectors $\mathbf{a} \in \mathbb{Z}_q^n$ are chosen uniformly at random and e is an error term drawn from an error distribution χ. Like in the LPN problem, the goal of the adversary is to find the secret $\mathbf{s}$. As the LWE problem is similar in structure to the LPN problem, it also enjoys many of the desirable properties of the LPN problem. In particular, LWE is random self reducible in the sense that if LWE is hard for one secret $\mathbf{s}$, then LWE is also hard for a randomly chosen secret $\mathbf{s}$ [Reg05]. Moreover, many instantiations of the LWE problem enjoy searcht-to-decision equivalences like the LPN problem [Reg05, Pei09, MM11, MP12, MP13]. This means that the distribution generated by the LWE sample oracle $\mathcal{O}_{LWE}$ looks random to any efficient observer.

However, the most remarkable feature of LWE is its strong worst-to-average case connection if the problem is instantiated with a gaussian error distribution χ. Regev [Reg05] showed that if there exists an efficient (classical or quantum) algorithm learning the secret $\mathbf{s}$ given only oracle access to $\mathcal{O}_{LWE}$, then there exists an efficient *quantum* algorithm solving worst case lattice problems. We will provide more details on this worst-to-average case reduction in Section 3.7. At the current state of knowledge, assuming that a problem is hard for efficient quantum algorithms is a stronger assumption than assuming that the same problem is hard for efficient

classical algorithms. However, currently there are no quantum algorithms for worst case lattice problems known that outperform the best classical algorithms asymptotically. Basically, the best know quantum algorithm for worst-case lattice problems is running Grover's search algorithm [Gro96] combined with the best known classical algorithm. This yields a square-root speedup over the best classical algorithm. But as the best known classical algorithms for worst case lattice problems require exponential time, this speedup is asymptotically irrelevant. Thus, assuming that worst-case lattice problems are hard for quantum algorithms seems to be a valid assumption at the current state of knowledge. On the other hand, Regev's worst-to-average case reduction yields very strong approximation factors. In particular, the worst case lattice assumption on which (a particular instantiation of) LWE is based is that the shortest vector in any lattice is hard to approximate to withing a factor of $\tilde{O}(n^{1.5})$, while earlier works had approximation factors as large as $\tilde{O}(n^7)$ [AD97]. The worst-to-average case reduction for LWE was improved by Peikert [Pei09], who succeeded in removing the quantum part from Regev's reduction for certain parameter choices. Thus, Peikert could show that certain instantiations of LWE, that suffice for virtually all applications, are as hard as worst-case lattice problems for efficient classical algorithms. Building on this result, Brakerski et al. [BLP+13] recently showed that standard-LWE is as hard as worst case lattice problems.

Regev [Reg05] also provided a conceptually simple public key cryptosystem based on the hardness of LWE, which will serve as a template for our constructions in Chapter 6. From a technical point of view, the advantage of LWE over other lattice based hardness assumptions lies in structural simplicity, which hides all the technical intricacies concerning its worst-to-average case reduction and provides an easy-to-use assumption. As a consequence, LWE became maybe the biggest cryptographic success story of the last decade, giving rise to a wide range of instantiations of cryptographic tasks [GPV08, PVW08, PW08]. Many cryptographic primitives that previously only had instantiations based on specific number-theoretic assumptions were instantiated from the LWE problem, such as identity based encryption [CHKP10, ABB10]. More importantly, several cryptographic primitives that have no known instantiations from number-theoretic assumptions were instantiated from LWE, such as fully homomorphic encryption [BV11, BGV12, GSW13, BV14].

3.2. LPN and LWE as Decoding Problems

In this section, we will provide a formal definition of the LPN and LWE problems. For syntactic reasons, we will define LPN as a special case of LWE, though the LPN problem has been introduced before the LWE problem. Furthermore, we will provide treatment of common features of LPN and LWE. While structurally the LPN and LWE problems are very similar, the main difference between the two problems lies in the geometric aspects of their error distributions. For the LPN problem, the natural metric to measure the length of error terms is the hamming metric, while for the LWE problem a natural metric is the euclidean metric.

The standard formulations of the LPN and LWE problems provide an unbounded number of samples to the adversary, where the errors on each sample are chosen independently. Somewhat in abuse of notation we will also refer to decoding problems with not necessarily independent errors as LPN or LWE if the number of samples m is a priori fixed.

Problem 3.1 (Learning With Errors (LWE) and Learning Parity with Noise (LPN) Search Problems)**.** *Let λ be a security parameter, let $q \geq 2$ be a modulus and let $n = \mathsf{poly}(\lambda)$ be a positive integer.*

- ***Unbounded Samples Version:*** *Let χ be an error distribution defined on $\mathbb{Z}_q$. Let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ be chosen uniformly at random but then fixed. Let $\mathcal{O}_{LWE}$ be an oracle that generates samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_q^n$ is chosen uniformly at random and e is chosen according to χ. The goal of the $\mathsf{LWE}(n, q, \chi)$ problem is to find $\mathbf{s}$, given only oracle access to $\mathcal{O}_{LWE}$.*
- ***Bounded Samples Version:*** *Let $m = \mathsf{poly}(\lambda)$ be a positive integer and let $\boldsymbol{\chi}$ be an error distribution defined on $\mathbb{Z}_q^m$. Let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ be chosen uniformly at random. Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ be chosen uniformly at random and $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$. The goal of the $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ problem is to find $\mathbf{s}$, given $(\mathbf{A}, \mathbf{As} + \mathbf{e})$.*

We say that $\mathsf{LWE}(n, q, \chi)$ (resp. $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$) is hard*, if no PPT algorithm solves the problem with non-negligible probability.*

For errors distributions χ defined on $\mathbb{F}_2$ and $\boldsymbol{\chi}$ defined on $\mathbb{F}_2^m$ we define the learning parity with noise problems by $\mathsf{LPN}(n, \chi) = \mathsf{LWE}(n, 2, \chi)$ and $\mathsf{LPN}(n, m, \boldsymbol{\chi}) = \mathsf{LWE}(n, m, 2, \boldsymbol{\chi})$.

In the case of unbounded samples, the oracle $\mathcal{O}_{LWE}$ can be seen as the interface to an arbitrarily long code $\mathbf{A}$ and a noisy codeword $\mathbf{As}+\mathbf{e}$. In both the unbounded and the bounded version of the problem, an adversary/learning algorithm will receive at most a polynomial number of samples. The difference is however, that in the unbounded version this polynomial bound may depend on the adversary, while in the bounded version it is the same *for every* adversary.

The secret $\mathbf{s}$ in Problem 3.1 is chosen uniformly at random. At first glance it may seem that assuming that LWE/LPN is hard for a uniformly chosen secret is a stronger assumption than assuming that LWE/LPN is hard for any secret, i.e. a worst-case secret. It can be shown, however, that this it not that case. If there exists an efficient algorithm $\mathcal{A}$ that solves LWE/LPN for a uniformly chosen secret $\mathbf{s}$ with non-negligible probability, then there exists an efficient algorithm $\mathcal{A}'$ that solves LWE/LPN for *any* secret $\mathbf{s}$ with non-negligible probability (over the choice of the instance). The reason for this is that the problem is random self reducible, i.e. we can convert worst case instances of the problem into average case instances. This is substantiated in the following lemma.

Lemma 3.1 (e.g. [BFKL93, Reg05, Reg09])**.** *Let λ be a security parameter, let $q \geq 2$ be a modulus, let $n = \mathsf{poly}(\lambda)$ be a positive integer.*

1. *Let χ be an error distribution on $\mathbb{Z}_q$. Assume there exists a PPT algorithm $\mathcal{A}$ that solves $\mathsf{LWE}(n, q, \chi)$ with non-negligible probability. Then there exists a PPT algorithm $\mathcal{A}'$ which, for any $\mathbf{s} \in \mathbb{Z}_q^n$ finds $\mathbf{s}$ with non-negligible probability, given only access to an oracle $\mathcal{O}$ that generates samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ (with uniformly random $\mathbf{a} \in \mathbb{Z}_q^n$ and e chosen from χ).*
2. *Let $m = \mathsf{poly}(\lambda)$ and $\boldsymbol{\chi}$ be an error distribution on $\mathbb{Z}_q^m$. Assume there exists a PPT algorithm $\mathcal{A}$ that solves $\mathsf{LWE}(n, m, q, \chi)$ with non-negligible probability. Then there exists a PPT algorithm $\mathcal{A}'$, which, for any $\mathbf{s} \in \mathbb{Z}_q^n$ finds $\mathbf{s}$ with*

non-negligible probability given $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ *(where* $\mathbf{A}$ *is chosen uniformly at random and* $\mathbf{e}$ *is chosen from* $\boldsymbol{\chi}$*. The probability includes the choice of* $\mathbf{A}$ *and* $\mathbf{e}$*).*

In the first case, the success-probability of $\mathcal{A}'$ can be amplified to $1-\mathsf{negl}(\lambda)$ if there is an efficient way to test whether a given solution $\mathbf{s}$ is valid. This is for instance the case if the distribution χ generates *short* outputs with high probability. Then, candidate solutions $\mathbf{s}$ can be verified by testing whether $\mathbf{e}=\mathbf{y}-\mathbf{As}$ is short, where $(\mathbf{A}, \mathbf{y})$ is a set of test-samples.

Proof. We will only prove item 1, item 2 follows analogously. Assume towards contradiction that there exists a PPT algorithm $\mathcal{A}$ that solves $\mathsf{LWE}(n, q, \chi)$ with non-negligible probability ϵ. $\mathcal{A}'$ is given as follows.

Adversary $\mathcal{A}'$	Sample Oracle $\mathcal{O}'_{LWE}()$
Setup: Has access to an oracle $\mathcal{O}()$	$(\mathbf{a}, y) \leftarrow \mathcal{O}()$
$\mathbf{s}' \leftarrow_\$ \mathbb{Z}_q^n$	$y' \leftarrow y + \langle \mathbf{a}, \mathbf{s}' \rangle$
$\mathbf{s}'' \leftarrow \mathcal{A}^{\mathcal{O}'_{LWE}}(1^\lambda)$	Return $(\mathbf{a}, y')$
$\mathbf{s} \leftarrow \mathbf{s}'' - \mathbf{s}'$	
Return $\mathbf{s}$	

We will analyze the success-probability of $\mathcal{A}'$. Let $\mathbf{s}$ be the secret $\mathcal{A}'$ is supposed to find. Thus, the samples generated by $\mathcal{O}_{LWE}$ are of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ with uniformly chosen $\mathbf{a}$ and e chosen from χ. Thus it holds for the samples $(\mathbf{a}, y')$ generated by $\mathcal{O}'_{LWE}$ that

$$y' = y + \langle \mathbf{a}, \mathbf{s}' \rangle = \langle \mathbf{a}, \mathbf{s} \rangle + e + \langle \mathbf{a}, \mathbf{s}' \rangle = \langle \mathbf{a}, \mathbf{s} + \mathbf{s}' \rangle + e.$$

For any fixed $\mathbf{s}^*$, the random variable $\mathbf{s}^* = \mathbf{s} + \mathbf{s}'$ is distributed uniformly at random as $\mathbf{s}'$ is distributed uniformly random. Therefore, $\mathcal{O}'_{LWE}$ generates exactly the same distribution as the sample oracle $\mathcal{O}_{LWE}$ in problem 3.1 and consequently $\mathcal{A}$ finds $\mathbf{s}'' = \mathbf{s} + \mathbf{s}'$ with probability at least ϵ. If this is the case, $\mathcal{A}^*$ outputs $\mathbf{s} = \mathbf{s}'' - \mathbf{s}'$ with non-negligible probability ϵ, which concludes the proof. □

We will now discuss briefly how many samples m are needed in an information-theoretical sense to determine the secret $\mathbf{s}$ uniquely. Certainly, this depends on the error distribution. In the LPN case, i.e. $q = 2$, if the noise terms e are drawn independently, then χ can only be a Bernoulli distribution, hence $\chi = \mathsf{Ber}(\rho)$ for some $\rho \in [0, 1]$. If $\rho > \frac{1}{2}$, then by symmetry we can sample χ by $1 - x$, where x is distributed according to $\mathsf{Ber}(1-\rho)$. Thus, it is sufficient to consider $\rho \leq \frac{1}{2}$. Let $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ be chosen uniformly at random and $\mathbf{y} = \mathbf{As} + \mathbf{e}$, where $\mathbf{s}$ is chosen unformly at random and $\mathbf{e}$ is chosen from $\mathsf{Ber}(m, \rho)$. So, the question whether $\mathbf{s}$ can be uniquely determined from $\mathbf{A}$ and $\mathbf{y}$ can be reformulated as whether the code generated by $\mathbf{A}$ can correct errors from $\mathsf{Ber}(m, \rho)$. As $\mathbf{A}$ is chosen uniformly at random, Shannon's coding theorem for binary symmetric channels (Theorem 2.3) yields that if

$$n \leq (1 - H(\rho) - \epsilon)m,$$

for an arbitrary constant $\epsilon > 0$, then $\mathbf{A}$ and $\mathbf{y} = \mathbf{Ax} + \mathbf{e}$ uniquely determine $\mathbf{s}$ with overwhelming probability over the choice of $\mathbf{A}$ and $\mathbf{e}$. Thus,

$$m \geq \frac{n}{1 - H(\rho) - \epsilon}$$

samples are, with overwhelming probability, sufficient to determine $\mathbf{s}$ uniquely. For error distributions $\boldsymbol{\chi}$ defined on $\mathbb{F}_2^m$ which are not necessarily component-wise independent, the Gilbert-Varshamov bound provides an upper bound on the number of samples required. Let the error distribution $\boldsymbol{\chi}$ be such that for an $\mathbf{e}$ distributed according to $\boldsymbol{\chi}$ it holds that $\mathsf{wgt}(\mathbf{e}) \leq \gamma m$, except with negligible probability. If the code generated by $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ has minimum-distance greater than $2\gamma m$, then $\mathbf{A}$ and $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$ uniquely determine $\mathbf{s}$. The Gilbert Varshamov bound (Theorem 2.4) states that if

$$n \leq (1 + H(2\gamma) - \epsilon)m$$

for an arbitrary constant $\epsilon > 0$, then a matrix $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m \times n}$ chosen uniformly at random generates a linear code with minimum distance greater than $2\gamma m$, except with negligible probability. Thus, in this case

$$m \geq \frac{n}{1 - H(2\gamma) - \epsilon}$$

samples are sufficient (with overwhelming probability). For the LWE case, i.e. for larger moduli q, we usually require that the error distribution $\boldsymbol{\chi}$ is bounded in its euclidean norm. Specifically, we require that if $\mathbf{e}$ is distributed according to $\boldsymbol{\chi}$, then $\|\mathbf{e}\|_2 \leq B$ for some bound B, except with negligible probability. By Lemma 2.14 it holds that if $n \leq (1 - \log_q(4B) - \epsilon)m$ and $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{m \times n}$ is chosen uniformly at random, then the shortest vector of $\Lambda_q(\mathbf{A})$ has length at least $2B$, except with negligible probability. We can wrap this up in the following lemma.

Lemma 3.2. *Let λ be a security parameter. Let $m, n = \mathsf{poly}(n)$ be positive integers. Let $\epsilon > 0$ be an arbitrarily small constant.*

1. *Let $\rho = \rho(\lambda) \in [0, \frac{1}{2})$. If $m \geq \frac{n}{1-H(\rho)-\epsilon}$, then instances of $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ possess unique solutions, except with negligible probability.*

2. *Let $\boldsymbol{\chi}$ be a distribution on $\mathbb{F}_2^m$ which is γm bounded, except with negligible probability. If $m \geq \frac{n}{1-H(2\gamma)-\epsilon}$, then instances of $\mathsf{LPN}(n, m, \boldsymbol{\chi})$ posses unique solutions, except with negligible probability.*

3. *Let q be a modulus. Let $\boldsymbol{\chi}$ be a distribution on $\mathbb{Z}_q^m$, which is B-bounded, except with negligible probability. If $m \geq \frac{n}{1-\log_q(4B)-\epsilon}$, then instances of the $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ problem possess a unique solution, except with negligible probability.*

Finally, we mention that we will refer to LPN with Bernoulli distributed errors as standard LPN and to LWE with gaussian errors as standard LWE.

3.3. Seach-To-Decision Reductions

The Goldreich-Levin Theorem (Theorem 2.2) states that every search problem can be converted into a decisional problem. For many cryptographic tasks, especially pseudorandom generators and public key cryptosystems, this generic approach however leads to contrived and mostly impractical constructions. For many computational problems, such as the Diffie-Hellman problem and its variants in bilinear groups [DH76, Gam84], hardness of the decisional problem is an explicit assumption. While these assumptions either enable constructions in the first place [CS98, HW10]

or lead to more elegant and practical constructions, assuming the hardness of decisional problem is, at the current level of knowledge, a stronger assumption then assuming the hardness of the search problem.

This is different for the decoding problems introduced in the last section. The LWE and especially the LPN problems are somewhat unique among hardness assumptions in that their natural decisional problems can be proven to be as hard as the search versions. While the goal of search problems is to recover a secret $\mathbf{s}$, given either $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ or oracle access to $\mathcal{O}_{LWE}$, the goal of the decisional version of the problem is to distinguish $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ from uniformly random, respectively the oracle $\mathcal{O}_{LWE}$ from an oracle that produces uniformly random output.

Problem 3.2 (Decisional LWE and LPN Problems)**.** *Let λ be a security parameter, let $q \geq 2$ be a modulus and let $n = \mathsf{poly}(\lambda)$ be a positive integer.*

- ***Unbounded Samples Version:*** *Let χ be an error distribution defined on $\mathbb{Z}_q$. Let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ be chosen uniformly at random but then fixed. Let $\mathcal{O}_{LWE}$ be an oracle that generates samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \leftarrow_\$ \mathbb{Z}_q^n$ is chosen uniformly at random and e is chosen according to χ. Let $\mathcal{O}_{rand}$ be an oracle that generates samples of the form $(\mathbf{a}, u)$ with uniformly chosen $\mathbf{a} \leftarrow_\$ \mathbb{Z}_q^n$ and uniformly chosen $u \leftarrow_\$ \mathbb{Z}_q$. Let $\mathcal{O}$ be an oracle which is either $\mathcal{O}_{LWE}$ or $\mathcal{O}_{rand}$. The goal of the $\mathsf{DLWE}(n, q, \boldsymbol{\chi})$ problem is, given access to the oracle $\mathcal{O}$, to decide whether $\mathcal{O}$ is $\mathcal{O}_{LWE}$ or $\mathcal{O}_{rand}$.*

- ***Bounded Samples Version:*** *Let $m = \mathsf{poly}(\lambda)$ be a positive integer and let $\boldsymbol{\chi}$ be an error distribution defined on $\mathbb{Z}_q^m$. Let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ be chosen uniformly at random. Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ be chosen uniformly at random and $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$. Let $\mathbf{u} \leftarrow_\$ \mathbb{Z}_q^m$ be chosen uniformly at random. The goal of the $\mathsf{DLWE}(n, m, q, \boldsymbol{\chi})$ problem is, given $(\mathbf{A}, \mathbf{y})$, to decide whether $\mathbf{y} = \mathbf{As} + \mathbf{e}$ or $\mathbf{y} = \mathbf{u}$.*

We say that $\mathsf{DLWE}(n, q, \chi)$ (resp. $\mathsf{DLWE}(n, m, q, \boldsymbol{\chi})$) is hard*, if no PPT distinguisher distinguishes between the two distributions with non-negligible advantage.*

For error distributions χ defined on $\mathbb{F}_2$ and $\boldsymbol{\chi}$ defined on $\mathbb{F}_2^m$ we define the decisional learning parity with noise problems by $\mathsf{DLPN}(n, \chi) = \mathsf{DLWE}(n, 2, \chi)$ and $\mathsf{DLPN}(n, m, \boldsymbol{\chi}) = \mathsf{DLWE}(n, m, 2, \boldsymbol{\chi})$.

Whenever the error distribution $\boldsymbol{\chi}$ is sufficiently bounded (which will be the case for all error distributions that we explicitly consider) and the distributions $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$ are not statistically close (which is the case if the criteria in Lemma 3.2 are met), then the hardness of the decisional LWE problem implies the hardness of the LWE search problem. This can be seen as follows. Given $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, an algorithm $\mathcal{A}$ solving the LWE search problem can be used to find $\mathbf{s}$ and we can test whether $\mathbf{e} = \mathbf{y} - \mathbf{As}$ is short. On input $(\mathbf{A}, \mathbf{u})$, $\mathcal{A}$ cannot find such an $\mathbf{s}$, as with high probability $\mathbf{u}$ is far away from the code/lattice generated by $\mathbf{A}$ and consequently no such $\mathbf{s}$ exists. Thus, we can distinguish the distributions $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$ using $\mathcal{A}$, contradicting the hardness of decisional LWE.

For the learning parity with noise problem, a series of works [BFKL93, HB01, KS06, KSS10, AIK07] have shown that the converse is also true, i.e. the hardness of LPN implies the hardness of DLPN. While usually such reductions require a large amount of samples, Applebaum et al. [AIK07] provide a sample preserving reduction, i.e. they show that the hardness of $\mathsf{LPN}(m, n, \boldsymbol{\chi})$ implies the hardness of $\mathsf{DLPN}(m, n, \boldsymbol{\chi})$. Since this reduction is elementary, we will provide it here.

Lemma 3.3 (Applebaum, Ishai and Kushilevitz [AIK07])**.** *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$ and $\boldsymbol{\chi}$ be an error distribution defined on $\mathbb{F}_2^m$. The problem $\mathsf{DLPN}(m, n, \boldsymbol{\chi})$ is hard, given that $\mathsf{LPN}(m, n, \boldsymbol{\chi})$ is hard.*

Proof. Assume towards contradiction that there exists a PPT-algorithm $\mathcal{D}$ that distinguishes $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$ with non-negligible advantage ϵ. Assume *wlog* that

$$\Pr[\mathcal{D}(\mathbf{A}, \mathbf{u}) = 1] - \Pr[\mathcal{D}(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] \geq \epsilon.$$

We will construct an algorithm $\mathcal{A}'$ that computes the Goldreich-Levin hardcore-bit $\langle \mathbf{r}, \mathbf{s} \rangle$ of $\mathbf{s}$, given $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ with advantage $\epsilon/2$.

By the Goldreich Levin theorem (Theorem 2.2) this algorithm $\mathcal{A}'$ can be used to construct an algorithm $\mathcal{A}''$ that solves the LPN search problem with non-negligible advantage, contradicting the hardness of $\mathsf{LPN}(m, n, \boldsymbol{\chi})$. $\mathcal{A}'$ is given as follows.

Adversary $\mathcal{A}'$
Input: $(\mathbf{A}, \mathbf{y}) \in \mathbb{F}_2^{m \times n} \times \mathbb{F}_2^m$ and $\mathbf{r} \in \mathbb{F}_2^n$
$\mathbf{u} \leftarrow_\$ \mathbb{F}_2^m$
$\mathbf{B} \leftarrow \mathbf{A} - \mathbf{u} \cdot \mathbf{r}^T$
$b \leftarrow \mathcal{D}(\mathbf{B}, \mathbf{y})$
return b

We will now analyze the success probability of $\mathcal{A}'$. Let $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ and $\mathbf{r}$ be $\mathcal{A}'$'s input. First notice that since $\mathbf{A}$ is distributed uniformly at random, so is $\mathbf{B} = \mathbf{A} - \mathbf{ur}^T$, as the uniform distribution is shift invariant. Moreover, since $\mathbf{A} = \mathbf{B} + \mathbf{u} \cdot \mathbf{r}^T$, it holds that $\mathbf{y} = \mathbf{As} + \mathbf{e} = \mathbf{Bs} + \mathbf{ur}^T\mathbf{s} + \mathbf{e} = \mathbf{Bs} + \mathbf{u}\langle \mathbf{r}, \mathbf{s} \rangle + \mathbf{e}$. On one hand, if $\langle \mathbf{r}, \mathbf{s} \rangle = 0$, then $(\mathbf{B}, \mathbf{y})$ has the distribution $(\mathbf{B}, \mathbf{Bs} + \mathbf{e})$. On the other hand, if $\langle \mathbf{r}, \mathbf{s} \rangle = 1$, then $\mathbf{y} = \mathbf{Bs} + \mathbf{e} + \mathbf{u}$. As $\mathbf{u}$ is uniformly distributed (independently of $\mathbf{B}$, $\mathbf{s}$ and $\mathbf{e}$), $\mathbf{y}$ is also uniformly distributed. Thus, $(\mathbf{B}, \mathbf{y})$ has the distribution $(\mathbf{B}, \mathbf{u}')$, for uniformly chosen $\mathbf{u}'$. We conclude that

$$\begin{aligned}
\mathsf{Succ}(\mathcal{A}') &= \Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}), \mathbf{r}) = \langle \mathbf{r}, \mathbf{s} \rangle] \\
&= \frac{1}{2} \Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}), \mathbf{r}) = 0 | \langle \mathbf{r}, \mathbf{s} \rangle = 0] \\
&\quad + \frac{1}{2} \Pr[\mathcal{A}'((\mathbf{A}, \mathbf{As} + \mathbf{e}), \mathbf{r}) = 1 | \langle \mathbf{r}, \mathbf{s} \rangle = 1] \\
&= \frac{1}{2} \Pr[\mathcal{D}(\mathbf{B}, \mathbf{Bs} + \mathbf{e}) = 0] + \frac{1}{2} \Pr[\mathcal{D}(\mathbf{B}, \mathbf{u}') = 1] \\
&= \frac{1}{2} + \frac{1}{2} \cdot (\Pr[\mathcal{D}(\mathbf{B}, \mathbf{u}') = 1] - \Pr[\mathcal{D}(\mathbf{B}, \mathbf{Bs} + \mathbf{e}) = 1]) \\
&\geq \frac{1}{2} + \frac{\epsilon}{2},
\end{aligned}$$

i.e. $\mathcal{A}'$ computes the Goldreich-Levin hardcore bit $\langle \mathbf{r}, \mathbf{s} \rangle$ with probability non-negligibly better than $\frac{1}{2}$, contradicting the hardness of $\mathsf{LPN}(m, n, \boldsymbol{\chi})$. □

For the LWE problem, there exist search-to-decision reductions that apply to any error distribution and others that only apply to specific error distributions. The first search-to-decision reduction for LWE, provided by Regev [Reg05], works for any error distribution, but is rather wasteful regarding the number of samples required. Specifically, to establish the hardness of the decisional problem $\mathsf{DLWE}(n, m, q, \chi)$, the reduction provided in [Reg05] needs to assume the hardness of

$\mathsf{LWE}(n, \mathsf{poly}(n,m,q), q, \chi)$, i.e. there is a polynomial loss in the number of samples. Micciancio and Mol [MM11] provided a sample preserving search-to-decision reduction, i.e. the hardness of $\mathsf{DLWE}(n,m,q,\chi)$ can be established from the hardness of $\mathsf{LWE}(n,m,q,\chi)$. This is a direct LWE-analogue of Lemma 3.3.

Theorem 3.1 (Micciancio and Mol [MM11])**.** *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$, $q = \mathsf{poly}(\lambda)$ be a prime modulus and let $\boldsymbol{\chi}$ be* any *distribution over $\mathbb{Z}_q^m$. Assume there exists a PPT-distinguisher $\mathcal{D}$ that distinguishes $\mathsf{DLWE}(n,m,q,\boldsymbol{\chi})$ with non-negligible advantage, then there exists a PPT-adversary $\mathcal{A}$ that inverts $\mathsf{LWE}(n,m,q,\boldsymbol{\chi})$ with non-negligible success-probability.*

Another line of works considered only search-to-decision reductions for LWE with gaussian errors. The first such reduction was given by Peikert [Pei09]. While Peikert's reduction only applied to LWE with moduli q that are products of distinct small primes, this has since been generalized to a wider class of moduli [ACPS09, MP12]. We will state the most general such reduction due to Micciancio and Peikert [MP12]

Theorem 3.2 (Micciancio and Peikert [MP12])**.** *Let λ be a security parameter. Let $n = \mathsf{poly}(\lambda)$, $q = p_1^{e_1} \cdot \cdots \cdot p_k^{e_k}$ for pairwise distinct $\mathsf{poly}(\lambda)$ bounded primes p_i with each $e_1 \geq 1$. Let $0 < \alpha < 1/\omega(\sqrt{\log(n)})$. Let ℓ be the number of prime factors $p_i < \omega(\sqrt{n})/\alpha$. Let $\alpha' \geq \alpha$ be such that $\alpha' \geq \omega(\sqrt{\log(n)})/p_i^{e_i}$ for every i and $(\alpha')^\ell \geq \alpha\omega(\sqrt{\log(n)})^{\ell+1}$. Assume there exists a PPT-distinguisher $\mathcal{D}$ against $\mathsf{DLWE}(n,q,D_{\alpha' q})$, then there exists an efficient algorithm solving $\mathsf{DLWE}(n,q,D_{\alpha q})$.*

In particular, Theorem 3.2 allows to establish the hardness of the decisional LWE problem for moduli that are powers of 2.

3.4. Variants of LPN and LWE

We will now introduce a several variants of LPN and LWE that will be useful in the construction of our public key encryption schemes in Chapter 6.

3.4.1. Matrix Version

We first introduce a matrix versions of both the LWE and LPN problems. Roughly speaking, an instance of the matrix version consists of several LPN instances that use the same matrix $\mathbf{A}$, but use independent vectors $\mathbf{x}_i$ and $\mathbf{e}_i$. We will directly state the decisional version of the problem.

Problem 3.3 (Decisional Matrix LPN and LWE)**.** *Let λ be a security parameter. Let $m, n, k = \mathsf{poly}(\lambda)$ be positive integers, let $q \geq 2$ be a modulus and let $\boldsymbol{\chi}$ be an error distribution on $\mathbb{Z}_q^m$. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be chosen uniformly at random, $\mathbf{X}$ in $\mathbb{Z}_q^{n\times k}$ be chosen uniformly at random and $\mathbf{E} \in \mathbb{Z}_q^{m\times k}$ be chosen according to $\boldsymbol{\chi}^k$, i.e. each column $\mathbf{e}_i$ of $\mathbf{E}$ is chosen independently from $\boldsymbol{\chi}$. Let $\mathbf{U}$ be distributed uniformly on $\mathbb{Z}_q^{m\times k}$. The decisional matrix LWE problem $\mathsf{DMLWE}(n,m,k,q,\boldsymbol{\chi})$ is to distinguish the distributions $(\mathbf{A}, \mathbf{AX}+\mathbf{E})$ and $(\mathbf{A},\mathbf{U})$. For the modulus $q = 2$, the decisional matrix LPN problem is defined as $\mathsf{DMLPN}(n,m,k,\boldsymbol{\chi}) = \mathsf{DMLWE}(n,m,k,2,\boldsymbol{\chi})$.*

We will now show that $\mathsf{DMLWE}(n,m,k,q,\boldsymbol{\chi})$ is at least as hard as $\mathsf{DLWE}(n,m,q,\boldsymbol{\chi})$. This follows by a simple hybrid argument.

Lemma 3.4. *Let λ be a security parameter. Let $m, n, k = \mathsf{poly}(\lambda)$ be positive integers, let $q \geq 2$ be a modulus and let $\boldsymbol{\chi}$ be an error distribution on $\mathbb{Z}_q^m$. Assume there exists a PPT distinguisher $\mathcal{D}$ against $\mathsf{DMLWE}(n, m, k, q, \boldsymbol{\chi})$ with non-negligible advantage ϵ. Then there exists a PPT distinguisher $\mathcal{D}'$ that has has non-negligible advantage ϵ/k against $\mathsf{DLWE}(n, m, q, \boldsymbol{\chi})$.*

Proof. We will first provide a description of the distinguisher $\mathcal{D}'$.

Distinguisher $\mathcal{D}'$
Input: An Instance $(\mathbf{A}, \mathbf{y})$ of $\mathsf{DLWE}(n, m, q, \boldsymbol{\chi})$
$i \leftarrow_\$ \{1, \ldots, k\}$
For $j = 1, \ldots, i-1$
  $\mathbf{y}_j \leftarrow_\$ \mathbb{Z}_q^m$
$\mathbf{y}_i \leftarrow \mathbf{y}$
For $j = i+1, \ldots, k$
  $\mathbf{x}_j \leftarrow_\$ \mathbb{Z}_q^n$
  $\mathbf{e}_j \leftarrow_\$ \boldsymbol{\chi}$
  $\mathbf{y}_j \leftarrow \mathbf{A}\mathbf{x}_i + \mathbf{e}_i$
$\mathbf{Y} \leftarrow (\mathbf{y}_1 \| \ldots \| \mathbf{y}_k)$
$b \leftarrow \mathcal{D}(\mathbf{A}, \mathbf{Y})$
return b

We will now analyze the distinguishing advantage of $\mathcal{D}'$. By assumption, it holds that

$$\mathsf{Adv}(\mathcal{D}) = |\Pr[\mathcal{D}(\mathbf{A}, \mathbf{AX} + \mathbf{E}) = 1] - \Pr[\mathcal{D}(\mathbf{A}, \mathbf{U}) = 1]| \geq \epsilon.$$

For a fixed $t \in \{0, \ldots, k\}$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ define the random variable $\mathbf{Y}^{(t)} \in \mathbb{Z}_q^{m \times k}$ as follows. The first t columns $\mathbf{y}_1, \ldots, \mathbf{y}_t$ of $\mathbf{Y}^{(t)}$ are chosen uniformly at random, while the remaining $k-t$ columns $\mathbf{y}_{t+1}, \ldots, \mathbf{y}_k$ are computed by $\mathbf{y}_j \leftarrow \mathbf{A}\mathbf{s}_j + \mathbf{e}_j$, where the $\mathbf{s}_j \leftarrow_\$ \mathbb{Z}_q^n$ are chosen uniformly at random and $\mathbf{e}_j \leftarrow_\$ \boldsymbol{\chi}$ are chosen according to $\boldsymbol{\chi}$.

We will first analyze what happens when $\mathcal{D}'$'s input is distributed by $(\mathbf{A}, \mathbf{As} + \mathbf{e})$. Fix a $t \in \{1, \ldots, k\}$. Then, conditioned to $i = t$, the matrix $\mathbf{Y}$ assembled by $\mathcal{D}'$ is distributed according identical to $\mathbf{Y}^{(t-1)}$. Thus it holds that

$$\Pr[\mathcal{D}'(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1 | i = t] = \Pr[\mathcal{D}(\mathbf{A}, \mathbf{Y}^{(t-1)}) = 1].$$

As i is chosen uniformly at random from $\{1, \ldots, k\}$, it holds that

$$\begin{aligned}\Pr[\mathcal{D}'(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] &= \sum_{t=1}^{k} \Pr[i = t] \Pr[\mathcal{D}'(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1 | i = t] \\ &= \sum_{t=1}^{k} \frac{1}{k} \Pr[\mathcal{D}(\mathbf{A}, \mathbf{Y}^{(t-1)}) = 1].\end{aligned}$$

Now assume that $\mathcal{D}'$'s input is distributed by $(\mathbf{A}, \mathbf{u})$ for a uniformly random $\mathbf{u} \in \mathbb{Z}_q^m$. Again fix a $t \in \{1, \ldots, k\}$. Then, conditioned to $i = t$, the matrix $\mathbf{Y}$ assembled by $\mathcal{D}'$ is distributed according identical to $\mathbf{Y}^{(t)}$. Thus we have

$$\Pr[\mathcal{D}'(\mathbf{A}, \mathbf{u} | i = t] = \Pr[\mathcal{D}(\mathbf{A}, \mathbf{Y}^{(t)}) = 1]$$

and

$$\begin{aligned}\Pr[\mathcal{D}'(\mathbf{A},\mathbf{u})=1] &= \sum_{t=1}^{k}\Pr[i=t]\Pr[\mathcal{D}'(\mathbf{A},\mathbf{u})=1|i=t]\\ &= \sum_{t=1}^{k}\frac{1}{k}\Pr[\mathcal{D}(\mathbf{A},\mathbf{Y}^{(t)})=1].\end{aligned}$$

Putting all together yields

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}') &= |\Pr[\mathcal{D}'(\mathbf{A},\mathbf{As}+\mathbf{e})=1]-\Pr[\mathcal{D}'(\mathbf{A},\mathbf{u})=1]|\\ &= \left|\sum_{t=1}^{k}\frac{1}{k}\Pr[\mathcal{D}(\mathbf{A},\mathbf{Y}^{(t-1)})=1]-\sum_{t=1}^{k}\frac{1}{k}\Pr[\mathcal{D}(\mathbf{A},\mathbf{Y}^{(t)})=1]\right|\\ &= \frac{1}{k}|\Pr[\mathcal{D}(\mathbf{A},\mathbf{Y}^{(0)})=1]-\Pr[\mathcal{D}(\mathbf{A},\mathbf{Y}^{(k)})=1]|\\ &= \frac{1}{k}|\Pr[\mathcal{D}(\mathbf{A},\mathbf{AX}+\mathbf{E})=1]-\Pr[\mathcal{D}(\mathbf{A},\mathbf{U})=1]|\\ &= \frac{1}{k}\mathsf{Adv}(\mathcal{D})\geq\frac{\epsilon}{k}.\end{aligned}$$

Thus, $\mathcal{D}'$ has advantage ϵ/k against $\mathsf{DLWE}(n,m,q,\boldsymbol{\chi})$, which concludes the proof. □

3.4.2. Dual Matrix Version

From a coding theoretic view point, in the formulation of Problems 3.1 and 3.2 the task is, given a generator matrix and a noisy codeword, to decode the noisy codeword or distinguish the noisy codeword from uniform random respectively. We will now solely focus on the LPN case. We will provide a variant of the LPN problem that reformulates the problem in dual or *parity check* terms. In this formulation one is given a parity check matrix and a syndrome and has to find a short error corresponding to this syndrome or distinguish from random respectively. We will only formulate this problem for the decisional version of the problem, as the hardness of the search version follows instantly. Sometimes this version is also referred to a knapsack version in literature [ACPS09, KMP14]

Problem 3.4 (Decisional Dual Matrix LPN)**.** *Let λ be a security parameter. Let $k,l,m=\mathsf{poly}(\lambda)$ be positive integers with $l<m$ and let $\boldsymbol{\chi}$ be an error distribution defined on $\mathbb{F}_2^m$. Let $\mathbf{H}\in\mathbb{F}_2^{l\times m}$ be chosen uniformly at random and $\mathbf{E}\in\mathbb{F}_2^{m\times k}$ be chosen according to $\boldsymbol{\chi}^k$. Let $\mathbf{U}$ be distributed uniformly on $\mathbb{F}_2^{l\times k}$. The $\mathsf{DDMLPN}(l,m,k,\boldsymbol{\chi})$ problem is to distinguish the distributions $(\mathbf{H},\mathbf{HE})$ and $(\mathbf{H},\mathbf{U})$. For $k=1$ we define $\mathsf{DDLPN}(l,m,\boldsymbol{\chi})=\mathsf{DDMLPN}(l,m,1,\boldsymbol{\chi})$*

We will now establish that the DDMLPN problem is at least as hard as the DMLPN problem, in fact the two problems are equivalent. This basically follows from that fact that if the generator matrix $\mathbf{A}\in\mathbb{F}_2^{m\times n}$ is chosen uniformly at random, then we can sample a dual matrix $\mathbf{H}\in\mathbb{F}_2^{(m-n)\times m}$ with $\mathbf{H}\cdot\mathbf{A}=0$ which has a marginal distribution statistically close to uniform.

Lemma 3.5. *Let λ be a security parameter. Let $k,m,n=\mathsf{poly}(\lambda)$ be positive integers with $m\geq n+\omega(\log(\lambda))$. Let $\boldsymbol{\chi}$ be an error distribution defined on $\mathbb{F}_2^m$. If there exists a PPT distinguisher $\mathcal{D}_1$ that distinguishes $\mathsf{DDMLPN}(m-n,m,k,\boldsymbol{\chi})$ with advantage ϵ, then there exists a distinguisher $\mathcal{D}_2$ that distinguishes $\mathsf{DMLPN}(n,m,k,\boldsymbol{\chi})$ with advantage $\epsilon-\mathsf{negl}(\lambda)$.*

Proof. Let $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m\times n}$ be chosen uniformly at random. By Lemma 2.6 the probability that the matrix $\mathbf{A}$ has full rank is at least $1 - 2^{n-m+1} \geq 1 - \mathsf{negl}(\lambda)$, as $m \geq n+\omega(\log(\lambda))$. Conditioned that $\mathbf{A}$ has full rank, $\mathsf{ker}(\mathbf{A}^T)$ is a uniformly chosen $m-n$ dimensional subspace of $\mathbb{F}_2^n$. Now let $\mathbf{H}^T$ be a random basis matrix of $\mathsf{ker}(\mathbf{A}^T)$. Then $\mathbf{H}$ is uniformly random among all full rank matrices in $\mathbb{F}_2^{(m-n)\times n}$. However, since (again by Lemma 2.6) a uniformly chosen matrix $\mathbf{H}' \in \mathbb{F}_2^{(m-n)\times m}$ is full rank with probability at least $1 - 2^{m-n-m+1} = 1 - 2^{-n+1} \geq 1 - \mathsf{negl}(\lambda)$, we get that $\mathbf{H}$ is statistically close to uniform in $\mathbb{F}_2^{(m-n)\times m}$. As $\mathbf{H}^T$ is a basis matrix of $\mathsf{ker}(\mathbf{A}^T)$, it holds that $\mathbf{A}^T \cdot \mathbf{H}^T = 0$ and thus $\mathbf{H}\cdot\mathbf{A} = 0$. Thus, $\mathbf{H}$ is a dual of $\mathbf{A}$ and its marginal distribution is statistically close to uniform. Observe that we can obtain a random $\mathbf{A}$ from $\mathbf{H}$ in the same way. We can now construct the distingisher $\mathcal{D}_2$.

Distinguisher $\mathcal{D}_2$
Input: $(\mathbf{A}, \mathbf{Y}) \in \mathbb{F}_2^{m\times n} \times \mathbb{F}_2^{m\times k}$
Choose a basis matrix $\mathbf{H}^T$ of $\mathsf{ker}(\mathbf{A})$ uniformly at random
$\mathbf{C} \leftarrow \mathbf{HY}$
$b \leftarrow \mathcal{D}_1(\mathbf{H}, \mathbf{C})$
return b

We will now analyze the distinguishing advantage of $\mathcal{D}_2$. By the above the matrix $\mathbf{H}$ is distributed statistically close to uniform from the view of $\mathcal{D}_2$. First assume that $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{AS} + \mathbf{E})$. Then it holds that

$$\mathbf{C} = \mathbf{HY} = \mathbf{H}(\mathbf{AS} + \mathbf{E}) = \mathbf{HAS} + \mathbf{HE} = \mathbf{HE}$$

Now assume that $(\mathbf{H}, \mathbf{y})$ is of the form $(\mathbf{H}, \mathbf{U})$ with uniformly random $\mathbf{U}$. Then $\mathbf{C} = \mathbf{HY} = \mathbf{HU}$ is also uniformly random, as $\mathbf{H}$ has full rank. It follows that

$$\begin{aligned}
\mathsf{Adv}(\mathcal{D}_2) &= |\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{AS}+\mathbf{E}) = 1] - \Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{U}) = 1]| \\
&\geq |\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{HE}) = 1] - \Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{U}) = 1]| - \mathsf{negl}(\lambda) \\
&= \mathsf{Adv}(\mathcal{D}_2) - \mathsf{negl}(\lambda) \\
&\geq \epsilon - \mathsf{negl}(\lambda).
\end{aligned}$$

Thus, $\mathcal{D}_2$ has advantage at least $\epsilon - \mathsf{negl}(\lambda)$ against $\mathsf{DMLPN}(n, m, k, \boldsymbol{\chi})$, which concludes the proof. □

3.4.3. Fixed Weight Errors

Next, we will consider a variant of the LPN problem where the error vector $\mathbf{e}$ is chosen from a fixed weight distribution. In the original proposal of the McEliece cryptosystem [McE78] the error vector $\mathbf{e}$ is chosen uniformly with a fixed weight. We will now show that the hardness of the LPN search problem for Bernoulli distributed errors implies the hardness of LPN with fixed weight errors. More specifically, let c be a positive integer constant and let $\mathbf{m} = (m_1, \ldots, m_c)$ be such that $m = \sum_{i=1}^{c} m_i$. Define the set

$$M_{\mathbf{m},\rho} = \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor) \times \cdots \times \mathsf{S}_{m_c}(\lfloor \rho m_c \rfloor) \subseteq \mathbb{F}_2^m.$$

We show that LPN with errors chosen uniformly from $M_{\mathbf{m},\rho}$ is at least as hard as standard LPN with error distribution $\mathsf{Ber}(m, \rho)$. The hardness of the decisional problem $\mathsf{DLPN}(n, m, M_{\mathbf{m},\rho})$ then follows by Lemma 3.3. The idea behind Lemma 3.6

is as follows. We may think of m being partitioned in c buckets $m_1, \ldots, m_c$. Now, taking a Bernoulli distributed error $\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$ and partitioning $\mathbf{e}$ in c substrings yields c independently distributed vectors $\mathbf{e}_1, \ldots, \mathbf{e}_c$, where $\mathbf{e}_i$ is distributed according to $\mathsf{Ber}(m_1, \rho)$. Then, each $\mathbf{e}_i$ has weight ρm_i with substantial probability, and since c is constant all $\mathbf{e}_i$ are such that that $\mathsf{wgt}(\mathbf{e}_i) = \lfloor \rho m_i \rfloor$ simultaneously with substantial probability.

Lemma 3.6. *Let λ be a security parameter. Let $c > 0$ be a constant integer, let $m, n = \mathsf{poly}(\lambda)$ be positive integers, let $\rho = \rho(\lambda) \in (\frac{1}{\mathsf{poly}(\lambda)}, \frac{1}{2})$ and let $\mathbf{m} = (m_1, \ldots, m_c)$ be a partition of m, i.e. $\sum_{i=1}^{c} m_i = m$. Assume there exists an adversary $\mathcal{A}$ against $\mathsf{LPN}(n, m, M_{\mathbf{m},\rho})$ with non-negligible success probability ϵ. Then $\mathcal{A}$ has non-negligible success probability ϵ' against $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$.*

Proof. Let $\mathbf{e}$ be distributed according to $\mathsf{Ber}(m, \rho)$. We will show that

1. $\mathbf{e}$ is in $M_{\mathbf{m},\rho}$ with substantial probability $p \geq \frac{1}{\mathsf{poly}(\lambda)}$.
2. Conditioned to the event that $\mathbf{e}$ is in $M_{\mathbf{m},\rho}$, the conditional distribution of $\mathbf{e}$ is the uniform distribution on $M_{\mathbf{m},\rho}$.

Thus, with probability p a sample of $\mathsf{Ber}(m, \rho)$ is also a sample of the uniform distribution on $M_{\mathbf{m},\rho}$. Consequently an instance $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ of $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ is also an instance of $\mathsf{LPN}(n, m, M_{\mathbf{m},\rho})$ with probability p. Thus, any algorithm $\mathcal{A}$ with success probability ϵ against $\mathsf{LPN}(n, m, M_{\mathbf{m},\rho})$ must have non-negligible success probability $\epsilon' = p \cdot \epsilon$ against $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$. The main lever to prove the two assertions is that all components e_i of $\mathbf{e}$ are chosen independently. Let $\mathbf{e} = (\mathbf{e}_1^T \| \ldots \| \mathbf{e}_c^T)^T$ be distributed according to $\mathsf{Ber}(m, \rho)$. Then each $\mathbf{e}_i$ is distributed according to $\mathsf{Ber}(m_i, \rho)$. Assertion 2 holds because each $\mathbf{v} \in \mathbb{F}_2^{m_i}$ with $\mathsf{wgt}(\mathbf{v}) = \lfloor \rho m_i \rfloor$ the has probability of occurrence

$$\Pr[\mathbf{e}_i = \mathbf{v}] = \rho^{\lfloor \rho m_i \rfloor}(1-\rho)^{m_i - \lfloor \rho m_i \rfloor},$$

i.e. all $\mathbf{v} \in \mathbb{F}_2^{m_i}$ with $\mathsf{wgt}(\mathbf{v}) = \lfloor \rho m_i \rfloor$ have the same probability of occurrence. It follows that, conditioned to $\mathsf{wgt}(\mathbf{e}_i) = \lfloor \rho m_i \rfloor$, $\mathsf{Ber}(m, \rho)$ is the uniform distribution on $\mathsf{S}_{m_i}(\lfloor \rho m_i \rfloor)$. By the independence of the $\mathbf{e}_i$ assertion 2 follows. We will now turn to assertion 1. The hamming weight $\mathsf{wgt}(\mathbf{e}_i)$ is distributed according to $\mathsf{Bin}(m_i, \rho)$, i.e.

$$\Pr[\mathsf{wgt}(\mathbf{e}_i) = t] = \sum_{j=0}^{t} \binom{m_i}{j} \rho^j (1-\rho)^{m_i - j}.$$

It holds that

$$\begin{aligned}
\Pr[\mathsf{wgt}(\mathbf{e}_i) = \lfloor \rho m_i \rfloor] &\geq \binom{m_i}{\lfloor \rho m_i \rfloor} \rho^{\lfloor \rho m_i \rfloor}(1-\rho)^{m_i - \lfloor \rho m_i \rfloor} \\
&\geq \frac{2^{H(\rho) m_i}}{m_i + 1} \rho^{\lfloor \rho m_i \rfloor}(1-\rho)^{m_i - \lfloor \rho m_i \rfloor} \\
&\geq \frac{\rho(1-\rho)}{m_i + 1} 2^{H(\rho) m_i} \rho^{\rho m_i}(1-\rho)^{(1-\rho) m_i} \\
&= \frac{\rho(1-\rho)}{m_i + 1}.
\end{aligned}$$

The last equality holds as $2^{-H(\rho)m_i} = \rho^{\rho m_i}(1-\rho)^{(1-\rho)m_i}$. As the $\mathbf{e}_i$ are independently distributed, it holds that

$$\Pr[\forall i \in \{i, \ldots, c\} : \mathsf{wgt}(\mathbf{e}_i) = \lfloor \rho m_i \rfloor] = \prod_{i=1}^{c} \Pr[\mathsf{wgt}(\mathbf{e}_i) = \lfloor \rho m_i \rfloor] \geq \left(\frac{\rho(1-\rho)}{m+1}\right)^c.$$

This last expression is substantial in λ, as c is a constant and $\frac{1}{\mathsf{poly}(\lambda)} \leq \rho \leq \frac{1}{2}$. Thus assertion 1 also holds. This concludes the proof. □

3.4.4. Extended Dual Version

In order to establish the hardness of LPN with unbounded samples from LPN with bounded samples, we will use the following intermediate problem called extended decisional dual LPN (EDDLPN). The definition is a direct analogue of the extended LWE problem defined in [ASP12, KMP14]. In the EDDLPN problem, the adversary obtains an extra *advice* $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ about the error vector $\mathbf{e}$, where $\mathbf{z}$ is chosen from a low weight distribution (rather than uniform).

Problem 3.5 (Extended Decisional Dual LPN)**.** *Let λ be a security parameter. Let $l, m = \mathsf{poly}(\lambda)$ be positive integers with $l < m$ and let $\boldsymbol{\chi}$, $\boldsymbol{\psi}$ be distributions defined on $\mathbb{F}_2^m$. Let $\mathbf{H} \in \mathbb{F}_2^{k \times m}$ be chosen uniformly at random, $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$ and $\mathbf{z}$ be chosen according to $\boldsymbol{\psi}$. Let $\mathbf{u}$ be distributed uniformly on $\mathbb{F}_2^m$. The goal of the extended decisional dual LPN problem $\mathsf{EDDLPN}(k, m, \boldsymbol{\chi}, \boldsymbol{\psi})$ is to distinguish the distributions $(\mathbf{H}, \mathbf{He}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ and $(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$.*

The following Lemma, which is a direct adaption of Theorem 3.1 from [ASP12], establishes that EDDLPN is at least as hard as DDLPN.

Lemma 3.7. *Assume there exists a distinguisher $\mathcal{D}$ that distinguishes the problem $\mathsf{EDDLPN}(k, m, \mathsf{Ber}(m, \rho), \mathsf{S}_m(\lfloor \rho m \rfloor))$ with non-negligible advantage ϵ. Then there exists a distinguisher $\mathcal{D}'$ against $\mathsf{DDLPN}(k, m, \mathsf{Ber}(m, \rho))$ with advantage $\epsilon/2$.*

The proof of Lemma is identical to the proof given in [ASP12], except for a small detail. We provide it here for the sake of completeness.

Proof. We will first provide the description of the distinguisher $\mathcal{D}'$.

Distinguisher $\mathcal{D}'$
Input: $\mathsf{DDLPN}(k, m, \mathsf{Ber}(m, \rho))$ Instance $(\mathbf{H}, \mathbf{y})$
$\mathbf{z} \leftarrow_\$ \mathsf{S}_m(\lfloor \rho m \rfloor)$
$\mathbf{e}' \leftarrow_\$ \boldsymbol{\chi}$
$\mathbf{v} \leftarrow_\$ \mathbb{F}_2^m$
$\mathbf{H}' \leftarrow \mathbf{H} - \mathbf{v}\mathbf{z}^T$
$\mathbf{y}' \leftarrow \mathbf{y} - \mathbf{v}\langle \mathbf{z}, \mathbf{e}' \rangle$
$b \leftarrow \mathcal{D}(\mathbf{H}', \mathbf{y}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle)$
return b

We will now analyze the advantage of $\mathcal{D}'$. Assume first the $\mathcal{D}'$'s input is of the form $(\mathbf{H}, \mathbf{u})$ with uniformly random $\mathbf{u}$. Then clearly $\mathbf{H}' = \mathbf{H} - \mathbf{v}\mathbf{z}^T$, $\mathbf{y}' = \mathbf{u} - \langle \mathbf{z}, \mathbf{e}' \rangle \cdot \mathbf{v} =: u'$ are also uniformly random, independent of $\mathbf{z}$ and $\mathbf{e}'$. Thus it holds that

$$\Pr[\mathcal{D}'(\mathbf{H}, \mathbf{u}) = 1] = \Pr[\mathcal{D}(\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = 1].$$

Now assume that $\mathcal{D}'$'s input is of the form $(\mathbf{H}, \mathbf{He})$. Then $\mathbf{H}' = \mathbf{H} - \mathbf{vz}^T$ is independently uniformly distributed. Moreover, it holds that

$$\mathbf{y}' = \mathbf{y} - \mathbf{v}\langle \mathbf{z}, \mathbf{e}' \rangle = \mathbf{He} - \mathbf{v}\langle \mathbf{z}, \mathbf{e}' \rangle = \mathbf{H}'\mathbf{e} + \mathbf{v}\langle \mathbf{z}, \mathbf{e} - \mathbf{e}' \rangle.$$

Consequently, if $\langle \mathbf{z}, \mathbf{e} - \mathbf{e}' \rangle = 0$, which happens with probability $\geq \frac{1}{2}$, it immediately follows that $\langle \mathbf{z}, \mathbf{e} \rangle = \langle \mathbf{z}, \mathbf{e}' \rangle$ and thus

$$(\mathbf{H}', \mathbf{y}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = (\mathbf{H}', \mathbf{H}'\mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle).$$

On the other hand, if $\langle \mathbf{z}, \mathbf{e} - \mathbf{e}' \rangle = 1$, then $\mathbf{y}' = \mathbf{y} - \mathbf{v} =: \mathbf{u}'$ is uniformly random, as $\mathbf{v}$ is uniformly random. Thus, in this case

$$(\mathbf{H}', \mathbf{y}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = (\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle).$$

Let $\alpha = \Pr[\langle \mathbf{z}, \mathbf{e} - \mathbf{e}' \rangle = 0]$. Then it holds that

$$\Pr[\mathcal{D}'(\mathbf{H}, \mathbf{He}) = 1] = \alpha \Pr[\mathcal{D}(\mathbf{H}', \mathbf{H}'\mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] + (1-\alpha) \Pr[\mathcal{D}(\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = 1].$$

All together, this yields that

$$\begin{aligned}
\mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}') &= |\Pr[\mathcal{D}'(\mathbf{H}, \mathbf{He}) = 1] - \Pr[\mathcal{D}'(\mathbf{H}, \mathbf{u}) = 1]| \\
&= |\alpha \Pr[\mathcal{D}(\mathbf{H}', \mathbf{H}'\mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] \\
&\quad + (1-\alpha) \Pr[\mathcal{D}(\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = 1] \\
&\quad - \Pr[\mathcal{D}(\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = 1]| \\
&= \alpha |\Pr[\mathcal{D}(\mathbf{H}', \mathbf{H}'\mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] - \Pr[\mathcal{D}(\mathbf{H}', \mathbf{u}', \mathbf{z}, \langle \mathbf{z}, \mathbf{e}' \rangle) = 1]| \\
&= \alpha \mathsf{Adv}_{\mathsf{EDDLPN}}(\mathcal{D}) \geq \epsilon/2.
\end{aligned}$$

This concludes the proof. □

3.5. Getting More LPN Samples

In this Section we consider the following question:

> Assume that bounded samples LPN problem $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ is hard. Can we make any conclusions about the hardness of an unbounded sample LPN problem $\mathsf{LPN}(n, \mathsf{Ber}(\rho'))$ for some $\rho' \geq \rho$?

To the best of our knowledge, there are no prior results on this question in the LPN case. In the LWE case, a random self reduction using the generalized leftover hash lemma [DORS08] can be used to generate arbitrarily more samples from a given set of $m \approx n \log(q)$ samples, while the noise in the new samples rises only slightly. Specifically, if $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{z})$ is such a given set of *seed samples*, then we can generate new samples by drawing $\mathbf{e} \in \mathbb{Z}_q^m$ from a discrete gaussian and setting $\mathbf{a}' = \mathbf{A}^T\mathbf{e}$ and $y' = \mathbf{e}^T\mathbf{y}$. Now it holds $\mathbf{a}' = \mathbf{A}^T\mathbf{e}$ and

$$y' = \mathbf{e}^T\mathbf{y} = \mathbf{e}^T\mathbf{As} + \mathbf{e}^T\mathbf{z} = \mathbf{a}'^T\mathbf{s} + \langle \mathbf{e}, \mathbf{z} \rangle.$$

The error term $\langle \mathbf{e}, \mathbf{z} \rangle$ follows a discrete gaussian distribution [Pei10, AGHS13]. Thus $(\mathbf{a}', y')$ is a proper LWE sample, as by the generalized leftover hash lemma [DORS08] it holds that $\mathbf{a}'$ is statistically close to uniform even given the additional information $\langle \mathbf{e}, \mathbf{z} \rangle$ about $\mathbf{e}$. However, this approach cannot be directly transferred to the

LPN setting. For the vector $\mathbf{a}' = \mathbf{A}^T\mathbf{e}$ to be statistically close to uniform, $\mathbf{e}$ must have min-entropy $\approx n$, and thus high weight. But this in turn means that $\langle \mathbf{e}, \mathbf{z} \rangle$ will only have a small bias. In Section 3.6.3, we discuss a LPN algorithm due to Lyubashevsky [Lyu05], which basically uses this approach to attack LPN. Unsurprisingly, the algorithm has super-polynomial complexity and this technique is not applicable for a reduction. However, in the last section we have seen the EDDLPN problem, which remains pseudorandom even if a distinguisher is given some extra advice $\langle \mathbf{e}, \mathbf{z} \rangle$, which is just what we need for this type of random self reduction. The following theorem can be considered a *pseudorandom self reduction*, as we effectively substitute the leftover hash lemma by a pseudorandom analogue.

Theorem 3.3. *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$ be positive integers and let $\rho' \geq \rho^2 m$. If $\mathsf{DLPN}(n, m, \mathsf{Ber}(m, \rho))$ are $\mathsf{DDLPN}(n+1, m, \mathsf{Ber}(m, \rho))$ hard, then $\mathsf{DLPN}(n, \mathsf{Ber}(\rho'))$ is also hard.*

Theorem 3.3 is basically a trade-off between noise and extra samples. We tolerate that the amount of noise required gets squared, while in turn we get an arbitrary polynomial amount of samples.

Proof. Let for simplicity ρ' be such that if $\mathbf{z}$ is any fixed vector of weight $\lfloor \rho m \rfloor$ and $\mathbf{e}$ is chosen according to $\mathsf{Ber}(m, \rho)$, then $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed according to $\mathsf{Ber}(\rho')$. Clearly, by Lemma 2.2 it holds that

$$\rho' = \frac{1}{2}(1 - (1-2\rho)^{\lfloor \rho m \rfloor}) \leq \rho \lfloor \rho m \rfloor \leq \rho^2 m.$$

Thus, if the statement of the Theorem holds for this particular ρ', it also holds for any bigger value of ρ' as we can just add the *noise difference* accordingly.

Let $\mathcal{D}$ be a PPT distinguisher against $\mathsf{DLPN}(n, \mathsf{Ber}(\rho'))$. We will provide a series of hybrid experiments $\mathsf{Exp}_1, \mathsf{Exp}_2, \mathsf{Exp}_3, \mathsf{Exp}_4$ and show that from the view of $\mathcal{D}$ any two of experiments the are indistinguishable. We will provide the experiments by defining the sample oracles $\mathcal{O}$ the distinguisher $\mathcal{D}$ gets access to.

Clearly, experiment Exp_1 provides samples from the LPN distribution while experiment Exp_4 provides uniformly random samples. Thus, we need to establish that from the view of $\mathcal{D}$ the experiments Exp_1 and Exp_4 are indistinguishable. We will start with the indistinguishability of Exp_1 and Exp_2. Assume towards contradiction that $\mathcal{D}$ distinguishes with non-negligible advantage ϵ_1 between Exp_1 and Exp_2, i.e.

$$|\Pr[\mathsf{Exp}_1(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_2(\mathcal{D}) = 1]| \geq \epsilon_1.$$

Assume further that $k = \mathsf{poly}(\lambda)$ is an upper bound on the number of samples $\mathcal{D}$ queries. We will construct a PPT distinguisher $\mathcal{D}_1$ that distinguishes the problem $\mathsf{EDDLPN}(n, m, \mathsf{Ber}(m, \rho), \mathsf{S}_m(\lfloor \rho m \rfloor))$ with advantage $\geq \epsilon/k$. We will now provide $\mathcal{D}_1$.

Notice that $\mathcal{D}_1$ answers the first $i-1$ oracle queries of $\mathcal{D}$ exactly like Exp_1, while it answers the last $k-i-1$ queries like Exp_2. In the i-th query however, $\mathcal{D}_1$ embeds its own challenge. Moreover, notice that $\mathcal{D}_1$ is efficient as $\mathcal{D}$ is efficient. To analyze the distinguishing advantage of $\mathcal{D}_1$, we will define a sequence of hybrid experiments $\mathsf{H}_0, \ldots, \mathsf{H}_k$. H_i is crafted to answer the first i queries like Exp_1, while it answers the last $k-i$ queries like Exp_2.

We are now ready to analyze the distinguishing advantage of $\mathcal{D}_1$. First assume that $\mathcal{D}_1$'s input is of the form $(\mathbf{H}, \mathbf{He}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$. Fix the random choice $i^* = i$. Then

Experiment Exp_1
Initialization:
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
Sample Oracle $\mathcal{O}_{\mathsf{Exp}_1}()$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$e \leftarrow_\$ \mathsf{Ber}(\rho')$
$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$
Return $(\mathbf{a}, y)$

Experiment Exp_2
Initialization:
$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{z} \leftarrow_\$ \mathsf{S}_m(\lfloor \rho m \rfloor)$
$\mathbf{r} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$
Sample Oracle $\mathcal{O}_{\mathsf{Exp}_2}()$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T\mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
Return $(\mathbf{a}, y)$

Experiment Exp_3
Initialization:
$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$\mathbf{r} \leftarrow_\$ \mathbb{F}_2^m$
Sample Oracle $\mathcal{O}_{\mathsf{Exp}_3}()$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T\mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
Return $(\mathbf{a}, y)$

Experiment Exp_4
Initialization:
-
Sample Oracle $\mathcal{O}_{\mathsf{Exp}_4}()$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$y \leftarrow_\$ \mathbb{F}_2$
Return $(\mathbf{a}, y)$

the sample oracle $\mathcal{O}_{\mathcal{D}_1}()$ implemented by $\mathcal{D}_1$ behaves identical to the sample oracle of H_{i-1}. Consequently, it holds that

$$\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{He}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1 | i^* = i] = \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1]$$

and thus, as i^* is uniformly chosen from $\{1, \ldots, k\}$

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{He}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{He}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1].\end{aligned}$$

Next assume that $\mathcal{D}_1$'s input is of the form $(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$. Again, fix the random choice of i^* to $i^* = i$. Then the sample oracle $\mathcal{O}_{\mathcal{D}_1}()$ implemented by $\mathcal{D}_1$ behaves identical to the sample oracle of H_i, as $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed according to $\mathsf{Ber}(\rho')$. Consequently,

$$\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1 | i^* = i] = \Pr[\mathsf{H}_i(\mathcal{D}) = 1]$$

and thus

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}_i(\mathcal{D}) = 1].\end{aligned}$$

Distinguisher $\mathcal{D}_1$
Input: $(\mathbf{H}, \mathbf{c}, \mathbf{z}, t)$
$i^* \leftarrow_\$ \{1, \ldots, k\}$
$\mathbf{A} \leftarrow \mathbf{H}^T$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{r} = \mathbf{A}\mathbf{s} + \mathbf{z}$
$cnt = 1$
$b \leftarrow \mathcal{D}^{\mathcal{O}_{\mathcal{D}_1}()}$
return b

Sample Oracle $\mathcal{O}_{\mathcal{D}_1}()$
If $cnt < i^*$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$e \leftarrow_\$ \mathsf{Ber}(\rho')$
$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$
If $cnt = i^*$
$\mathbf{a} \leftarrow \mathbf{c}$
$y \leftarrow \langle \mathbf{c}, \mathbf{s} \rangle + t$
If $cnt > i^*$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T\mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Experiment H_i
Initialization:
$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{z} \leftarrow_\$ \mathsf{S}_m(\lfloor \rho m \rfloor)$
$\mathbf{r} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$
$cnt \leftarrow 1$

Sample Oracle $\mathcal{O}_{\mathsf{H}_i}()$
If $cnt \leq i$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$
$e \leftarrow_\$ \mathsf{Ber}(\rho')$
$y \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e$
If $cnt > i$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T\mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Together, this yields

$$\begin{aligned}
\mathsf{Adv}_{\mathsf{EDDLPN}}(\mathcal{D}_1) &= |\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{H}\mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1] - \Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) = 1]| \\
&= \left| \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}_{i-1}(\mathcal{D}) = 1] - \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}_i(\mathcal{D}) = 1] \right| \\
&= \frac{1}{k} |\Pr[\mathsf{H}_0(\mathcal{D}) = 1] - \Pr[\mathsf{H}_k(\mathcal{D}) = 1]| \\
&= \frac{1}{k} |\Pr[\mathsf{Exp}_2(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_1(\mathcal{D}) = 1]| \\
&\geq \epsilon_1 / k.
\end{aligned}$$

Thus, $\mathcal{D}_1$ distinguishes $\mathsf{EDDLPN}(n, m, \mathsf{Ber}(m, \rho), \mathsf{S}_m(\lfloor \rho m \rfloor))$ with non-negligible advantage ϵ_1/k, contradicting the hardness of $\mathsf{EDDLPN}(n, m, \mathsf{Ber}(m, \rho), \mathsf{S}_m(\lfloor \rho m \rfloor))$. By Lemma 3.7, this contradicts the hardness of $\mathsf{DDLPN}(n, m, \mathsf{Ber}(m, \rho))$.

Next, we turn to the indistinguishability of Exp_2 and Exp_3. Assume towards contradiction that $\mathcal{D}$ distinguishes between Exp_2 and Exp_3 with non-negligible advantage

ϵ_2, i.e.

$$|\Pr[\mathsf{Exp}_2(\mathcal{D})=1]-\Pr[\mathsf{Exp}_3(\mathcal{D})=1]|\geq\epsilon_2.$$

We will construct a PPT distinguisher $\mathcal{D}_2$ against $\mathsf{DLPN}(n,m,\mathsf{S}_m(\lfloor\rho m\rfloor))$. $\mathcal{D}_2$ is given as follows

Distinguisher $\mathcal{D}_2$
Input: $(\mathbf{A},\mathbf{r})$
$b\leftarrow\mathcal{D}^{\mathcal{O}_{\mathcal{D}_2}()}$
return b

Sample Oracle $\mathcal{O}_{\mathcal{D}_2}()$
$\mathbf{e}\leftarrow_{\$}\mathsf{Ber}(m,\rho)$
$\mathbf{a}\leftarrow\mathbf{e}^T\mathbf{A}$
$y\leftarrow\langle\mathbf{e},\mathbf{r}\rangle$
Return $(\mathbf{a},y)$

The distinguisher $\mathcal{D}_2$ is efficient, as $\mathcal{D}$ is efficient. First, assume that $\mathcal{D}_2$'s input is of the form $(\mathbf{A},\mathbf{As}+\mathbf{z})$, where $\mathbf{s}$ is chosen uniformly from $\mathbb{F}_2^n$ and $\mathbf{z}$ is chosen uniformly from $\mathsf{S}_m(\lfloor\rho m\rfloor)$. Then clearly the sample oracle $\mathcal{O}_{\mathcal{D}_2}$ behaves just as in Exp_2. On the other hand, if $\mathcal{D}_2$'s input is of the form $(\mathbf{A},\mathbf{u})$ with $\mathbf{u}$ chosen uniformly random from $\mathbb{F}_2^m$, then the sample $\mathcal{O}_{\mathcal{D}_2}$ simulated by $\mathcal{D}_2$ behaves like the sample oracle in Exp_3. Consequently, it holds that

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2)&=|\Pr[\mathcal{D}_2(\mathbf{A},\mathbf{As}+\mathbf{z})=1]-\Pr[\mathcal{D}_2(\mathbf{A},\mathbf{u})=1]|\\&=|\Pr[\mathsf{Exp}_2(\mathcal{D})=1]-\Pr[\mathsf{Exp}_3(\mathcal{D})=1]|\\&\geq\epsilon_2.\end{aligned}$$

Thus, the distinguishing advantage of $\mathcal{D}_2$ against $\mathsf{DLPN}(n,m,\mathsf{S}_m(\lfloor\rho m\rfloor))$ is at least ϵ_2, which by Lemma 3.6 contradicts the hardness of $\mathsf{DLPN}(n,m,\mathsf{Ber}(m,\rho))$.

We will finally turn to showing that from the view of $\mathcal{D}$, Exp_3 and Exp_4 are indistinguishable. Assume towards contradiction that $\mathcal{D}$ distinguishes between Exp_3 and Exp_4 with non-negligible advantage ϵ_3, i.e.

$$|\Pr[\mathsf{Exp}_3(\mathcal{D})=1]-\Pr[\mathsf{Exp}_4(\mathcal{D})=1]|\geq\epsilon_2.$$

Assume further $\mathcal{D}$ makes at most $k=\mathsf{poly}(\lambda)$ queries to its sample oracle. We will construct a PPT distinguisher $\mathcal{D}_3$ that distinguishes $\mathsf{DDLPN}(n+1,m,\mathsf{Ber}(m,\rho))$ with non-negligible advantage ϵ/k. The distinguisher $\mathcal{D}_3$ is given as follows.

We will first explain the first two operations of $\mathcal{D}_3$. Its input is $(\mathbf{H},\mathbf{c})\in\mathbb{F}_2^{(n+1)\times m}\times\mathbb{F}_2^{n+1}$. The matrix $\mathbf{H}^T$ is then split in its first n columns, which we call $\mathbf{A}\in\mathbb{F}_2^{m\times n}$, and its last column, which we call $\mathbf{r}\in\mathbb{F}_2^n$. We also split the vector $\mathbf{c}$ accordingly. Its first n elements form the vector $\mathbf{a}^*\in\mathbb{F}_2^n$, while its last element is the scalar $y^*\in\mathbb{F}_2$. The reason why this is done becomes evident in a moment. It is clear that $\mathcal{D}_3$ is efficient, once $\mathcal{D}$ is efficient.

Again, to analyze the distinguishing advantage of $\mathcal{D}_3$, we will define a sequence of hybrid experiments $\mathsf{H}'_0,\ldots,\mathsf{H}'_k$. H'_i is crafted to answer the first i queries like Exp_3, while it answers the last $k-i$ queries like Exp_4.

Clearly, in H'_i the first i queries to $\mathcal{O}_{\mathsf{H}'_i}()$ are answered like in Exp_3, while the remaining $k-i$ queries are answered like in Exp_4.

First assume that $\mathcal{D}_3$'s input in of the form $(\mathbf{H},\mathbf{He})$. Then it holds for the decomposed components $\mathbf{A},\mathbf{r},\mathbf{a}'$ and y' that

$$\begin{aligned}\mathbf{a}^*&=\mathbf{e}^T\mathbf{A}\\y^*&=\mathbf{e}^T\mathbf{r}=\langle\mathbf{e},\mathbf{r}\rangle\end{aligned}$$

Distinguisher $\mathcal{D}_3$
Input: $(\mathbf{H}, \mathbf{c})$
Parse $\mathbf{H}^T = (\mathbf{A} \| \mathbf{r})$
Parse $\mathbf{c}^T = (\mathbf{a}^{*T} \| y^*)^T$
$i^* \leftarrow_\$ \{1, \ldots, k\}$
$cnt = 1$
$b \leftarrow \mathcal{D}^{\mathcal{O}_{\mathcal{D}_3}()}$
return b

Sample Oracle $\mathcal{O}_{\mathcal{D}_3}()$
If $cnt < i^*$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T \mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
If $cnt = i^*$
$\mathbf{a} \leftarrow \mathbf{a}^*$
$y \leftarrow y^*$
If $cnt > i^*$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^m$
$y \leftarrow_\$ \mathbb{F}_2$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Experiment H'_i
Initialization:
$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$\mathbf{r} \leftarrow_\$ \mathbb{F}_2^m$
$cnt \leftarrow 1$

Sample Oracle $\mathcal{O}_{\mathsf{H}'_i}()$
If $cnt \leq i$
$\mathbf{e} \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{a} \leftarrow \mathbf{e}^T \mathbf{A}$
$y \leftarrow \langle \mathbf{e}, \mathbf{r} \rangle$
If $cnt > i$
$\mathbf{a} \leftarrow_\$ \mathbb{F}_2^m$
$y \leftarrow_\$ \mathbb{F}_2$
$cnt \leftarrow cnt + 1$
Return $(\mathbf{a}, y)$

Now fix a random choice $i^* = i$. Then $\mathcal{O}_{\mathcal{D}_3}()$ in $\mathcal{D}_3$'s simulation behaves identically to the sample oracle in H'_i. Thus it holds that

$$\Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{He}) = 1 | i^* = i] = \Pr[\mathsf{H}'_i(\mathcal{D}) = 1],$$

and consequently

$$\begin{aligned} \Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{He}) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{He}) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}'_i(\mathcal{D}) = 1]. \end{aligned}$$

Now suppose that $\mathcal{D}_3$'s input is of the form $(\mathbf{H}, \mathbf{u})$, where $\mathbf{u}$ is chosen uniformly at random from $\mathbb{F}_2^{n+1}$. Then it holds for the decomposed components $\mathbf{A}, \mathbf{r}, \mathbf{a}'$ and y' that

$$\begin{aligned} \mathbf{a}^* &= \mathbf{u}' \\ y^* &= u'' \end{aligned}$$

where $\mathbf{u}'$ is the vector of the first n components of $\mathbf{u}$ and u'' is the last component of $\mathbf{u}$. Again, fix a random choice $i' = i$. Then $\mathcal{O}_{\mathcal{D}_3}()$ in $\mathcal{D}_3$'s simulation behaves identically to the sample oracle in H'_{i-1}. Thus it holds that

$$\Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{u}) = 1 | i^* = i] = \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1],$$

and consequently

$$\begin{aligned}\Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{u}) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{u}) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1].\end{aligned}$$

Putting all together, we get

$$\begin{aligned}\mathsf{Adv}_{\mathsf{DDLPN}}(\mathcal{D}_3) &= |\Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{He}) = 1] - \Pr[\mathcal{D}_3(\mathbf{H}, \mathbf{u}) = 1]| \\ &= \left| \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}'_i(\mathcal{D}) = 1] - \sum_{i=1}^{k} \frac{1}{k} \cdot \Pr[\mathsf{H}'_{i-1}(\mathcal{D}) = 1] \right| \\ &= \frac{1}{k} |\Pr[\mathsf{H}'_k(\mathcal{D}) = 1] - \Pr[\mathsf{H}'_0(\mathcal{D}) = 1]| \\ &= \frac{1}{k} |\Pr[\mathsf{Exp}_3(\mathcal{D}) = 1] - \Pr[\mathsf{Exp}_4(\mathcal{D}) = 1]| \\ &\geq \epsilon_3 / k.\end{aligned}$$

Thus, $\mathcal{D}_3$ distinguishes $\mathsf{DDLPN}(n+1, m, \mathsf{Ber}(m, \rho))$ with non-negligible advantage ϵ_3/k, contradicting the hardness of $\mathsf{DDLPN}(n+1, m, \mathsf{Ber}(m, \rho))$ (Lemma 3.5).

□

3.6. Attacks and Assumed Hardness of LPN

We will now survey for which parameters the LPN problem is known to be easy, or conversely, for which parameters the LPN problem can be conjectured to be hard. In the last paragraph, we have seen that the central hub on which all LPN variants are based is the $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ problem. Thus, it suffices to investigate the hardness of $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$, as any attack on one of the variants implies an attack on this standard version of the problem. There are several (non-asymptotical) improvements to the attacks presented here discussed in literature [LF06, Kir11, BL12]. Moreover, Arora and Ge [AG11] provided an attack on LPN with structured noise, where the error terms obey a strong correlation. A variant of this algorithm is among the most efficient attacks against LWE with low noise (c.f. Section 3.8.2).

3.6.1. Brute Force Search

We will first examine the complexity of brute force search against the standard LPN problem $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$. Brute force search can either be mounted against the secret $\mathbf{s}$ or the error term $\mathbf{e}$. Let $(\mathbf{A}, \mathbf{y})$ be an instance of the problem $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$, i.e. $\mathbf{y} = \mathbf{As} + \mathbf{e}$ for a uniformly chosen $\mathbf{s}$ and an $\mathbf{e}$ distributed according to $\mathsf{Ber}(m, \rho)$. If we run a brute force search for the secret $\mathbf{s}$, then $2^{\Omega(n)}$ vectors $\mathbf{s}'$ need to be enumerated. For each $\mathbf{s}'$ we test if $\mathsf{wgt}(\mathbf{y} - \mathbf{As}) \approx \rho m$ and output $\mathbf{s}'$ if it meets this condition. Thus, the overall overhead for brute force search for the secret $\mathbf{s}$ is $2^{O(n)}$. Things are slightly different when we run a brute force

search for the error term. Due to concentration of mass, in this case the Chernoff Hoeffding bound (Theorem 2.1), an error term $\mathbf{e}$ chosen from $\mathsf{Ber}(m, \rho)$ has Hamming weight at most $O(\rho m)$, except with negligible probability. Therefore, we need to enumerate at most $2^{O(H(\rho)m)}$ error vectors $\mathbf{e}'$ and test whether there exists an $\mathbf{s}'$ such that $\mathbf{y} = \mathbf{A}\mathbf{s}' + \mathbf{e}'$. Thus, the overall complexity of brute force search for the error term is $2^{O(H(\rho)m)}$.

All together, the complexity of brute force search against $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \rho))$ is $2^{O(\min(n, H(\rho)m))}$. If ρ is a constant, then this becomes $2^{O(n)}$. However, if ρ is sub-constant, then $H(\rho)m$ can be significantly smaller than n. For instance if $\rho = O(n^{-\frac{1}{2}})$ and $m = O(n)$, then by Corollary 2.3 it holds that $H(\rho)m = O(\sqrt{n} \cdot \log n)$. Thus, for this parameter set the complexity of brute force search is $2^{O(\sqrt{n} \log n)}$. Consequently, in order to achieve 2^λ security we need to choose $n = \Omega\left(\left(\frac{\lambda}{\log \lambda}\right)^2\right)$. For this particular parameter set, brute force is the best known attack.

3.6.2. The Algorithm of Blum, Kalai and Wasserman

We will now provide an outline of the algorithm of Blum, Kalai and Wasserman [BKW03]. This algorithm attacks LPN in the high noise regime, i.e. $\rho = \frac{1}{2} - \epsilon$ for some $\epsilon = \epsilon(\lambda)$. The main theorem of [BKW03] can be stated as follows.

Theorem 3.4 (Blum Kalai Wasserman [BKW03])**.** *Let λ be a security parameter. Let $n = \mathsf{poly}(\lambda)$, $\epsilon = \epsilon(\lambda) \in (0, 1)$ and let a, b be such that $a \cdot b \geq n$. Let $m = \mathsf{poly}((2\epsilon)^{-2^a}, 2^b)$. There exists an algorithm solving $\mathsf{LPN}(n, m, \frac{1}{2} - \epsilon)$ in time $O(\mathsf{poly}(m))$.*

For a constant ϵ, setting $a = \frac{1}{2}\log(n)$ and $b = 2n/\log(n)$ yields $m = 2^{O(n/\log(n))}$, as

$$(2\epsilon)^{2^a} = (2\epsilon)^{\sqrt{n}} = 2^{O(n/\log(n))}$$

and

$$2^b = 2^{2n/\log(n)}.$$

Thus, the algorithm runs in time $2^{O(n/\log(n))}$ and needs $2^{O(n/\log(n))}$ samples. We will briefly outline a few ideas that lead to this algorithm. Let $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ be an LPN instance. Assume that we want to find the first bit s_1 of the secret vector $\mathbf{s}$. All other bits can be recovered analogously. The basic strategy is to try to find a vector $\mathbf{h} \in \mathbb{F}_2^m$ of weight 2^a such that $\mathbf{h}^T\mathbf{A} = \mathbf{e}_1^T$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{F}_2^n$ is the first unit vector. Once we found such a $\mathbf{h}$, it holds that

$$\mathbf{h}^T\mathbf{y} = \mathbf{h}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \mathbf{e}_1^T\mathbf{s} + \mathbf{h}^T\mathbf{y} = s_1 + \mathbf{h}^T\mathbf{e}.$$

By Lemma 2.2, for a fixed $\mathbf{h}$ the bias of $\mathbf{h}^T\mathbf{y}$ is

$$\frac{1}{2}(1 - 2(\frac{1}{2} - \epsilon))^{\mathsf{wgt}(\mathbf{h})} = \frac{1}{2} \cdot (2\epsilon)^{\mathsf{wgt}(\mathbf{h})} = \frac{1}{2}(2\epsilon)^{2^a}.$$

To amplify the success probability, this procedure is repeated $O(\frac{1}{2}(2\epsilon)^{2^a})$ times with fresh sets of samples and a majority vote is taken. By the Chernoff bound (Theorem 2.1), the majority vote is identical to s_1, except with negligible probability.

Such a vector $\mathbf{h}$ is found using a rather sophisticated recursive technique. The computation only uses $\mathbf{A}$. This ensures that $\mathbf{e}$ is independent of $\mathbf{e}$ and hence $\mathbf{h}^T\mathbf{e}$ has the right distribution. For details refer to [BKW03].

3.6.3. The Algorithm of Lyubashevsky

Lyubashevsky [Lyu05] noticed that the BKW algorithm can be used to solve LPN given a small polynomial amount of samples, if a random self reduction is used to amplify the amount of samples. Specifically, let $m = n^{1+\alpha}$ for some $\alpha > 0$ and let $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ be an instance of $\mathsf{LPN}(n, m, \mathsf{Ber}(m, \frac{1}{2} - \epsilon))$ for a constant $\epsilon > 0$. Let $\mathbf{h} \in \mathbb{F}_2^m$ be chosen uniformly from $\mathsf{S}_m(\lceil \frac{2n}{\log(m)} \rceil)$. Then we can argue that $(\mathbf{a}' = \mathbf{A}^T\mathbf{h}, y' = \mathbf{h}^T\mathbf{y})$ is a sample from $\mathsf{LPN}(n, \mathsf{Ber}(\frac{1}{2} - \frac{1}{2}(2\epsilon)^{\frac{2n}{\log(m)}}))$. First notice that

$$y' = \mathbf{h}^T(\mathbf{As} + \mathbf{e}) = \mathbf{h}^T\mathbf{As} + \mathbf{h}^T\mathbf{e}.$$

By Lemma 2.2, it holds that the bias of $e' = \mathbf{h}^T\mathbf{e}$ is

$$\epsilon' = \frac{1}{2}(2\epsilon)^{\mathsf{wgt}(\mathbf{h})} = (2\epsilon)^{\frac{2n}{\log(m)}}.$$

Moreover, by the generalized leftover hash lemma [DORS08], $\mathbf{Ah}^T$ is statistically close to uniform, even given e'. Thus, $(\mathbf{a}', y')$ is statistically close to a correct sample of $\mathsf{LPN}(n, \mathsf{Ber}(\frac{1}{2} - \frac{1}{2}(2\epsilon)^{\frac{2n}{\log(m)}}))$. Using this technique, we can generate an arbitrary number of fresh samples and use the BKW algorithm to find $\mathbf{s}$. Setting $a = \frac{1}{2}\log(\log(n))$ and $b = 2n/\log(\log(n))$ yields

$$(2\epsilon')^{2^a} = (4\epsilon^{2n/\log(m)})^{\sqrt{\log(n)}} = (4\epsilon^{\frac{2n}{(1+\alpha)\log(n)}})^{\sqrt{\log(n)}} = 2^{O(n/\sqrt{\log(n)})} = 2^{O(n/\log\log(n))}$$

and

$$2^b = 2^{2n/\log\log(n)}.$$

Thus, Theorem 3.4 provides an algorithm that runs in time $2^{O(n/\log(\log(n)))}$ using $2^{O(n/\log(\log(n)))}$ samples. These samples can be efficiently generated from $(\mathbf{A}, \mathbf{y})$ as sketched above.

3.6.4. Overview of Attacks

Table 3.1 provides a summary of the reviewed attacks.

Attack	noise bound	required Samples	Runtime
Brute Force Secret	-	-	$2^{\Theta(n)}$
Brute Force Error	-	-	$2^{\Theta(H(\rho)m)}$
BKW [BKW03]	$\frac{1}{2} - \epsilon$	$2^{O(n/\log(n))}$	$2^{O(n/\log(n))}$
Lyubashevsky [Lyu05]	$\frac{1}{2} - \epsilon$	$n^{1+\epsilon}$	$2^{O(n/\log\log(n))}$

Figure 3.1.: Comparison of LPN attacks

3.7. Worst-Case Hardness of LWE with Gaussian Errors

The hardness of the LWE problem might be conjectured as an assumption in its own right. However, by now the gold standard in lattice based cryptography is *worst-case hardness*. For the case of LWE, this means that any efficient algorithm that solves LWE on average can be used to construct an efficient algorithm that solves all instances of certain lattice problems. To the best of our knowledge, such strong hardness guarantees only exists for lattice based hardness assumptions.

The first such worst-to-average case connection for LWE was provided by Regev [Reg05]. Regev showed that any efficient algorithm solving the LWE search problem can be used to construct efficient *quantum* algorithms solving either the shortest independent vectors (SIVP) problem or approximate shortest vectors (GapSVP) problem (c.f. Section 2.6.2).

Theorem 3.5 (Worst-to-Average Case Reduction [Reg05])**.** *Let λ be a security parameter. Let $n = \mathsf{poly}(\lambda)$ and $q = q(\lambda)$ be a modulus, let $\alpha = \alpha(\lambda) \in (0,1)$ be such that $\alpha q > 2\sqrt{n}$. If there exists a PPT-algorithm solving $\mathsf{LWE}(n, q, D_{\alpha q})$ with non-negligible probability, then there exists an efficient quantum-algorithm that approximates the decision-version of the shortest vector problem $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ and the shortest independent vectors problem $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$ in the worst case.*

Regev's worst-to-average case reduction relies crucially on specific properties of gaussian distributions. Assume that $\mathcal{A}$ is an efficient algorithm solving LWE. The reduction consists of two parts. In the first part, the adversary $\mathcal{A}$ is used to construct an *bounded-distance decoding* algorithm $\mathcal{B}$. $\mathcal{B}$'s input is a basis $\mathbf{B}$ of a lattice $\Lambda^\perp$ and a point point $\mathbf{x}$ sufficiently close to $\Lambda^\perp$. Moreover, $\mathcal{B}$ gets access to an oracle providing samples from a gaussian distribution $D_{\Lambda,r}$. $\mathcal{B}$'s task is to compute the unique point $\mathbf{z} \in \Lambda^\perp$ closest to $\mathbf{x}$. We omit the details how $\mathcal{B}$ is constructed from $\mathcal{A}$ and only mention that $\mathcal{B}$ uses the gaussian distribution $D_{\Lambda,r}$ to simulate LWE samples who's secret corresponds to the error on $\mathbf{x}$. Regev's insight was that such a $\mathcal{B}$ can be used to sample a gaussian distribution with an even shorter $r' < r$ using the power of quantum computation. We omit the details but notice that $\mathcal{B}$ can be used to self-improve the gaussian sampler iteratively. We finally end up with a gaussian sampler that samples very short vectors in Λ that allow us to solve either the SIVP or GapSVP problem.

The natural question that arises from this result is whether *quantum power* is essential for such a reduction. Peikert [Pei09] showed that under certain circumstances the quantum part of the reduction can be avoided.

Theorem 3.6 (Worst-to-Average Case Reduction [Pei09])**.** *Let λ be a security parameter. Let $n = \mathsf{poly}(\lambda)$, $\alpha = \alpha(\lambda) \in (0,1)$ and $q = q(\lambda) \geq 2^{n/2}$. Assume there exists an efficient algorithm $\mathcal{A}$ solving $\mathsf{LWE}(n, q, D_{\alpha q})$. Then there exists an efficient classical algorithm solving $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ in the worst case.*

The price one has to pay for the classical hardness of LWE are very large moduli q. The idea behind Peikert's theorem is essentially this. Again, $\mathcal{A}$ can be used to construct a bounded distance decoding algorithm $\mathcal{B}$ using the same construction as Regev [Reg05]. As discrete gaussian sampler [GPV08, Pei10] is used to provide $\mathcal{B}$ with gaussian samples from $D_{\Lambda,r}$ for a reasonably small (yet exponentially large) s.

The algorithm $\mathcal{B}$ is now used in an entirely different way than in Regev's reduction. Recall that the goal of the GapSVP_γ problem is to decide whether a lattice Λ (given by a basis $\mathbf{B}$) has a non-zero vector shorter than d or whether all non-zero vectors in Λ are longer than $\gamma \cdot d$, i.e. if $(\mathbf{B}, d)$ is a `YES` or at `NO` instance of GapSVP_γ. In case of a `NO` instance, we can think of Λ as a good code, one for which all codewords (i.e. lattice points) are well separated. More specifically, if $\mathbf{x} \in \Lambda$ and $\mathbf{e}$ is a random error of length shorter than $\gamma d/2$, then $\mathbf{x}$ can be uniquely recovered from $\mathbf{y} = \mathbf{x} + \mathbf{e}$, as $\mathbf{x}$ is the unique vector in Λ closest to $\mathbf{y}$. Thus, if we run the decoding algorithm $\mathcal{B}$ on $\mathbf{y}$ it is guaranteed to output $\mathbf{x}$.

If we do the same for a YES instance however, adding a random error $\mathbf{e}$ will *destroy* information about $\mathbf{x}$. In this case, Λ is a bad code, i.e. the spheres around codewords have large intersections. Precisely, given the error $\mathbf{y} = \mathbf{x} + \mathbf{e}$, there is, with substantial probability over the choice of $\mathbf{e}$, at least one $\mathbf{x}' \in \Lambda$ distinct from $\mathbf{x}$ such that $\|\mathbf{y} - \mathbf{x}'\|_2 \leq \gamma d/2$. But this means that if we run the decoding algorithm $\mathcal{B}$ on $\mathbf{y}$, it cannot know for sure whether $\mathbf{x}$ or $\mathbf{x}'$ is the point we used to compute $\mathbf{y}$. Thus, with substantial probability $\mathcal{D}$ will output something different from $\mathbf{x}$.

We can therefore use the algorithm $\mathcal{B}$ to decide GapSVP_γ. In Chapter 7, we will use a conceptually very similar technique to switch the error distribution in LWE.

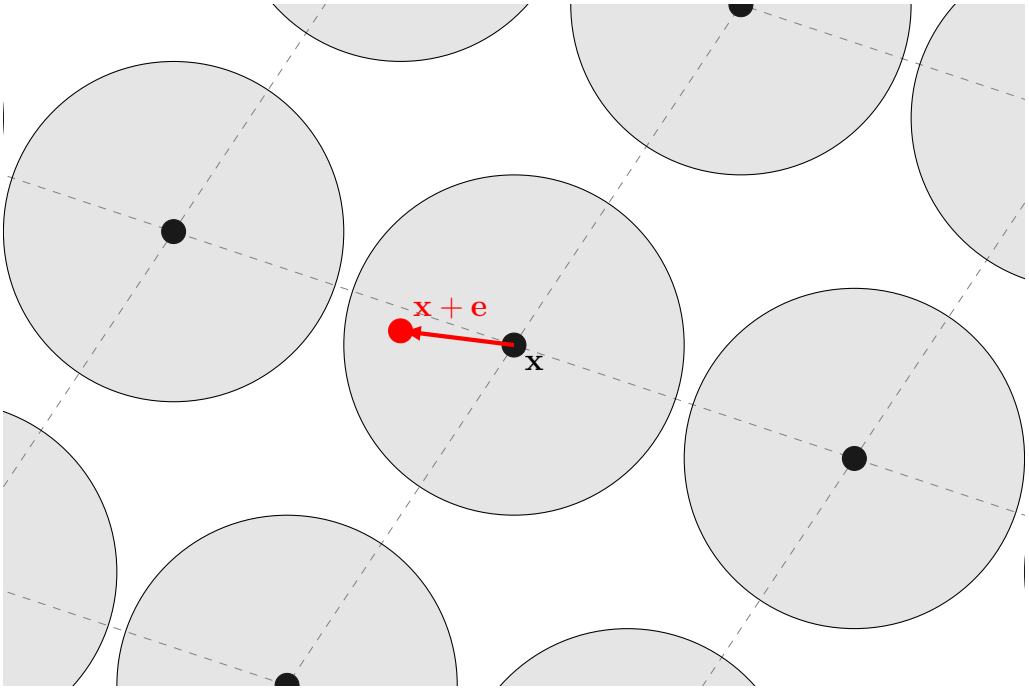

Figure 3.2.: A NO instance of GapSVP: The spheres are well separated

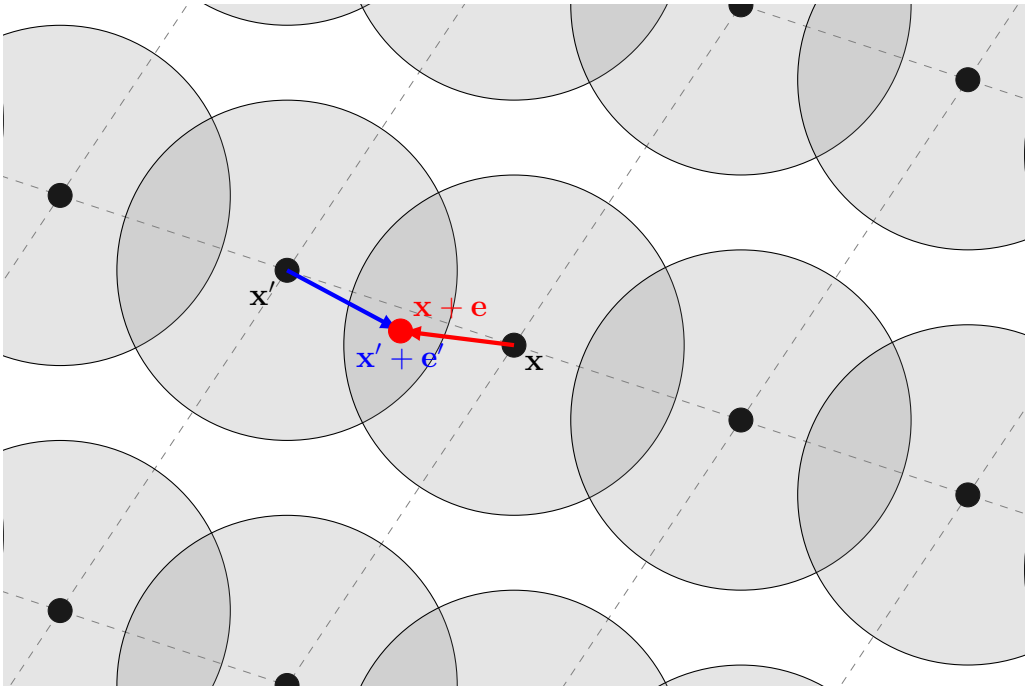

Figure 3.3.: A YES instance of GapSVP: The spheres are overlapping

3.8. Attacks on LWE

In this section, we will provide a brief overview of attacks against the LWE problem. We will only sketch the ideas underlying the attacks and omit parameter choices, as these are generally very subtle for lattice problems.

3.8.1. Attacks using Lattice Reduction

The standard way of attacking the LWE problem in literature is by using classical lattice reduction techniques. Let $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ be an LWE instance.

Decoding Attack

We will start by sketching an attack due to Lindner and Peikert [LP11]. The idea of the attack is to first use lattice reduction [LLL82] to compute a nearly orthogonal basis $\mathbf{B}$ of $\Lambda_q(\mathbf{A})$, then use a decoding algorithm like Babai's nearest hyperplane algorithm [Bab85] to decode $\mathbf{y}$. This attack works best if the number of samples m is large, as in this case the reduced basis $\mathbf{B}$ will be nearly orthogonal. The complexity of this attack is dominated by the lattice reduction step, i.e. the computation of $\mathbf{B}$. For standard parameter choices, this requires exponential time.

Distinguishing Attack

Variants of this attacks were described by Micciancio and Regev [MR08] as well as Rückert and Schneider [RS10]. The idea of this attack is to try to find a short dual vector $\mathbf{h}$ of $\Lambda_q(\mathbf{A})$, i.e. a $\mathbf{h}$ for which it holds $\mathbf{h}^T\mathbf{A} = \mathbf{0}$. Such a $\mathbf{h}$ can then be used to solve the decisional LWE problem for short error distributions. Given that the error vector $\mathbf{e}$ is sufficiently short, it holds that

$$\langle \mathbf{h}, \mathbf{y} \rangle = \mathbf{h}^T\mathbf{As} + \langle \mathbf{h}, \mathbf{e} \rangle = \langle \mathbf{h}, \mathbf{e} \rangle,$$

is small. On the other hand, for a uniformly random $\mathbf{u}$, $\langle \mathbf{h}, \mathbf{u} \rangle$ is also uniformly random and thus with high probability not small. For instantiations of LWE with search-to-decision equivalence (Theorems 3.1 and 3.2), such a distinguishing attack also yields an attack on search LWE.

3.8.2. Attacks using Linearization

Arora and Ge [AG11] provide a subexponential time algorithm that attacks LWE in the low noise case. Specifically, assume that $\alpha q < n^\epsilon$ for some $\epsilon \in (0, 1/2)$[5]. The algorithm attacks the decisional problem $\mathsf{DLWE}(n, q, D_{\alpha q})$. Let $(\mathbf{a}, y = \langle \mathbf{a}, \mathbf{s} \rangle + e)$ be an LWE sample. By the tail bound for the discrete gaussian distribution (Lemma 2.15), we know that the error terms $e \leftarrow_\$ D_{\alpha q}$ are smaller than a bound B with overwhelming probability. Thus, e suffices the condition $|e| \leq B$, which can be expressed by as a polynomial equation

$$P(e) = 0,$$

where P is a polynomial of degree $2B + 1$. As $e = y - \langle \mathbf{a}, \mathbf{s} \rangle$, this gives rise to a polynomial equation

$$Q_{\mathbf{a},y}(\mathbf{s}) = P(y - \langle \mathbf{a}, \mathbf{s} \rangle) = 0$$

for the secret $\mathbf{s}$. $Q_{\mathbf{a},y}$ is a multivariate polynomial of degree $2B + 1$ in n unknowns. Thus, there are at most $N = \binom{n+2B+1}{n}$ different monomials in $Q_{\mathbf{a},y}$. The strategy of Arora and Ge is then to replace each monomial be a new variable. This technique

[5] Notice that the worst-to-average case reductions in Theorems 3.5 and 3.6 require $\epsilon \geq \frac{1}{2}$

is commonly known as linearization [KS99]. Thus, each LWE sample gives rise to a linear equation in N unknowns. Such a linear equation system clearly has a solution, as we know that $\mathbf{s}$ exists. Such a solution can be found using linear algebra. However, this solution will not necessarily be one that corresponds to $\mathbf{s}$, as the linearization step may introduce many new solutions to the equation system.

However, Arora and Ge show that if y was chosen uniformly random instead, then with high probability the equation system does not possess a solution. Thus, we can distinguish $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from uniformly random. The time complexity of the algorithm is

$$\begin{aligned}
\mathsf{poly}(N) &= \mathsf{poly}\left(\binom{n+2B+1}{n}\right) \\
&= \mathsf{poly}\left(\binom{n+2B+1}{2B+1}\right) \\
&\leq \mathsf{poly}\left((n+2B+1)^{2B+1}\right) \\
&= 2^{O(n^{\epsilon}\log(n))},
\end{aligned}$$

which is subexponential. By the search-to-decision equivalence of LWE (Theorems 3.1 and 3.2), this also implies an attack on $\mathsf{LWE}(n, q, D_{\alpha q})$ with essentially the same complexity.

4. IND-CCA2 secure Public Key Encryption from Tag-Based Encryption

> Every innovation scraps its immediate predecessor and retrieves still older figures – it causes floods of antiques or nostalgic art forms and stimulates the search for museum pieces
>
> Marshall McLuhan and Barrington Nevitt [MN73]

4.1. Introduction

In this intermediate chapter, we will provide the technical foundations necessary for Chapters 5 and 6. This chapter contains no original contributions by the author. We will start by providing a brief overview of the history of IND-CCA secure encryption.

4.2. A Brief History of Chosen Ciphertext Security

As mentioned in [Dam91], it was noted in a discussion after the presentation of Rabin's public key cryptosystem [Rab79], that the very same property that enables proving security against passive adversaries leads to a complete break of the system if the adversary is not entirely passive. Thus, the standard security notion for public key cryptography, semantic security [GM82], falls short against active adversaries. Recall that semantic security, or security against chosen plaintext attacks (IND-CPA security) [MRS86, BDPR98] only guarantees that an *eavesdropping* (or passive) adversary will not learn any information about the message in an intercepted ciphertext *beyond* the a-priori information she already has. Once the adversary is given an additional resource, which for instance allows to *probe* the decryption function, IND-CPA security no longer provides any security guarantees. When a public

key encryption scheme is used for any task more complex than securely transmitting messages between two parties who otherwise do not interact, IND-CPA security is insufficient. This is especially the case if the adversary gets some feedback about whether the receiver is able to decrypt a given ciphertext. In some cases, such a feedback may be used to effectively implement a decryption oracle [Ble98].

The precise definition of IND-CCA security is given is Section 2.4.3. Just recall that in the IND-CCA1 security experiment the adversary is given a decryption oracle before it sees the challenge ciphertext, while in the IND-CCA2 experiment the adversary also gets access to the decryption oracle after seeing the challenge ciphertext, with the restriction that this decryption oracle decrypts only ciphertexts different from the challenge ciphertext.

The construction of IND-CPA secure public key encryption can usually be achieved in some natural and elegant way from a given decisional assumption [GM82, Gam84, Ale03, Reg05] and there exist generic and simple black-box constructions from any given injective trapdoor function [Yao82]. Roughly speaking, security reductions for IND-CPA security mostly just *pass along* their own challenge to the adversary in a simulated security experiment. Even more, the secret key sk is never used in the IND-CPA experiment and is, so to say, discarded right after its generation.

On the other hand, achieving (provable) IND-CCA security usually represents a tougher challenge. The reason for this is rooted in the fact that an IND-CCA adversary *demands* a decryption oracle. On one hand side, a security reduction must use the adversary's ability to break the IND-CCA security of a given scheme in an essential way to solve a hard computational problem. This means in particular that the reduction cannot use a *fully functional* secret key to simulate the decryption oracle. Otherwise, the reduction could just replace the adversaries ability to break the security by using its own secret key. Such a reduction would, of course, basically falsify the hardness of the problem it attacks, since it solves this problem unconditionally, i.e. independent of the adversary. Thus, a scheme we want to prove IND-CCA secure needs to be tailored in a way that enables a reduction to decrypt *most* ciphertexts, while still being able to use the adversary's ability to break the IND-CCA security to solve a hard problem.

4.2.1. Constructions from General Assumptions

Blum, Feldman and Micali [BFM88b] suggested to use non-interactive zero-knowledge proofs of knowledge (NIZKPoK) [BFM88a] to achieve IND-CCA1 security, but did not provide a construction. A similar approach was taken by Naor and Yung [NY90], who provided the first construction of an IND-CCA1 secure public key encryption scheme, based on the hardness of standard computational assumptions. Their enhanced scheme is composed of two instances of a standard IND-CPA secure scheme and a non-interactive zero-knowledge proof (NIZK) system. The public key of the enhanced scheme consists of two public keys of the standard scheme and a common reference string for the NIZK. Messages are encrypted twice, under each public key, and this pair of ciphertexts is augmented by a NIZK proving that both ciphertexts encrypt the same message. Decryption only decrypts valid ciphertexts, i.e. ciphertexts that contain a valid NIZK proof. Omitting many details, we will briefly explain why this trick allows for the simulation of a *proper* decryption oracle. Observe that it is sufficient to know one of the secret keys to simulate a decryption oracle, as the soundness of the NIZK guarantees that both ciphertexts encrypt the

same message. Thus, one of the two public keys may be chosen by the reduction, together with a corresponding secret key. This allows the reduction to faithfully simulate a decryption oracle. To embed its own challenge, the reduction uses the simulator of the NIZK to forge a consistency proof for the challenge ciphertext. If the inconsistency is detected by the adversary, it can either break the zero-knowledge property of the NIZK or the IND-CPA security of the public key encryption scheme.

Rackoff and Simon [RS91] observed that the notion of IND-CCA1 security is potentially insufficient for certain applications, as it excludes a so-called *playback* attack. In a playback attack the adversary tries to use its decryption oracle to attack its own challenge ciphertext. Clearly, one cannot allow the crudest playback attack, namely allowing the adversary to use its decryption oracle to directly decrypt the challenge ciphertext. Trivially, every scheme is insecure against this kind of attack. However, Rackoff and Simon [RS91] proposed to allow *any* other playback attack, i.e. the decryption oracle must decrypt every ciphertext that differs from the challenge ciphertext. This is exactly the security notion of adaptive chosen ciphertext security (IND-CCA2). Rackoff and Simon [RS91] also provided a construction of an IND-CCA2 secure public key encryption scheme, implementing the NIZKPoK approach of [BFM88b]. However, they required that every party (also the adversary) has their respective keys generated by a trusted third party.

Dolev, Dwork and Naor [DDN91] provided the first construction of an adaptive chosen ciphertext secure public key encryption scheme in the standard model. The central insight of [DDN91] is that ciphertexts of such an encryption scheme should be *non-malleable*. Roughly speaking, a public key encryption scheme is non-malleable if valid ciphertexts cannot be mauled into different valid ciphertexts, such that the corresponding plaintexts are related in a certain way. Put differently, this means that the only way of creating a new ciphertext is by encrypting a plaintext. While in [DDN91] it was conjectured that non-malleability is a stronger notion than ciphertext indistinguishability, it was later shown that for the case of adaptive chosen ciphertext attacks, the two notions are equivalent [BDPR98, DDN00]. The key idea behind the construction of [DDN91] is the following. While Naor and Yung [NY90] provide double encryptions of plaintext messages, Dolev, Dwork and Naor encrypt the plaintext under many public keys and use a NIZK proof to prove that all ciphertexts encrypt the same message. The main lever that allows proving this scheme IND-CCA2 secure is that each ciphertext is encrypted using a different subset of public keys selected from a master public key. On one hand side, knowing a single secret key for the public keys used to encrypt a message is sufficient to decrypt the message. Moreover, one can verify that all the other ciphertexts encrypt the same message by checking the NIZK proof. On the other hand, a simulator can construct a master public key such that it knows at least one (partial) secret key for every derived public key except for one. It will then be able to embed its own decryption challenge in the challenge ciphertext, using the NIZK simulator to forge a consistency proof. The techniques of [DDN91] have been a great source of inspiration for subsequent constructions, which focused at implementing ciphertext consistency checks without resorting to NIZK proofs. Specifically, subsequent works identified and abstracted the main technical components of [DDN91], which, for instance gave rise to the notion of tag-based encryption discussed in the next section.

At the heart of all constructions discussed so far are non-interactive zero-knowledge proofs. A series of works [Sah99, SCO+01, Lin03] showed how the construction of Naor and Yung [NY90] can be proven IND-CCA2 secure, if instantiated with an

appropriate NIZK system. Non-interactive zero-knowledge is amongst the heaviest machinery theoretical cryptography has to offer. Consequently, all these results should be considered as feasibility results, as trying to instantiate them in the real world would lead to astronomical key and ciphertext sizes.

4.2.2. Efficient Constructions in the Random Oracle Model

Thus, the question arose whether IND-CCA2 secure schemes can be constructed that are efficient in a *practical* sense, with the intention of using such schemes in real world cryptographic applications. Bellare and Rogaway [BR93, BR94] provided a simple and efficient construction of an IND-CCA2 secure public key encryption scheme based on any trapdoor permutation in the *random oracle model* (ROM) [FS86]. In the ROM, each party has access to a random oracle, which models an idealized hash function. In short, a random oracle is a random function that can be queried on arbitrary inputs. The critical point is that, even for adversarial parties the random oracle is like a black box. Its input-output behavior can be probed, but its implementation details are inaccessible. As a consequence, the only way of obtaining function values of a random oracle is by using it, i.e. providing input to the black box and receiving the output. In a way, this gives security reductions an unfair advantage. The construction of Bellare and Rogaway [BR93] is crafted in a way such that a valid ciphertext cannot be created without using the random oracle. This is of immense help when basing the security of the construction on the one-wayness of the trapdoor permutation. Whenever the adversary uses the random oracle in the (simulated) IND-CCA2 experiment, it reveals its input and random coins it uses for encryption. Put differently, its shows *awareness* of the plaintext corresponding to a ciphertext. This provides the security reduction an advantage in simulating the decryption oracle. As valid ciphertexts can only be created by using the random oracle, the security reduction simply *eavesdrops* every query the adversary asks the random oracle and checks if a ciphertext the adversary queries its decryption oracle with can be constructed from the random oracle queries of the adversary. Thus, the security reduction does not need the secret key of the trapdoor perimutation to simulate the decryption oracle.

The obvious concern with proofs in the random oracle model is what their implications for the real world are. Bellare and Rogaway [BR93] suggest to use a sufficiently complicated cryptographic hash function instead of the random oracle in a real world implementation [1]. However, unlike a random oracle a hash function has a short and explicit description in the form of its algorithm. Thus, a real world adversary may find other ways of obtaining function values of the hash function than just evaluating it. Consequently, the random oracle methodology is a heuristic at best. Concerns against proofs in the random oracle model grew when a series of works demonstrated that there are constructions in the random oracle model that are *uninstantiatable* [CGH98, Nie02, GK03, CGH04, BBP04].

4.2.3. Efficient Constructions in the Standard Model

The first step towards an efficient standard model IND-CCA secure public key encryption scheme was taken by Damgård [Dam91]. Damgård constructed a variant of the ElGamal scheme which is IND-CCA1 secure under the so-called *knowledge of exponent* assumption in cyclic groups. In a nutshell, if $\mathbb{G}$ is a cyclic group in

[1] An instantiation of [BR93] using the RSA trapdoor function [RSA78], called RSA-ES-OAEP, is described in the standard PKCS #11

which the Diffie-Hellman problem is hard and $g, h \leftarrow_\$ \mathbb{G}$ are randomly chosen, then this assumption states that the only way of computing (g^a, h^a) given (g, h) is by *knowing* a. This is formalized in terms of an *knowledge extractor*: For any efficient randomized algorithm $\mathcal{A}$ computing (g^a, h^a) given (g, h), there exists an efficient randomized algorithm $\mathcal{E}$ computing a given (g, h) which uses the same random coins as $\mathcal{A}$. Such an extractor can be used in a similar way to extract like the random oracle in the construction of [BR93], i.e. it gives a security reduction the extra edge to simulate a decryption oracle without having to know the trapdoor of the scheme. However, by now such knowledge assumptions [Nao03] are also considered highly nonstandard and there is evidence that they cannot be based on standard computational problems [GW11].

In a celebrated work, Cramer and Shoup [CS98] provided a very efficient IND-CCA2 secure public key encryption scheme in the standard model, based on standard computational assumptions. The key technical ingredient for this construction are *hash proof systems*. Omitting details, hash proof systems, as generalized in [CS02], are private coin proof systems for hard subset membership problems $\mathcal{L} \in \mathcal{NP}$. The private key of a hash proof system can be used to both verify the correctness of a proof and fake proofs for false statements. Hash proof systems can be used to construct public key encryption schemes in a way, such that a successful IND-CCA2 adversary can be used to decided the problem $\mathcal{L}$, contradicting its hardness. Following works have shown that hash proof systems can be constructed from a large variety of *decisional* number theoretic hardness assumptions [CS02, Luc02, KD04, CKS08]. Moreover, hash proof systems have found applications beyond chosen ciphertext security, for instance in the context of leakage resilience [NS09, HLAWW13].

Canetti, Halevi and Katz [CHK04] provided a generic and efficient construction of IND-CCA2 secure encryption schemes from any selective identity secure identity based encryption scheme. This will be discussed in more detail in the next section. Hofheinz and Kiltz [HK09] provided an efficient construction of an IND-CCA2 secure public key encryption scheme based on the hardness of factoring. Their scheme is a modification of the Blum Goldwasser [BG84] encryption scheme, which in turn relies on the Blum Blum Shub pseudorandom generator [BBS82]. The approach of Hofheinz and Kiltz deviates significantly from previous techniques, but borrows concepts from tag-based encryption explained in the next section.

Peikert and Waters [PW08] introduced the notion of lossy trapdoor functions. In a nutshell, lossy trapdoor functions can be operated in two different modes: An injective mode and a lossy mode. The injective mode allows efficient inversion given a trapdoor, while the lossy mode cannot be inverted information theoretically, i.e. each image of the function corresponds to a vast (i.e. superpolynomial) number of preimages. Lossy trapdoor functions give rise to natural and efficient constructions of IND-CCA2 secure encryption schemes following the tag-based encryption framework. Peikert and Waters [PW08] provided constructions of lossy trapdoor functions from the decisional Diffie Hellman and LWE problems. Thus, [PW08] provided the first efficient standard model construction of an IND-CCA2 secure scheme from a lattice assumption. Freeman et al. [FGK+09] provided constructions of lossy trapdoor functions from further assumptions. Later, Peikert [Pei09] and Micciancio and Peikert [MP12] provided more efficient constructions of IND-CCA2 secure public key encryption schemes from LWE, also following the tag-based encryption paradigm. Rosen and Segev [RS09] generalized the techniques of Peikert and Waters [PW08]. Rosen and Segev identified that a weaker notion than lossiness, called correlation se-

curity, is sufficient for the Peikert and Waters' construction of an IND-CCA2 secure scheme.

4.3. Tag-Based Encryption

The constructions of IND-CCA secure schemes presented in Chapters 5 and 6 will be based on the notion of *tag-based encryption*. This notion was introduced by MacKenzie, Reiter and Yang [MRY04], later refined by Kiltz [Kil06]. The purpose of this notion is to provide a general class of schemes for which the transformation-technique of Canetti, Halevi and Katz [CHK04] can be applied. Canetti et al. [CHK04] show how any identity based encryption scheme (IBE) can be transformed into an IND-CCA2 secure scheme using simple black-box techniques. We will refer to this technique as CHK-transformation henceforth.

In a tag-based encryption scheme, each ciphertext is associated with a *tag*, which is generally an unstructured bit string of sufficient length. The tag is provided as an additional input to the encryption and decryption algorithms. The security experiment for tag-based encryption is identical to the IND-CCA experiment, with the exception that the adversary announces a *target-tag* τ^* before seeing the public key of the scheme. Consequently, the challenge-ciphertext will be encrypted using the target-tag τ^*. Moreover, the adversary's decryption oracle will not answer decryption-queries for ciphertexts associated with the target-tag.

The notion of tag-based encryption is particularly useful (and originally intended) as an intermediate stage in the construction of IND-CCA secure public key encryption. The construction of IND-CCA1 secure public key encryption from tag-based encryption is particularly simple. Given that the tag-space has super-polynomial size, we obtain an IND-CCA1 secure scheme from a tag-based scheme by letting a modified encryption algorithm sample tags uniformly at random and append the tag to the ciphertext. The CCA1 security experiment now coincides with the security experiment for tag-based encryption.

We will now provide the definitions for tag-based encryptions, as taken from [Kil06]. We start by providing the syntactic definition of tag-based encryption schemes.

Definition 4.1. *A tag-based encryption scheme* TBE *consists of three PPT-algorithms* $\mathsf{TBE.KeyGen}$, $\mathsf{TBE.Enc}$ *and* $\mathsf{TBE.Dec}$

- $\mathsf{TBE.KeyGen}(1^\lambda)$*: Takes as input a security-parameter* 1^λ *and outputs a pair of public and secret keys* (pk, sk).
- $\mathsf{TBE.Enc}(pk, \tau, \mathfrak{m})$*: Takes as input a public key* pk*, a tag* τ *and a plaintext* $\mathfrak{m}$ *and outputs a ciphertext* $\mathfrak{c}$.
- $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c})$*: Takes as input a secret key* sk*, a tag* τ *and a ciphertext* $\mathfrak{c}$ *and outputs a plaintext* $\mathfrak{m}$ *or* $\perp$.

We will assume that the plaintext-space $\mathfrak{M}_\lambda$ *and the tag-space* $\mathfrak{T}_\lambda$ *of* TBE *only depend on the security parameter* λ.

A standard requirement for encryption schemes is completeness. The definition is analogous to the completeness definition for public key encryption schemes (Definition 2.7), except that the tags τ have to be taken into account.

Definition 4.2. *We say that* $\mathsf{TBE} = (\mathsf{TBE.KeyGen}, \mathsf{TBE.Enc}, \mathsf{TBE.Dec})$ *is complete, if it holds for all plaintexts* $\mathfrak{m} \in \mathfrak{M}_\lambda$ *and all tags* $\tau \in \mathfrak{T}_\lambda$ *that*

$$\Pr[\mathsf{TBE.Dec}(sk, \tau, \mathsf{TBE.Enc}(pk, \tau, \mathfrak{m})) \neq \mathfrak{m} : (pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^k)] < \mathsf{negl}(\lambda).$$

MacKenzie et al. [MRY04] provide a definition of ciphertext indistinguishability for tag-based encryption where the adversary is allowed to choose the target-tag adaptively. More precisely, the adversary first learns the public key pk and is only allowed to query the decryption oracle before it has to announce a target tag τ^* together with the challenge-messages $\mathfrak{m}_0$ and $\mathfrak{m}_1$. Kiltz [Kil06] observed that adaptive security is not necessary for the CHK-transformation and introduced the notion of *selective* IND-CCA security for tag-based encryption. This security-notion requires the adversary to announce the target-tag both before seeing the public key and being allowed to query the decryption oracle.

Definition 4.3. *We say a tag-based encryption-scheme* TBE *is ciphertext-indistinguishable under selective tag and adaptively chosen ciphertext attacks (IND-STAG-CCA2), if every PPT-adversary* $\mathcal{A}$ *has success-probability at most negligibly better than* $1/2$ *in the experiment* $\mathsf{IND\text{-}STAG\text{-}CCA2}$*, i.e.* $\Pr[\mathsf{IND\text{-}STAG\text{-}CCA2}(\mathcal{A}) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$.

Experiment IND-STAG-CCA2
$(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathtt{init}, 1^\lambda)$
$(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \cdot, \cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{c}^* \leftarrow \mathsf{TBE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \cdot, \cdot)}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \tau, \mathfrak{c})$
If $\tau = \tau^*$
Return $\perp$
$\mathfrak{m} \leftarrow \mathsf{TBE.Dec}(sk, \tau, \mathfrak{c})$
Return $\mathfrak{m}$

Observe that, unlike in the definiton of the IND-CCA2 experiment, the decryption oracle is the same in the `find`- and `guess`-stage. In both stages, $\mathcal{A}$ is not allowed to query its decryption-oracle $\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \cdot, \cdot)$ with the tag τ^*.

From a technical perspective, the usefulness of the notion of tag-based encryption stems from the fact that in the IND-STAG-CCA2 experiment, the decision whether the decryption-oracle decrypts its query or not does not depend on the ciphertext $\mathfrak{c}$, but only on the tag τ. Contrast this to the notion of IND-CCA2 security, where the adversary may query its decryption-oracle in the `guess`-phase on any ciphertext but the challenge-ciphertext $\mathfrak{c}^*$.

4.4. The Canetti-Halevi-Katz Transformation

We will now show how to transform an IND-STAG-CCA2 secure tag-based encryption scheme TBE into an IND-CCA secure public key encryption scheme PKE via the CHK transformation. The proof follows [CHK04, MRY04, Kil06].

Construction 4.1 (CHK Transformation [CHK04, MRY04, Kil06])**.** *Let* $\mathsf{TBE} = (\mathsf{TBE.KeyGen}, \mathsf{TBE.Enc}, \mathsf{TBE.Dec})$ *be a tag-based encryption scheme with tag-space*

$\mathfrak{T}_\lambda$. *Let* $\mathsf{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Verify})$ *be a one-time signature scheme for which the verification keys* vk *generated by* $\mathsf{OTS.Gen}(1^\lambda)$ *can be embedded into* $\mathfrak{T}_\lambda$. *The public key encryption scheme* $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ *is given as follows.*

- $\mathsf{PKE.KeyGen}(1^\lambda)$*:*
  $(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
  Return (pk, sk)

- $\mathsf{PKE.Enc}(pk, \mathfrak{m})$*:*
  $(vk, sgk) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
  $\tau \leftarrow vk$
  $\mathfrak{c}' \leftarrow \mathsf{TBE.Enc}(pk, \tau, \mathfrak{m})$
  $\sigma \leftarrow \mathsf{OTS.Sign}(sgk, \mathfrak{c}')$
  $\mathfrak{c} \leftarrow (\mathfrak{c}', vk, \sigma)$
  Return $\mathfrak{c}$

- $\mathsf{PKE.Dec}(sk, \mathfrak{c})$*:*
  Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
  $\tau \leftarrow vk$
  If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 0$
  Return $\perp$
  $\mathfrak{m} \leftarrow \mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
  Return $\mathfrak{m}$

Before showing that this construction actually yields an IND-CCA2 secure encryption scheme, we will first comment on the requirement that the verification keys output by $\mathsf{OTS.Gen}(1^\lambda)$ can be embedded into the tag-space $\mathfrak{T}_\lambda$.

While this requirement seems like a restriction for the choice of the one-time signature scheme OTS at first, standard techniques can be employed to increase the size of the tag-space $\mathfrak{T}_\lambda$ such that the verification keys vk *fit* into it. Given that the size of the tag-space $\mathfrak{T}_\lambda$ is at least sub-exponential, i.e. $|\mathfrak{T}_\lambda| = 2^{\Omega(\mathsf{poly}(\lambda))}$, one may pursue one of the following approaches.

The first approach is to choose different security-parameters λ_1 for TBE and λ_2 for OTS to ensure that verification keys vk output by $\mathsf{OTS.Gen}(1^{\lambda_2})$ can be embedded into $\mathfrak{T}_{\lambda_1}$. Assume that $\mathfrak{T}_{\lambda_1} = \{0,1\}^{p_1(\lambda_1)}$ and that the verification keys vk can be described by $p_2(\lambda_2)$ bits, for some polynomials p_1 and p_2. Then we can always find a polynomial p_3 such that $p_1(p_3(\lambda_2)) \geq p_2(\lambda_2)$. Thus, choosing $\lambda_1 \geq p_3(\lambda_2)$ ensures that the tag-space $\mathfrak{T}_{\lambda_1}$ is large enough such that verification keys vk can be represented as elements of $\mathfrak{T}_{\lambda_1}$.

While this *parameter-tweak* is sound in theory, it may lead to impractical and unnecessarily large security parameters λ_1. As an alternative, we can *artificially* increase the size of the tag-space $\mathfrak{T}_\lambda$ by taking a collision-resistant hash-function (CRHF) and hash tags of size $\mathsf{poly}(\lambda)$ down to size λ. While this would introduce further hardness assumption (i.e. the existence of CRHFs), we note that full collision resistance is not necessary here. Instead, target collision resistance, as provided by universal one way hash functions (UOWHF) is sufficient, as observed in [DDN91]. This holds because in the IND-STAG-CCA2 experiment the adversary must announce a target-tag τ^* before getting to see the public key. Thus, we can include a key for a UOWHF in the public key and compress tags using the UOWHF. Unlike

CRHFs, UOWHFs are known to be constructible from one-way functions [Rom90]. Thus, this transformation will not introduce new complexity assumptions.

We will now show that Construction 4.1 actually yields an IND-CCA2 secure public key cryptosystem, given that the underlying tag-based encryption scheme is IND-STAG-CCA2 secure.

Theorem 4.1 ([CHK04, MRY04, Kil06])**.** *Assume that* TBE *is an IND-STAG-CCA2 secure tag-based encryption scheme and that the size of the tag-space $\mathfrak{T}_\lambda$ is at least subexponential, i.e. $|\mathfrak{T}_\lambda| \geq 2^{\mathsf{poly}(\lambda)}$. Assume further that* OTS *is a strongly one-time unforgeable signature scheme. Then the public key encryption scheme* PKE *given in Construction 4.1 is IND-CCA2 secure.*

Proof. Assume towards contradiction that there exists a PPT-adversary $\mathcal{A}$ with non-negligible advantage ϵ against the IND-CCA2 security of PKE. Consider the experiments **Game 0**, **Game 1** and **Game 2**.

Game 0
$(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec1}}(sk,\cdot)}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0,1\}$
$(vk^*, sgk^*) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$\tau^* \leftarrow vk$
$\mathfrak{c}'^* \leftarrow \mathsf{TBE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$\sigma^* \leftarrow \mathsf{OTS.Sign}(sgk^*, \mathfrak{c}'^*)$
$\mathfrak{c}^* \leftarrow (\mathfrak{c}'^*, vk^*, \sigma^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec2}}(sk,\mathfrak{c}^*,\cdot)}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec1}}(sk, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$
$\mathcal{O}_{\mathsf{Dec2}}(sk, \mathfrak{c}^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
If $\mathfrak{c} = \mathfrak{c}^*$ return $\perp$
$\tau \leftarrow vk$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

Clearly, **Game 0** is the IND-CCA2 experiment for PKE, for which we have substituted the algorithms PKE.KeyGen, PKE.Enc and PKE.Dec with their implementations according to Construction 4.1.

In **Game 1** we have changed two aspects. First, the generation of the keys vk^* and sgk^* and the assignment of the tag τ^* has been moved to the beginning of the experiment. Second, the `find`-stage decryption oracle $\mathcal{O}_{\mathsf{Dec1}}$ now receives the challenge tag τ^* as additional input and returns $\perp$ if the adversary sends a decryption query with tag $\tau = \tau^*$.

In **Game 2** we make the implementation of the decryption oracle $\mathcal{O}_{\mathsf{Dec2}}$ identical to $\mathcal{O}_{\mathsf{Dec1}}$. More specifically, instead of receiving additional input $\mathfrak{c}^*$, $\mathcal{O}_{\mathsf{Dec2}}$ now receives additional input τ^* and the check whether $\mathfrak{c} = \mathfrak{c}^*$ is replaced by the check $\tau = \tau^*$.

We will first show that **Game 0**, **Game 1** and **Game 2** are computationally indistinguishable from the view of $\mathcal{A}$. We will subdivide this into two claims. A third claim will assert that $\mathcal{A}$ has only-negligible success-probability in **Game 2** given that TBE is IND-STAG-CCA2 secure. This will conclude the proof.

Game 1

$(vk^*, sgk^*) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$\tau^* \leftarrow vk$
$(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec1}}(sk,\tau^*,\cdot)}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}'^* \leftarrow \mathsf{TBE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$\sigma^* \leftarrow \mathsf{OTS.Sign}(sgk^*, \mathfrak{c}'^*)$
$\mathfrak{c}^* \leftarrow (\mathfrak{c}'^*, vk^*, \sigma^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec2}}(sk,\mathfrak{c}^*,\cdot)}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec1}}(sk, \tau^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\tau = \tau^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

$\mathcal{O}_{\mathsf{Dec2}}(sk, \mathfrak{c}^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\mathfrak{c} = \mathfrak{c}^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

Claim 1

We claim that **Game 0** and **Game 1** are statistically close from the view of $\mathcal{A}$. First notice that moving the generation of vk^* and sk^* to the beginning of the experiment makes no difference from the view of $\mathcal{A}$. Given that $\mathcal{A}$ does not send a decryption-query with tag τ^* to $\mathcal{O}_{\mathsf{Dec1}}$, the decryption oracle $\mathcal{O}_{\mathsf{Dec1}}$ behaves identically from the view of $\mathcal{A}$. Thus, in this case **Game 0** and **Game 1** are identically distributed from the view of $\mathcal{A}$. We will now bound the probability that $\mathcal{A}$ sends a decryption query with tag $\tau = \tau^*$ to the decryption-oracle $\mathcal{O}_{\mathsf{Dec1}}$.

Assume that $\mathcal{A}$ makes at most $q \leq \mathsf{poly}(\lambda)$ many decryption-queries to $\mathcal{O}_{\mathsf{Dec1}}$. Recall that the size of the tag-space is at least sub-exponential, i.e. it holds that $|\mathfrak{T}_\lambda| \geq 2^{\mathsf{poly}(\lambda)}$. Call the tags of the q decryption-queries $\tau_1, \ldots, \tau_q$. Since the challenge-tag τ^* is not announced to $\mathcal{A}$ before the $\mathtt{guess}$-stage, $\mathcal{A}$ can exclude at most one additional tag per decryption query from the list of possible candidates for τ^*. Thus, we can bound $\Pr[\tau_i = \tau^*] \leq \frac{i}{|\mathfrak{T}_\lambda|} \leq \frac{q}{|\mathfrak{T}_\lambda|}$ for $i \in \{1, \ldots, q\}$. By a union-bound it holds that

$$\Pr[\exists i : \tau_i = \tau^*] \leq \sum_{i=1}^{q} \underbrace{\Pr[\tau_i = \tau^*]}_{\leq q/|\mathfrak{T}_\lambda|} \leq q \cdot \frac{q}{|\mathfrak{T}_\lambda|} \leq \frac{\mathsf{poly}(\lambda)}{2^{\mathsf{poly}(\lambda)}},$$

which is negligible. Therefore, **Game 0** and **Game 1** are statistically close from the view of $\mathcal{A}$.

Claim 2

We claim that **Game 1** and **Game 2** are computationally indistinguishable from the view of $\mathcal{A}$, given that the signature scheme OTS is strongly existentially unforgeable under one-time chosen message attacks. First observe that if all the decryption-queries for $\mathcal{O}_{\mathsf{Dec2}}$ with tag τ^* are rejected in **Game 1**, then $\mathcal{A}$'s views in **Game 1** and **Game 2** are identically distributed. Thus, in order to distinguish both experiments, $\mathcal{A}$ must be able to produce (with non-negligible probability) a decryption query with tag τ^* in **Game 1**, which is not rejected by the decryption oracle $\mathcal{O}_{\mathsf{Dec2}}$. The decryption oracle $\mathcal{O}_{\mathsf{Dec2}}$ rejects decryption queries $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$ if it holds that

Game 2

$(vk^*, sgk^*) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$\tau^* \leftarrow vk$
$(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec1}}(sk,\tau^*,\cdot)}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}'^* \leftarrow \mathsf{TBE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$\sigma^* \leftarrow \mathsf{OTS.Sign}(sgk^*, \mathfrak{c}'^*)$
$\mathfrak{c}^* \leftarrow (\mathfrak{c}'^*, vk^*, \sigma^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec2}}(sk,\tau^*,\cdot)}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$

$\mathcal{O}_{\mathsf{Dec1}}(sk, \tau^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\tau = \tau^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

$\mathcal{O}_{\mathsf{Dec2}}(sk, \tau^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\tau = \tau^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

$\mathfrak{c} = \mathfrak{c}^*$ or $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 0$. Thus, if a decryption query with tag $\tau^* = vk^*$ is not rejected, it must hold that $\mathsf{OTS.Verify}(vk^*, \mathfrak{c}', \sigma) = 1$ and $(\mathfrak{c}', \sigma) \neq (\mathfrak{c}'^*, \sigma^*)$.

Assume thus that $\mathcal{A}$ produces a decryption-query $\mathfrak{c} = (\mathfrak{c}', vk^*, \sigma)$ for $\mathcal{O}_{\mathsf{Dec2}}$ with $\mathsf{OTS.Verify}(vk^*, \mathfrak{c}', \sigma) = 1$ and $(\mathfrak{c}', \sigma) \neq (\mathfrak{c}'^*, \sigma^*)$ with non-negligible probability ϵ. We will construct adversary $\mathcal{B}$ that breaks the OT-sEUF-CMA security of OTS with probability ϵ.

Adversary $\mathcal{B}$ simulates **Game 1**, including the decryption oracles $\mathcal{O}_{\mathsf{Dec1}}$ and $\mathcal{O}_{\mathsf{Dec2}}$ faithfully, except for the following differences. Instead of generating the verification key vk^* using $\mathsf{OTS.Gen}$, it uses the verification key vk^* provided by the OT-sEUF-CMA experiment. Moreover, it uses the signing oracle $\mathcal{O}_{\mathsf{Sign}}(sgk^*, \cdot)$ provided by the OT-sEUF-CMA experiment to generate the signature σ^* on the challenge-ciphertext $\mathfrak{c}^*$. Finally, once $\mathcal{A}$ sends a decryption-query $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$ to $\mathcal{O}_{\mathsf{Dec2}}$, $\mathcal{B}$ checks whether $(\mathfrak{c}', \sigma) \neq (\mathfrak{c}'^*, \sigma^*)$ and $\mathsf{OTS.Verify}(vk^*, \mathfrak{c}', \sigma) = 1$, and if so halts the simulation and outputs $(\mathfrak{c}', \sigma)$.

Observe first that from $\mathcal{A}$'s view **Game 1** and $\mathcal{B}$'s simulation are identically distributed, since the generation of the keys vk^* and sgk^* as well as the computation of the signature σ^* on the challenge ciphertext are only transfered to the OT-sEUF-CMA experiment but remain otherwise identical.

Since we assume that $\mathcal{A}$ produces a decryption-query $\mathfrak{c} = (\mathfrak{c}', vk^*, \sigma)$ for $\mathcal{O}_{\mathsf{Dec2}}$ with $\mathsf{OTS.Verify}(vk^*, \mathfrak{c}', \sigma) = 1$ and $(\mathfrak{c}', \sigma) \neq (\mathfrak{c}'^*, \sigma^*)$ with probability ϵ in **Game 1**, $\mathcal{B}$ wins the OT-sEUF-CMA experiment with probability ϵ, contradicting the OT-sEUF-CMA security of OTS. This concludes the proof of the claim.

Claim 2

We claim that $\mathcal{A}$ has only negligible advantage in **Game 2**. Assume towards contradiction that $\mathcal{A}$ has non-negligible advantage δ in **Game 2**. We will construct an adversary $\mathcal{A}'$ with advantage δ against the IND-STAG-CCA2 security of TBE. Let $\mathcal{O}'_{\mathsf{Dec}}(\cdot,\cdot)$ be the decryption oracle provided to $\mathcal{A}'$ by the IND-STAG-CCA2 experiment. First notice that the decryption oracles $\mathcal{O}_{\mathsf{Dec1}}$ and $\mathcal{O}_{\mathsf{Dec2}}$ in **Game 2** are identical, we can thus replace them by a single decryption oracle $\mathcal{O}_{\mathsf{Dec}}$. $\mathcal{A}'$ simulates

Adversary $\mathcal{B}$

Input: vk^*
$\tau^* \leftarrow vk^*$
$(pk, sk) \leftarrow \mathsf{TBE.KeyGen}(1^\lambda)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec1}}(sk,\tau^*,\cdot)}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{c}'^* \leftarrow \mathsf{TBE.Enc}(pk, \tau^*, \mathfrak{m}_b)$
$\sigma^* \leftarrow \mathcal{O}_{\mathsf{Sign}}(sgk^*, \mathfrak{c}'^*)$
$\mathfrak{c}^* \leftarrow (\mathfrak{c}'^*, vk^*, \sigma^*)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec2}}(sk,\mathfrak{c}^*,\cdot)}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
return b'

$\mathcal{O}_{\mathsf{Dec1}}(sk, \tau^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
$\tau \leftarrow vk$
If $\tau = \tau^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

$\mathcal{O}_{\mathsf{Dec2}}(sk, \mathfrak{c}^*, \mathfrak{c})$
Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
Parse $\mathfrak{c}^* = (\mathfrak{c}'^*, vk^*, \sigma^*)$
If $(\mathfrak{c}', \sigma) \neq (\mathfrak{c}'^*, \sigma^*)$ and
$\mathsf{OTS.Verify}(vk^*, \mathfrak{c}', \sigma) = 1$
Halt simulation and
return $(\mathfrak{c}', \sigma)$
$\tau \leftarrow vk$
If $\tau = \tau^*$ return $\perp$
If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
return $\mathsf{TBE.Dec}(sk, \tau, \mathfrak{c}')$
Otherwise return $\perp$

Game 2 faithfully, except for the following differences. Instead of generating pk and sk itself, it uses the public key pk provided by the IND-STAG-CCA2 experiment and it uses its own decryption oracle $\mathcal{O}'_{\mathsf{Dec}}$ to implement decryption in the simulation of the decryption oracle $\mathcal{O}_{\mathsf{Dec}}$ for $\mathcal{A}$. Moreover, instead of generating the challenge ciphertext $\mathfrak{c}'^*$ itself, it provides the challenge messages $\mathfrak{m}_0$ and $\mathfrak{m}_1$ to the IND-STAG-CCA2 experiment and uses its own challenge ciphertext $\mathfrak{c}'^*$ to construct the challenge ciphertext for $\mathcal{A}$.

From the view of $\mathcal{A}$, **Game 2** and the simulation of $\mathcal{A}'$ are identically distributed, as in the latter certain computations have merely been transfered to the IND-STAG-CCA2 experiment but remain otherwise identical. Thus, we can conclude that $\mathcal{A}'$'s advantage in the IND-STAG-CCA2 experiment is identical to $\mathcal{A}$'s advantage in **Game 2**, which is non-negligible. Since this contradicts the IND-STAG-CCA2 security of TBE, we can conclude the proof. □

Adversary $\mathcal{A}'$

$\mathcal{A}'(\texttt{init}, 1^\lambda)$
- $(vk^*, sgk^*) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
- $\mathsf{st}_0 \leftarrow (vk^*, sgk^*)$
- $\tau^* \leftarrow vk$
- return (τ^*, st_0)

$\mathcal{A}'(\texttt{find}, \mathsf{st}_0, pk)$
- Parse $\mathsf{st}_0 = (vk^*, sgk^*)$
- $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tau^*, \cdot)}(\texttt{find}, pk)$
- $\mathsf{st}_1 \leftarrow (\mathsf{st}, vk^*, sgk^*)$
- return $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1)$

$\mathcal{A}'(\texttt{guess}, \mathsf{st}_1, \mathfrak{c}'^*)$
- Parse $\mathsf{st}_1 = (\mathsf{st}, vk^*, sgk^*)$
- $\sigma^* \leftarrow \mathsf{OTS.Sign}(sgk^*, \mathfrak{c}'^*)$
- $\mathfrak{c}^* \leftarrow (\mathfrak{c}'^*, vk^*, \sigma^*)$
- $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tau^*, \cdot)}(\texttt{guess}, \mathsf{st}, \mathfrak{c}^*)$
- Return b'.

$\mathcal{O}_{\mathsf{Dec}}(\tau^*, \mathfrak{c})$
- Parse $\mathfrak{c} = (\mathfrak{c}', vk, \sigma)$
- $\tau \leftarrow vk$
- If $\tau = \tau^*$ return $\perp$
- If $\mathsf{OTS.Verify}(vk, \mathfrak{c}', \sigma) = 1$
  - return $\mathcal{O}'_{\mathsf{Dec}}(\tau, \mathfrak{c}')$
- Otherwise return $\perp$

5. IND-CCA2 Secure Public Key Encryption from the McEliece Assumption with small Ciphertext Expansion

Mr. Watson, come here - I want to see you

Alexander Graham Bell to Thomas Watson, first phone call

5.1. Introduction

The McEliece cryptosystem [McE78] was one of the eariliest proposals of a public key cryptosystem. An often mentioned criticism of this scheme are the comparatively large key sizes of size $O(\lambda^2)$. On the other hand, key-generation, encryption and decryption of McEliece schemes are highly parallelizable and can be implemented using low depth binary circuits. Modular exponentiations on the other hand, the cornerstone of all implementations of number theory based encryption schemes, are reluctant to parallelization. However, maybe the most important aspect of this public key cryptosystem is its conjectured post quantum security. While all numbertheoretic hardness assumptions can be broken by efficient quantum algorithms [Sho94], coding based assumptions have resisted quantum cryptanalysis so far, which makes them worthwhile objects of consideration.

In this Chapter we will provide the construction of an IND-CCA2 secure public key encryption scheme with small ciphertext expansion based on the decisional McEliece assumtion and the LPN problem. Following the paradigm of Section 4.4, we will first construct an IND-STAG-CCA2 secure tag-based encryption scheme and then invoke the CHK-transformation (Theorem 4.1). The resulting scheme has essentially the same key sizes, ciphertext sizes and efficiency. The original construction of an IND-CCA2 secure public key encryption scheme based on the McEliece and LPN assumption is due to Dowsley, Müller-Quade and Nascimento [DMQN09]. The

more efficient version presented here is due to Döttling, Dowsley, Müller-Quade and Nascimento [DDMQN12]. The construction loosely follows the correlated products paradigm of Rosen and Segev [RS09]. We will start by providing a short outline of the scheme.

5.1.1. Outline

The basic building block for our scheme is a semantically secure version of the McEliece cryptosystem [NIKM08]. We will first describe an enhanced version of the basic scheme called $\mathsf{PKE}_{\mathsf{McE}}$ which can be decrypted using incomplete secret keys. Using this scheme, we will construct a tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$. Given such a scheme,the CHK-transformation (Construction 4.1) yields an IND-CCA2 secure scheme with essentially the same efficiency.

We will now sketch the *building block* scheme $\mathsf{PKE}_{\mathsf{McE}}$. Given a long message $\mathfrak{m}$ of bitlength $l \cdot n$, we first split up $\mathfrak{m}$ into blocks of size n bits, i.e. $\mathfrak{m} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_l)$. Next, we prepend an extra random block $\mathbf{s}$ and obtain

$$\mathbf{z} = (\mathbf{s}, \mathfrak{m}_1, \ldots, \mathfrak{m}_l).$$

We now interpret $\mathbf{z}$ as a vector in $\mathbb{F}_{2^n}^{l+1}$ and encode $\mathbf{z}$ using a $[k, l+1, k-l]$ Reed Solomon code. This Reed Solomon code acts like the outer code in a concatenated code. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ be the codeword obtained in this way. Next, we interpret each $\mathbf{x}_i$ as binary vector in $\mathbb{F}_2^n$ and encrypt them with a McEliece trapdoor function with public key $\mathbf{A}_i$, i.e. we compute

$$\mathbf{y}_i \leftarrow \mathbf{A}_i \mathbf{x}_i + \mathbf{e}_i$$

for a noise term $\mathbf{e}_i$.

One can also consider this step as encoding $\mathbf{x}_i$ with a scrambled Goppa code and adding noise. Thus, encryption basically encodes a message using a concatenated code with outer Reed Solomon code and inner scrambled Goppa code, and then adds noise. IND-CPA Security of this construction is established as follows. In a first hybrid step, all the McEliece public keys are replaced by random matrices. Given that the decisional McEliece assumption holds, no efficient adversary will notice a difference. In the second step, we will replace all $\mathbf{y}_i$ by purely random values. Given that the decisional LPN problem holds, again no efficient adversary will notice a difference. Such defunct random ciphertexts are independent of the message, thus we have established IND-CPA security.

Now, notice that we don't need to recover all the $\mathbf{x}_i$ to decrypt successfully. As $\mathbf{x}$ is a codeword of a $[k, l+1, k-l]$ Reed Solomon code, we can decode up to $k-l-1$ erasures. This in turn means that $l+1$ of the McEliece secret keys are sufficient to decrypt successfully and obtain $\mathbf{z}$. Even more, given the public keys $pk_1, \ldots, pk_k$, we can *re-encrypt* $\mathbf{z}$ and recover the noise terms $\mathbf{e}_i$ used for encryption. We can therefore check if decryption of the blocks for which we did not possess the secret key would have succeeded or not, by testing if the corresponding noise term has a sufficiently low Hamming weight. Now, if we define decryption of this scheme such that it outputs $\perp$ if decryption of any block fails, then this decryption function can be computed using incomplete secret keys, for which up to $k-l-1$ block keys may be missing.

This feature will be essential in the proof of selective tag chosen ciphertext security of tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$, where the security reduction needs to able to simulate a decryption oracle, while not being in possession of a full secret key.

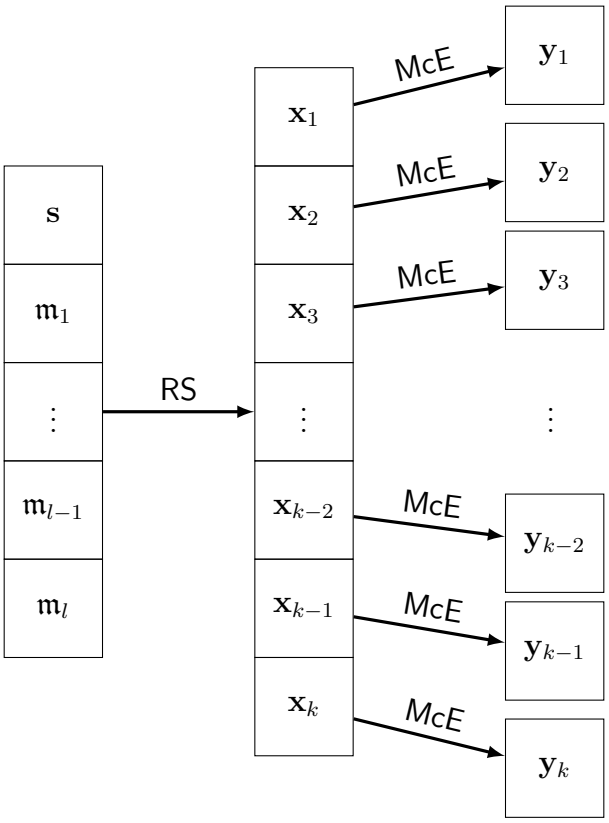

Figure 5.1.: Structure of $\mathsf{PKE}_{\mathsf{McE}}$

5.2. The McEliece Assumption

In this Section, we will provide an overview of the McEliece assumption [McE78].

5.2.1. Search McEliece

As described in Section 3.1.1, at the heart of McEliece's construction are *scrambled* Goppa Codes (c.f. Section 2.5.6). Let C be an irreducible Goppa code of dimension n, length $m = n + l \cdot t$ and designed distance $2t + 1$. Let $\mathbf{G}$ be a generator matrix of C. Moreover, let $(\boldsymbol{\alpha}, g)$ be a description of C that allows efficient decoding of t errors. McEliece's proposal was to scramble the matrix $\mathbf{G}$ by choosing a random permutation matrix $\mathbf{P} \in \mathbb{F}_2^{m \times m}$ and a random invertible matrix $\mathbf{T} \in \mathbb{F}_2^{n \times n}$ and compute $\mathbf{A} \leftarrow \mathbf{PGT}$. Clearly, $\mathbf{A}$ is the generator matrix of an equivalent code $\mathbf{C}'$ of $\mathbf{C}$. The assumption underlying this transformation is that the code C' should be infeasible to decode given the generator matrix $\mathbf{A}$ but *not* its factors $\mathbf{P}$, $\mathbf{G}$ and $\mathbf{T}$. More specifically, the original McEliece assumption is that the function $f_{\mathbf{A}}$ given by

$$f_{\mathbf{A}}(\mathbf{x}, \mathbf{e}) = \mathbf{As} + \mathbf{e}$$

is one-way, if $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$ is chosen uniformly at random and $\mathbf{e} \leftarrow_{\$} \mathsf{S}_m(t)$ is chosen uniformly of weight t. As the code C' is equivalent to C, it has the same minimum distance as C, which is at least $2t + 1$. Thus, $(\mathbf{s}, \mathbf{e})$ are information theoretically uniquely defined given $\mathbf{A}$ and $\mathbf{y} = f_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$. On the other hand, if we know the the factors $\mathbf{P}$, $\mathbf{T}$ and $(\boldsymbol{\alpha}, g)$, then the problem of decoding C' is easy. Let

$$\mathbf{y} = \mathbf{As} + \mathbf{e}.$$

As $\mathbf{A} = \mathbf{PGT}$ we can rewrite this as

$$\mathbf{y} = \mathbf{PGTs} + \mathbf{e}.$$

As $\mathbf{P}$ is a permutation matrix it holds that $\mathbf{P}^T\mathbf{P} = \mathbf{I}$. This yields

$$\mathbf{P}^T\mathbf{y} = \mathbf{GTs} + \mathbf{P}^T\mathbf{e}.$$

As $\mathbf{P}^T$ is also a permutation matrix, it holds that $\mathsf{wgt}(\mathbf{P}^T\mathbf{e}) = \mathsf{wgt}(\mathbf{e}) = t$. Thus, we have reduced the problem of decoding the code C' to decoding the code C. To recover $\mathbf{s}' = \mathbf{Ts}$ we can use the efficient decoder of $\mathsf{C} = \Gamma(\boldsymbol{\alpha}, g)$ to obtain

$$\mathbf{s}' = \mathsf{C.Decode}(\mathbf{P}^T\mathbf{y})$$

from which we can compute $\mathbf{s}$ by $\mathbf{s} = \mathbf{T}^{-1}\mathbf{s}'$. Thus, the original McEliece assumption [McE78] conjectures that $f_{\mathbf{A}}$ is an injective trapdoor function with trapdoor $(\mathbf{P}, \mathbf{T}, \boldsymbol{\alpha}, g)$, if $\mathbf{A}$ is generated by $\mathbf{A} \leftarrow \mathbf{PGT}$.

5.2.2. Decisional McEliece

For our purposes however, we will need a slightly different assumption. McEliece [McE78] assumed that scrambled Goppa codes are hard to decode as they meet the Gilbert-Varshamov bound (Theorem 2.4) and there is further no efficient way of telling them apart from random matrices. Thus, one might as well conjecture that scrambled Goppa codes are actually pseudorandom, i.e. indistinguishable from random codes [CFS01, NIKM08]. This is exactly the decisional McEliece assumption. Assume there exists an efficient algorithm $\mathsf{SampleGoppa}$, which, given parameters ℓ and t samples a the description $(\boldsymbol{\alpha}, g)$ of an irreducible $[m, n, 2t+1]$ Goppa code. Some care must be taken when sampling $(\boldsymbol{\alpha}, g)$, as we explain in the next paragraph.

Construction 5.1. *Let λ be a security parameter. Let $\ell, t = \mathsf{poly}(\lambda)$ be positive integers, let $m = 2^\ell$ and $n = m - \ell \cdot t$. Let $\mathsf{Perm}(m) \subseteq \mathbb{F}_2^{m \times m}$ be the group of $m \times m$ permutation matrices and $\mathsf{GL}_n(\mathbb{F}_2) \subseteq \mathbb{F}_2^{n \times n}$ be the group of invertible $n \times n$ matrices over $\mathbb{F}_2$. Assume that the decoder $\mathsf{C.Decode}$ outputs an information word $\mathbf{s}$ and an error vector $\mathbf{e}$. Define the algorithms $\mathsf{McE.Gen}$ and $\mathsf{McE.Decode}$ as follows.*

- $\mathsf{McE.Gen}(1^\lambda)$*:*
  $(\boldsymbol{\alpha}, g) \leftarrow \mathsf{SampleGoppa}(\ell, t)$
  $\mathsf{C} \leftarrow \Gamma(\boldsymbol{\alpha}, g)$
  $\mathbf{G} \leftarrow \mathsf{C.Generator}()$
  $\mathbf{P} \leftarrow_\$ \mathsf{Perm}(m)$
  $\mathbf{T} \leftarrow_\$ \mathsf{GL}_n(\mathbb{F}_2)$
  $\mathbf{A} \leftarrow \mathbf{PGT}$
  $\mathsf{td} \leftarrow (\mathbf{P}, \mathbf{T}, \boldsymbol{\alpha}, g)$
  Return $(\mathbf{A}, \mathsf{td})$

- $\mathsf{McE.Decode}(\mathsf{td}, \mathbf{y})$*:*
  Parse $\mathsf{td} = (\mathbf{P}, \mathbf{T}, \boldsymbol{\alpha}, g)$
  $\mathsf{C} \leftarrow \Gamma(\boldsymbol{\alpha}, g)$
  $\mathbf{y}' \leftarrow \mathbf{P}^T\mathbf{y}$
  $(\mathbf{s}', \mathbf{e}) \leftarrow \mathsf{C.Decode}(\mathbf{y}')$
  If $\mathbf{s}' = \perp$ *or* $\mathsf{wgt}(\mathbf{e}) > t$
  Return $\perp$
  $\mathbf{s} \leftarrow \mathbf{T}^{-1}\mathbf{s}'$
  Return $\mathbf{s}$

The correctness of the decoding algorithm $\mathsf{McE.Dec}$ follows as above. We remark at this point that we will use $\mathsf{McE.Gen}$ and $\mathsf{McE.Decode}$ in a rather axiomatic way. More specifically, we will not use any particularities of Goppa codes and just assume that $\mathsf{McE.Decode}$ decodes up to t errors but outputs $\perp$ if more than t errors occur.

We will now state the decisional McEliece problem.

Problem 5.1 (Decisional McEliece Problem)**.** *Let λ be a security parameter. Let $m, n, \ell, t = \mathsf{poly}(\lambda)$. Let $(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{McE.Gen}(1^\lambda)$ and let $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{m \times n}$ be chosen uniformly at random. The goal of the decisional McEliece problem is to distinguish the distributions $\mathbf{A}$ and $\mathbf{U}$.*

Assumption 5.1 (The McEliece assumption)**.** *Every PPT distinguisher $\mathcal{D}$ has at most negligible advantage distinguishing problem 5.1, i.e.*

$$\mathsf{Adv}_{\mathsf{McE}}(\mathcal{D}) = |\Pr[\mathcal{D}(\mathbf{A}) = 1] - \Pr[\mathcal{D}(\mathbf{U}) = 1]| \leq \mathsf{negl}(\lambda).$$

where $\mathbf{A}$ is generated by $(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{McE.Gen}(1^\lambda)$ and $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{m \times n}$ is chosen uniformly at random.

Notice now that the original McEliece assumption, i.e. the one-wayness of the trapdoor function

$$f_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$$

follows from the McEliece assumption and the hardness of $\mathsf{LPN}(n, m, \mathsf{S}_m(t))$: In a first hybrid step we can replace the matrix $\mathbf{A}$ by a uniformly chosen matrix $\mathbf{U}$. Any adversary that notices a will help distinguishing the McEliece distribution from uniformly random. After this transformation, inverting $f_{\mathbf{A}}$ is identical to solving $\mathsf{LPN}(n, m, \mathsf{S}_m(t))$. Thus, an adversary against the one-wayness of $f_{\mathbf{A}}$ can be used to either solve the McEliece problem or the LPN problem $\mathsf{LPN}(n, m, \mathsf{S}_m(t))$, contradicting the hardness of one of them.

5.2.3. Attacks and Variants

There are very few structural attacks against the McEliece assumption. If the matrix $\mathbf{G}$ is known, then one can target the permutation $\mathbf{P}$. If both $\mathbf{P}$ and $\mathbf{G}$ are known for a given $\mathbf{A}$, then the matrix $\mathbf{T}$ can be found using linear algebra, i.e. we can solve

$$\mathbf{A} = \mathbf{PGT}$$

for $\mathbf{T}$ using e.g. gaussian elimination. The best known algorithm to recover $\mathbf{P}$ from $\mathbf{A} = \mathbf{PGT}$ and $\mathbf{G}$ is Sendrier's support splitting algorithm [Sen00]. This algorithm's runtime is exponential in the dimension of the *hull* of $\mathsf{C} = \mathsf{C}(\mathbf{G})$, which is the intersection of C and its dual $\mathsf{C}^\perp$.

Loidreau and Sendrier [LS01] showed that there exists bad choices for the Goppa code $\mathsf{C} = \Gamma(\boldsymbol{\alpha}, g)$, i.e. $\boldsymbol{\alpha}, g$ for which the McEliece problem is easier than usual. In particular, whenever the polynomial $g \in \mathbb{F}_{2^\ell}[X]$ has binary coefficients then a combination of exhaustive search through all $g \in \mathbb{F}_2[X]$ of degree t and the support splitting algorithm will find both $\mathbf{P}$ and $\mathbf{G}$. Thus, binary polynomials g should be avoided when generating the Goppa code. There are, however, efficient distinguishing attacks when the chosen class of Goppa codes is more restricted. Faugère et al. [FOPT10] demonstrated an algebraic attack against the McEliece problem which leads to an efficient distinguishing attack for *quasi-cyclic* and *dyadic* Goppa codes. In a follow up work Faugère et al. [FGUO$^+$13] showed that if the rate of the

irreducible Goppa code $\mathsf{C} = \Gamma(\boldsymbol{\alpha}, g)$ is chosen too close to 1, then there also exists an efficient distinguisher for the McEliece problem. Very recently, Couvreur et al. [COT14] provided an efficient distinguisher against the McEliece problem with so called *wild Goppa codes*.

On the other hand, there are no structural attacks known against the original McEliece assumption and it seems that brute forcing the trapdoor $(\boldsymbol{\alpha}, g)$ and applying support splitting is thus far the most efficient attack. As a consequence, actual attacks against McEliece cryptosystems usually target the LPN part (c.f. Section 3.6).

Several alternatives for irreducible Goppa codes have been proposed as a trapdoor for McEliece. Among these proposals were generalized Reed Solomon codes [Nie85], Reed-Muller codes [Sid94], algebraic geometric codes [JM96] and quasi-cyclic codes [Gab05, BCGM07]. However, most of them were completely broken: generalized Reed Solomon codes [SS92], Reed Muller codes [MS07], algebraic geometric codes [FM08] and quasi-cyclic codes [OTD10]. It was shown that using low density parity check codes is neither a good idea [MRS00].

5.3. The Building Block IND-CPA Scheme

We will now provide the elementary building block for our IND-CCA2 secure scheme: An IND-CPA secure encryption scheme from the McEliece and LPN assumptions which will serve as the basis for the construction of a tag-based encryption scheme in the next Section. The scheme is basically an upscaled version of the basic McEliece scheme. Instead of encrypting a single message block, we encrypt l message blocks $\mathfrak{m}_1, \ldots, \mathfrak{m}_l$.

To facilitate the construction of the tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$ in the next section, we will construct $\mathsf{PKE}_{\mathsf{McE}}$ directly such that it admits decryption using incomplete secret keys. Therefore, the plaintexts will be encoded using an erasure correcting code. Any blocks for which the corresponding secret key is missing will be later reconstructed using erasure correction. As we are using message blocks of large size, Reed Solomon codes are the optimal choice for the erasure correcting code.

Construction 5.2. *Let λ be a security parameter. Let $m, n, t = \mathsf{poly}(\lambda)$ be parameters for which binary $[m, n, 2t+1]$ Goppa codes exist. Let $l, k = \mathsf{poly}(\lambda)$ be positive integers with $k > l$. Let $\rho = \rho(\lambda) \in (0, 1)$. Let RS be a $[k, l+1, k-l]$ Reed Solomon Code over $\mathbb{F}_{2^n}$. The public key encryption scheme $\mathsf{PKE}_{\mathsf{McE}} = (\mathsf{PKE}_{\mathsf{McE}}.\mathsf{KeyGen}, \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}, \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec})$ is specified as follows.*

- $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{KeyGen}(1^\lambda)$*:*
  For $i = 1, \ldots k$
  $(\mathbf{A}_i, \mathsf{td}_i) \leftarrow \mathsf{McE}.\mathsf{Gen}(1^\lambda)$
  $pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
  $sk \leftarrow (\mathsf{td}_1, \ldots, \mathsf{td}_k)$
  Return (pk, sk)

- $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}(pk, \mathfrak{m})$*:*
  Parse $pk = (\mathbf{A}_1, \ldots, \mathbf{A}_k) \in (\mathbb{F}_2^{m \times n})^k$ *and* $\mathfrak{m} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_l) \in (\mathbb{F}_2^n)^l$
  $\boldsymbol{s} \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1, \ldots, \mathfrak{m}_l)$
  $\mathbf{x} \leftarrow \mathsf{RS}.\mathsf{Encode}(\mathbf{z})$

Parse $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$
For $i = 1, \ldots, k$
$\mathbf{e}_i \leftarrow_{\$} \mathsf{Ber}(m, \rho)$
$\mathbf{y}_i \leftarrow \mathbf{A}_i \cdot \mathbf{x}_i + \mathbf{e}_i$
$\mathfrak{c} \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
Return $\mathfrak{c}$

- $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c})$*:*
  Parse $sk = (\mathsf{td}_1, \ldots, \mathsf{td}_k)$ *and* $\mathfrak{c} = (\mathbf{y}_1, \ldots, \mathbf{y}_k) \in (\mathbb{F}_2^m)^k$
  For $i = 1, \ldots, k$
  $\mathbf{x}_i \leftarrow \mathsf{McE.Decode}(\mathsf{td}_i, \mathbf{y}_i)$
  If $\mathbf{x}_i = \perp$
  Return $\perp$
  $\mathbf{x} \leftarrow (\mathbf{x}_1, \ldots, \mathbf{x}_k)$
  $\mathbf{z} \leftarrow \mathsf{RS.Decode}(\mathbf{x})$
  If $\mathbf{z} = \perp$
  Return $\perp$
  Parse $\mathbf{z} = (\mathbf{s}, \mathfrak{m}_1, \ldots, \mathfrak{m}_l)$
  $\mathfrak{m} \leftarrow (\mathfrak{m}_1, \ldots, \mathfrak{m}_l)$
  Return $\mathfrak{m}$

The plaintext space of $\mathsf{PKE}_{\mathsf{McE}}$ *is* $\mathfrak{M}_{\mathsf{McE},\lambda} = \mathbb{F}_2^{l \cdot n}$ *and the ciphertext space is* $\mathfrak{C}_{\mathsf{McE},\lambda} = \mathbb{F}_2^{k \cdot m}$.

5.3.1. Completeness

We will first show that the encryption scheme $\mathsf{PKE}_{\mathsf{McE}}$ is complete.

Lemma 5.1. *The scheme* $\mathsf{PKE}_{\mathsf{McE}}$ *is complete, if* $(1 + \beta)\rho m \leq t$ *for a constant* $\beta > 0$.

Proof. Let $\mathfrak{c} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$ be a ciphertext generated by $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}$, i.e. each $\mathbf{y}_i$ is of the form $\mathbf{A}_i\mathbf{x}_i + \mathbf{e}_i$. Since each error vector $\mathbf{e}_i$ is chosen from the Bernoulli distribution $\mathsf{Ber}(m, \rho)$, the Chernoff bound (Theorem 2.1) yields that

$$\Pr[\mathsf{wgt}(\mathbf{e}_i) \geq t] \leq \Pr[\mathsf{wgt}(\mathbf{e}_i) \geq (1 + \beta)\rho m] < e^{-\beta^2 \rho m/2}.$$

Given that $\mathsf{wgt}(\mathbf{e}_i) \leq t$, the McEliece decoder $\mathsf{McE.Decode}(\mathsf{td}_i, \cdot)$ correctly recovers $\mathbf{x}_i$ from $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i + \mathbf{e}_i$. By the union bound it holds that

$$\Pr[\exists i : \mathsf{McE.Decode}(\mathsf{td}_i, \mathbf{y}_i) \neq \mathbf{x}_i] \leq \sum_{i=1}^{k} \Pr[\mathsf{McE.Decode}(\mathsf{td}_i, \mathbf{y}_i) \neq \mathbf{x}_i] < k \cdot e^{-\beta^2 \rho m/2},$$

which is negligible in λ. Thus, the vector $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is recovered correctly by the decryption algorithm, except with probability $k \cdot e^{-\beta^2 \rho m/2}$. If the vector $\mathbf{x}$ is correctly recovered, the Reed Solomon decoder $\mathsf{RS.Decode}$ will also recover $\mathbf{z} = (\mathbf{s}, \mathfrak{m}_1, \ldots, \mathfrak{m}_l)$. We can conclude that $\Pr[\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}(pk, \mathfrak{m})) \neq \mathfrak{m}] < \mathsf{negl}(\lambda)$. □

5.3.2. IND-CPA Security

We will now prove that the scheme $\mathsf{PKE}_{\mathsf{McE}}$ is IND-CPA secure.

Theorem 5.1. *The scheme $\mathsf{PKE}_{\mathsf{McE}}$ is IND-CPA secure, given that the decisional McEliece assumption (Assumption 5.1) and the $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$ assumption hold.*

Proof. Let $\mathcal{A}$ be a PPT-adversary against the IND-CPA security of $\mathsf{PKE}_{\mathsf{McE}}$. Consider the following experiments **Game 0**, **Game 1** and **Game 2**.

Game 0

For $i = 1, \ldots, k$:
  $(\mathbf{A}_i, \mathsf{td}_i) \leftarrow \mathsf{McE.Gen}(1^\lambda)$
$pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
For $i = 1, \ldots, k$:
  $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}_i \leftarrow \mathbf{A}_i \mathbf{x}_i + \mathbf{e}_i$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
$b' \leftarrow \mathcal{A}(\mathsf{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Game 1

For $i = 1, \ldots, k$:
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
For $i = 1, \ldots, k$:
  $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  $\mathbf{y}_i \leftarrow \mathbf{A}_i \mathbf{x}_i + \mathbf{e}_i$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
$b' \leftarrow \mathcal{A}(\mathsf{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Game 2

For $i = 1, \ldots, k$:
  $\mathbf{A}_i \leftarrow_\$ \mathbb{F}_2^{m \times n}$
$pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
For $i = 1, \ldots, k$:
  $\mathbf{y}_i \leftarrow_\$ \mathbb{F}_2^m$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
$b' \leftarrow \mathcal{A}(\mathsf{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Clearly, **Game 0** is the IND-CPA experiment for $\mathsf{PKE}_{\mathsf{McE}}$, for which we have subsituted $\mathsf{PKE.KeyGen}$ and $\mathsf{PKE.Enc}$ with their implementations according to $\mathsf{PKE}_{\mathsf{McE}}$. **Game 1** is identical to **Game 0**, except that the matrices $\mathbf{A}_i$ are not generated by the McEliece generation function $\mathsf{McE.Gen}$ but chosen uniformly at random. **Game 2** is identical to **Game 1**, except that the vectors $\mathbf{y}_i$ are not computed by $\mathbf{y}_i \leftarrow \mathbf{A}_i\mathbf{x}_i + \mathbf{e}_i$, but chosen uniformly at random. In **Game 2** the challenge ciphertext $\mathfrak{c}^*$ is independent of the challenge message $\mathfrak{m}^*$ and consequently the advantage of $\mathcal{A}$ in **Game 2** is 0. Thus, to prove IND-CPA security, it remains to show that from the view of $\mathcal{A}$ **Game 0** and **Game 1** as well as **Game 1** and **Game 2** are indistinguishable.

Claim 1

We claim that **Game 0** and **Game 1** are computationally indistinguishable from the view of $\mathcal{A}$, given that the decisional McEliece assumption (Assumption 5.1) holds. Assume towards contradiction that $\mathcal{A}$ distinguishes between **Game 0** and **Game 1** with non-negligible advantage ϵ, i.e.

$$\Pr[\mathbf{Game0}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1] \geq \epsilon.$$

We will construct a distinguisher $\mathcal{D}_1$ that distinguishes the decisional McEliece problem with advantage ϵ/k. We will first provide the distinguisher $\mathcal{D}_1$.

Distinguisher $\mathcal{D}_1$
- Input $\mathbf{A} \in \mathbb{F}_2^{m \times n}$
- $i^* \leftarrow_\$ \{1, \ldots, k\}$
- For $j = 1, \ldots, i^* - 1$
  - $\mathbf{A}_j \leftarrow_\$ \mathbb{F}_2^{m \times n}$
- $\mathbf{A}_{i^*} \leftarrow \mathbf{A}$
- For $j = i^* + 1, \ldots, k$
  - $(\mathbf{A}_j, \mathsf{td}_j) \leftarrow \mathsf{McE.Gen}(1^\lambda)$
- $pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
- $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
- $b \leftarrow_\$ \{0, 1\}$
- $\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
- Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
- $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
- $\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
- For $i = 1, \ldots, k$:
  - $\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
  - $\mathbf{y}_i \leftarrow \mathbf{A}_i\mathbf{x}_i + \mathbf{e}_i$
- $\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
- $b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
- Return 1 iff $b = b'$.

Before we analyze the distinguishing advantage of $\mathcal{D}_1$, we will define $k + 1$ hybrid experiments $\mathsf{H}_0, \ldots, \mathsf{H}_k$. Experiment H_i is given as follows.

Clearly, experiment H_0 is identical to **Game 1** while experiment H_k is identical to **Game 2**. We will now analyze the distinguishing advantage of $\mathcal{D}_1$. First assume

Experiment H_i
For $j = 1, \ldots, i$
$\mathbf{A}_j \leftarrow_\$ \mathbb{F}_2^{m \times n}$
For $j = i+1, \ldots, k$
$(\mathbf{A}_j, \mathsf{td}_j) \leftarrow \mathsf{McE.Gen}(1^\lambda)$
$pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$, $\mathbf{z} \leftarrow (\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
For $i = 1, \ldots, k$:
$\mathbf{e}_i \leftarrow_\$ \mathsf{Ber}(m, \rho)$
$\mathbf{y}_i \leftarrow \mathbf{A}_i \mathbf{x}_i + \mathbf{e}_i$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

that $\mathcal{D}_1$'s input $\mathbf{A}$ is generated by the McEliece generation, i.e. $\mathbf{A} = \mathbf{A}_{\mathsf{McE}}$ where $(\mathbf{A}_{\mathsf{McE}}, \mathsf{td}) \leftarrow \mathsf{McE.Gen}(1^\lambda)$. Fix the random choice i^* to $i^* = i$. Then it holds that from the view of $\mathcal{A}$ the simulation of $\mathcal{D}_1$ is identically distributed to experiment H_{i-1}. Therefore, we have that

$$\Pr[\mathcal{D}_1(\mathbf{A}_{\mathsf{McE}}) = 1 | i^* = i] = \Pr[\mathsf{H}_{i-1}(\mathcal{A}) = 1].$$

Consequently, it holds that

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{A}_{\mathsf{McE}}) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathcal{D}_1(\mathbf{A}_{\mathsf{McE}}) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}_{i-1}(\mathcal{A}) = 1].\end{aligned}$$

Now assume that $\mathcal{D}_1$'s input $\mathbf{A}$ is chosen uniformly at random, i.e. $\mathbf{A} = \mathbf{U}$ where $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{m \times n}$. Again, fix the random choice i^* to $i^* = i$. Then it holds that from the view of $\mathcal{A}$, the simulation of $\mathcal{D}_1$ is identically distributed to H_i. Therefore, we have

$$\Pr[\mathcal{D}_1(\mathbf{U}) = 1 | i^* = i] = \Pr[\mathsf{H}_i(\mathcal{A}) = 1].$$

Consequently, it holds that

$$\begin{aligned}\Pr[\mathcal{D}_1(\mathbf{U}) = 1] &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathcal{D}_1(\mathbf{U}) = 1 | i^* = i] \\ &= \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}_i(\mathcal{A}) = 1].\end{aligned}$$

Together this yields

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}_1) &= \Pr[\mathcal{D}_1(\mathbf{A}_{\mathsf{McE}}) = 1] - \Pr[\mathcal{D}_1(\mathbf{U}) = 1] \\ &= \left|\sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}_{i-1}(\mathcal{A}) = 1] - \sum_{i=1}^{k} \frac{1}{k} \Pr[\mathsf{H}_i(\mathcal{A}) = 1]\right| \\ &= \frac{1}{k} |\Pr[\mathsf{H}_0(\mathcal{A}) = 1] - \Pr[\mathsf{H}_k(\mathcal{A}) = 1]| \\ &= \frac{1}{k} |\Pr[\mathbf{Game0}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1]| \\ &\geq \frac{\epsilon}{k}.\end{aligned}$$

This however contradicts the decisional McEliece assumption which concludes the claim.

Claim 2

We claim that **Game 1** and **Game 2** are computationally indistinguishable from the view of $\mathcal{A}$, given that the $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$ assumption holds. Assume towards contradiction that $\mathcal{A}$ distinguishes between **Game 1** and **Game 2** with advantage ϵ, i.e.

$$\Pr[\mathbf{Game1}(\mathcal{A}) = 1] - \Pr[\mathbf{Game2}(\mathcal{A}) = 1] \geq \epsilon.$$

We will construct a distinguisher $\mathcal{D}_2$ that distinguishes the $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$ problem with advantage ϵ, contradicting the hardness of $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$. We will now provide the distinguisher $\mathcal{D}_2$.

Distinguisher $\mathcal{D}_2$
Input $(\mathbf{A}, \mathbf{v}) \in \mathbb{F}_2^{k \cdot m \times n} \times \mathbb{F}_2^{k \cdot m}$
Parse $\mathbf{A} = (\mathbf{A}_1^T \| \ldots \| \mathbf{A}_k^T)^T$ and $\mathbf{v} = (\mathbf{v}_1^T \| \ldots \| \mathbf{v}_k^T)^T$
$pk \leftarrow (\mathbf{A}_1, \ldots, \mathbf{A}_k)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_{\$} \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
Parse $\mathfrak{m}^* = (\mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{z} \leftarrow (0, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*)$
$\mathbf{x} \leftarrow \mathsf{RS.Encode}(\mathbf{x})$
For $i = 1, \ldots, k$:
    $\mathbf{y}_i \leftarrow \mathbf{v}_i + \mathbf{A}_i \mathbf{x}_i$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

We will now analyze the distinguishing advantage of $\mathcal{D}_2$. Assume first that $\mathcal{D}_2$'s input $(\mathbf{A}, \mathbf{v}$ is of the form $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, where $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$ is chosen uniformly at random and $\mathbf{e} \leftarrow_{\$} \mathsf{Ber}(k \cdot m, \rho)$. Then the $\mathbf{v}_i$ are of the form $\mathbf{v}_i = \mathbf{A}_i \mathbf{s} + \mathbf{e}_i$. Thus $\mathbf{y}_i = \mathbf{v}_i + \mathbf{A}_i \mathbf{x}_i = \mathbf{A}_i \mathbf{s} + \mathbf{e}_i + \mathbf{A}_i \mathbf{x}_i = \mathbf{A}_i(\mathbf{s} + \mathbf{x}_i) + \mathbf{e}_i$. It holds for all $i \in \{1, \ldots, k\}$

that $\mathsf{RS.Encode}_i(\mathbf{s}, 0, \ldots, 0) = \mathbf{s}$, as this input corresponds to a constant polynomial. Thus we have

$$\begin{aligned}\mathbf{s} + \mathbf{x}_i &= \mathsf{RS.Encode}_i(\mathbf{s}, 0, \ldots, 0) + \mathsf{RS.Encode}_i(0, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*) \\ &= \mathsf{RS.Encode}_i(\mathbf{s}, \mathfrak{m}_1^*, \ldots, \mathfrak{m}_l^*).\end{aligned}$$

We conclude that the $\mathbf{y}_i$ are distributed as in **Game 1** and thus $\mathcal{A}$'s view is distributed as in **Game 1**. On the other hand, if $\mathcal{D}_2$'s input $(\mathbf{A}, \mathbf{v})$ is of the form $(\mathbf{A}, \mathbf{u})$, with a uniformly random $\mathbf{u}$, then the $\mathbf{y}_i = \mathbf{v}_i + \mathbf{A}_i\mathbf{x}_i = \mathbf{u}_i + \mathbf{A}_i\mathbf{x}_i$ are also distributed uniformly random. Thus the $\mathbf{y}_i$ are distributed as in **Game 2** and consequently $\mathcal{A}$'s view is distributed as in **Game 2**. We get that

$$\begin{aligned}\mathsf{Adv}(\mathcal{D}_2) &= \Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{As} + \mathbf{e}) = 1] - \Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{u}) = 1] \\ &= \Pr[\mathbf{Game1}(\mathcal{A}) = 1] - \Pr[\mathbf{Game2}(\mathcal{A}) = 1] \\ &\geq \epsilon,\end{aligned}$$

contradicting the hardness of $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$. □

5.3.3. Alternative Decryption

We will now construct an alternative decryption algorithm Dec^* for $\mathsf{PKE}_{\mathsf{McE}}$ that can decrypt ciphertexts using *incomplete* secret keys. We will use this alternative decryption algorithm in the proof of Theorem 5.2 in order to simulate a decryption oracle using incomplete secret keys. Dec^* will correctly decrypt ciphertexts $\mathfrak{c}$ given the public key pk and an incomplete secret key $\tilde{sk}$ of which at most $\mathsf{d}(\mathsf{RS}) - 1 = k-l-1$ components are missing. We need the behavior of Dec^* to be identical to the behavior of $\mathsf{PKE}_{\mathsf{McE}}$. To this end $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}$ has been crafted in a way such that it rejects a ciphertext $\mathfrak{c}$ if Dec^* would not be able to decrypt it. The construction of Dec^* follows a simple idea. Dec^* first decrypts all the components for which it has the secret keys and then uses erasure correction to reconstruct the other components. Finally, it re-encrypts the components and compares them with the input ciphertext to ensure that $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}$ would not have rejected any of them.

Construction 5.3. *Let λ be a security parameter and let m, n, l, k, ρ, t be as in the definition of $\mathsf{PKE}_{\mathsf{McE}}$ (Construction 5.2). The alternative decryption algorithm Dec^* for $\mathsf{PKE}_{\mathsf{McE}}$ is defined as follows.*

$\mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$*:*
  Parse $pk = (\mathbf{A}_1, \ldots, \mathbf{A}_k)$, $\tilde{sk} = (\mathsf{td}_1, \ldots, \mathsf{td}_k)$ *and* $\mathfrak{c} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$
  For $i = 1, \ldots, k$
    If $\mathsf{td}_i \neq \perp$
      $\mathbf{x}_i \leftarrow \mathsf{McE.Decode}(\mathsf{td}_i, \mathbf{y}_i)$
    Otherwise
      $\mathbf{x}_i \leftarrow \perp$
  $\mathbf{x} \leftarrow (\mathbf{x}_1, \ldots, \mathbf{x}_k)$
  $\mathbf{z} \leftarrow \mathsf{RS.Decode}(\mathbf{x})$
  If $\mathbf{z} = \perp$
    Return $\perp$
  $\mathbf{x}' \leftarrow \mathsf{RS.Encode}(\mathbf{z})$
  Parse $\mathbf{x}' = (\mathbf{x}'_1, \ldots, \mathbf{x}'_k)$
  For $i = 1, \ldots, k$

$\mathbf{e}_i \leftarrow \mathbf{y}_i - \mathbf{A}_i\mathbf{x}'_i$
If $\mathsf{wgt}(\mathbf{e}_i) > t$
Return $\perp$
Parse $\mathbf{z} = (\mathbf{s}, \mathfrak{m}_1, \ldots, \mathfrak{m}_l)$
$\mathfrak{m} \leftarrow (\mathfrak{m}_1, \ldots, \mathfrak{m}_l)$
Return $\mathfrak{m}$

Lemma 5.2. *Fix a pair (pk, sk) of public and secret keys generated by the key generation algorithm $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{KeyGen}$. Fix an $\tilde{sk}$ such that $\tilde{sk}$ contains at most $k-l-1$ erasures but is otherwise identical to sk. Then it holds for every $\mathfrak{c} \in \mathbb{F}_2^{m \cdot k}$ (i.e. every $\mathfrak{c}$ that could syntactically be a ciphertext) that*

$$\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c}) = \mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c}).$$

Proof. Let $pk = (\mathbf{A}_1, \ldots, \mathbf{A}_k)$ and $\mathfrak{c} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$. Let $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i^\dagger + \mathbf{e}_i^\dagger$ with $\mathbf{x}_i^\dagger \in \mathbb{F}_2^n$ and $\mathbf{e}_i^\dagger \in \mathbb{F}_2^m$ such that $\mathsf{wgt}(\mathbf{e}_i^\dagger)$ is as small as possible. We will distinguish two cases:

1. It holds for all i that $\mathsf{wgt}(\mathbf{e}_i^\dagger) \leq t$.
2. There exists an i such that $\mathsf{wgt}(\mathbf{e}_i^\dagger) > t$.

We will now show that in both cases it holds that

$$\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c}) = \mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c}).$$

We will start with the first case. Observe that in this case $\mathbf{x}_i^\dagger$ and $\mathbf{e}_i^\dagger$ are uniquely defined by $\mathbf{A}_i$ and $\mathbf{y}_i$, as $\mathbf{A}_i$ generates an error correcting code of distance $\geq 2t+1$ and $\mathsf{wgt}(\mathbf{e}_i^\dagger) \leq t$. As $\mathsf{McE}.\mathsf{Decode}(\mathsf{td}_i, \cdot)$ corrects up to t errors, it will output $\mathbf{x}_i = \mathbf{x}_i^\dagger$. Consequently, if $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is a codeword of RS, then $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c})$ will produce a unique output $\mathfrak{m} \neq \perp$. Now consider the alternative decoding algorithm Dec^*. There are at most $k-l-1$ erasures in $\tilde{sk}$. For all other i, $\mathsf{McE}.\mathsf{Decode}(\mathsf{td}_i, \mathbf{y}_i)$ will output $\mathbf{x}_i = \mathbf{x}_i^\dagger$, as $\mathbf{e}_i^\dagger$ is within the decoding bound for the McEliece decoder. Consequently, Dec^* recovers at least $l+1$ components $\mathbf{x}_i^\dagger \neq \perp$ of the vector $\mathbf{x}^\dagger$. Thus, $\mathbf{x}$ has at most $k-l-1$ erasures and the Reed-Solomon Decoder $\mathsf{RS}.\mathsf{Decode}(\mathbf{x})$ will produce the same output $\mathbf{z}$ as in the erasure-free case. Moreover, if $\mathbf{z} \neq \perp$ then $\mathbf{x}' = \mathsf{RS}.\mathsf{Encode}(\mathbf{z}) = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is identical to $\mathbf{x}^\dagger = (\mathbf{x}_1^\dagger, \ldots, \mathbf{x}_k^\dagger)$. Therefore, it holds for all i that $\mathbf{e}_i = \mathbf{y}_i - \mathbf{A}_i\mathbf{x}'_i = \mathbf{A}_i\mathbf{x}_i^\dagger + \mathbf{e}_i^\dagger - \mathbf{A}_i\mathbf{x}_i = \mathbf{e}_i^\dagger$ and the checks $\mathsf{wgt}(\mathbf{e}_i) \leq t$ will be passed as $\mathsf{wgt}(\mathbf{e}_i^\dagger) \leq t$. Thus the unique output $\mathfrak{m} \neq \perp$ is produced. Consequently, in this case we have $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c}) = \mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$.

In the second case there exists an index i such that $\mathsf{wgt}(\mathbf{e}_i^\dagger) > t$. For this particular index i $\mathsf{McE}.\mathsf{Decode}(\mathsf{td}_i, \mathbf{y}_i)$ will output $\perp$ by definition. Thus, $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c}) = \perp$. Now consider $\mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$. It computes a vector $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$. If $\mathbf{x}$ has more than $k-l-1$ erasures, then $\mathsf{RS}.\mathsf{Decode}(\mathbf{x})$ outputs $\perp$ and thus $\mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$ outputs $\perp$. If however $\mathsf{RS}.\mathsf{Decode}(\mathbf{x})$ outputs a $\mathbf{z} \neq \perp$, then $\mathbf{x}' = \mathsf{RS}.\mathsf{Encode}(\mathbf{z})$ is identical to $\mathbf{x}^\dagger$. Thus we have $\mathbf{e}_i = \mathbf{y}_i - \mathbf{A}_i\mathbf{x}'_i = \mathbf{A}_i\mathbf{x}_i^\dagger + \mathbf{e}_i^\dagger - \mathbf{A}_i\mathbf{x}_i = \mathbf{e}_i^\dagger$ and the check $\mathsf{wgt}(\mathbf{e}_i) \leq t$ is not passed as $\mathsf{wgt}(\mathbf{e}_i) > t$. Thus $\mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$ outputs $\perp$ and we conclude that also in this case it holds that $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \mathfrak{c}) = \mathsf{Dec}^*(pk, \tilde{sk}, \mathfrak{c})$. □

5.4. Tag-based Encryption from McEliece

We will now construct a tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$ from the public key encryption scheme $\mathsf{PKE}_{\mathsf{McE}}$ provided in the last Section. We will first outline the

ideas and strategies that lead to this construction. In order to prove IND-STAG-CCA security for our scheme, we need to be able to simulate a decryption oracle that can handle decryption queries for all tags except for the challenge tag. We have seen in the last section that when we use the alternative decryption algorithm Dec^*, then incomplete secret keys are sufficient for this task, given that not too many components of the secret key are missing.

The tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$ will use (virtually) the same encryption and decryption algorithms as $\mathsf{PKE}_{\mathsf{McE}}$. The only difference is that we augment both algorithms by key-derivation mechanisms, that derive tag-specific keys from the full keys. More specifically, encryption takes a full public key pk and derives a tag-specific public key pk_τ, which is basically a projection of pk onto *coordinates* corresponding to τ. Similarly, decryption takes a full secret key sk and derives a tag-specific secret key sk_τ, which is again a projection of sk to coordinates specified by τ.

When we reduce the IND-STAG-CCA security of $\mathsf{TBE}_{\mathsf{McE}}$ to the IND-CPA security of $\mathsf{PKE}_{\mathsf{McE}}$, the reduction will not be in possession of the secret key corresponding to the challenge tag τ^*. We will therefore set all components of the full secret key sk that appear in sk_{τ^*} to $\perp$ (erasure). We need to ensure that no other tag τ shares too many components with the challenge tag τ^*, so that we can use Dec^* to answer decryption queries for any tag $\tau \neq \tau^*$. We will accomplish this by encoding the tags with a code of minimum distance $l+1$. This guarantees that two distinct tags τ_1 and τ_2 agree on at most $k-l-1$ locations. Consequently, every tag $\tau \neq \tau^*$ agrees with τ^* on at most $k-l-1$ locations.

Construction 5.4. *Let λ be a security parameter. Let $m, n, t = \mathsf{poly}(n)$ be such that a binary $[m, n, 2t+1]$ Goppa codes exists. Let $q = \mathsf{poly}(\lambda)$ be a prime power and $l, k, r = \mathsf{poly}(\lambda)$ be such that a q-ary linear $[k, r, l+1]$ code $\mathsf{C}_{\mathsf{tag}}$ exists. The tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}} = (\mathsf{TBE}_{\mathsf{McE}}.\mathsf{KeyGen}, \mathsf{TBE}_{\mathsf{McE}}.\mathsf{Enc}, \mathsf{TBE}_{\mathsf{McE}}.\mathsf{Dec})$ is specified as follows.*

- $\mathsf{TBE}_{\mathsf{McE}}.\mathsf{KeyGen}(1^\lambda)$*:*
  For $i = 1, \ldots, k$ and $a \in \mathbb{F}_q$
  $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow \mathsf{McE}.\mathsf{Gen}(1^\lambda)$
  $pk \leftarrow (\mathbf{A}_{i,a})_{i,a}$
  $sk \leftarrow (\mathsf{td}_{i,a})_{i,a}$
  Return (pk, sk)

- $\mathsf{TBE}_{\mathsf{McE}}.\mathsf{Enc}(pk, \tau, \mathfrak{m})$*:*
  Parse $pk = (\mathbf{A}_{i,a})_{i,a}$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_k)$
  $pk_\tau \leftarrow (\mathbf{A}_{1,\hat{\tau}_1}, \ldots, \mathbf{A}_{k,\hat{\tau}_k})$
  $\mathfrak{c} \leftarrow \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}(pk_\tau, \mathfrak{m})$
  Return $\mathfrak{c}$.

- $\mathsf{TBE}_{\mathsf{McE}}.\mathsf{Dec}(sk, \tau, \mathfrak{c})$*:*
  Parse $sk = (\mathsf{td}_{i,a})_{i,a}$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_k)$
  $sk_\tau \leftarrow (\mathsf{td}_{1,\hat{\tau}_1}, \ldots, \mathsf{td}_{k,\hat{\tau}_k})$
  $\mathfrak{m} \leftarrow \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

The plaintext space of $\mathsf{TBE}_{\mathsf{McE}}$ *is* $\mathfrak{M}_{\mathsf{McE},\lambda} = \mathbb{F}_2^{l \cdot n}$*, the ciphertext space is* $\mathfrak{C}_{\mathsf{McE},\lambda} = \mathbb{F}_2^{k \cdot m}$ *and the tag space is* $\mathfrak{T}_{\mathsf{McE},\lambda} = \mathbb{F}_q^r$.

We will first show that the scheme $\mathsf{TBE}_{\mathsf{McE}}$ fulfills the completeness requirement.

Lemma 5.3. *The tag-based encryption scheme* $\mathsf{TBE}_{\mathsf{McE}}$ *is complete.*

Proof. Since the derived public key pk_τ is a proper public key for $\mathsf{PKE}_{\mathsf{McE}}$, the completeness of $\mathsf{TBE}_{\mathsf{McE}}$ follows directly from the completeness of $\mathsf{PKE}_{\mathsf{McE}}$ (Lemma 5.1). □

5.4.1. IND-STAG-CCA2 Security

We will now establish that the tag-based encryption scheme $\mathsf{TBE}_{\mathsf{McE}}$ is IND-STAG-CCA2 secure.

Theorem 5.2. $\mathsf{TBE}_{\mathsf{McE}}$ *is IND-STAG-CCA2 secure, given that* $\mathsf{PKE}_{\mathsf{McE}}$ *is IND-CPA secure.*

Proof. Let $\mathcal{A}$ be a PPT-adversary with advantage ϵ against the IND-STAG-CCA2 security of $\mathsf{TBE}_{\mathsf{McE}}$. We will construct an adversary $\mathcal{A}'$ with advantage ϵ against the IND-CPA security of $\mathsf{PKE}_{\mathsf{McE}}$. Consider the following experiments **Game 0** and **Game 1**.

Game 0

$(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathtt{init}, 1^\lambda)$
For $i = 1, \ldots, k$ and $a \in \mathbb{F}_q$
  $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow \mathsf{McE.Gen}(1^\lambda)$
$pk \leftarrow (\mathbf{A}_{i,a})_{i,a}$
$sk \leftarrow (\mathsf{td}_{i,a})_{i,a}$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk,\tau^*,\cdot,\cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0,1\}$
$\hat{\tau}^* \leftarrow \mathsf{C_{tag}.Encode}(\tau^*)$
Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_k^*)$
$pk_{\tau^*} \leftarrow (\mathbf{A}_{1,\hat{\tau}_1^*}, \ldots, \mathbf{A}_{k,\hat{\tau}_k^*})$
$\mathfrak{c}^* \leftarrow \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Enc}(pk_{\tau^*}, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk,\tau^*,\cdot,\cdot)}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \tau, \mathfrak{c})$
  If $\tau = \tau^*$
    Return $\perp$
  Parse $sk = (\mathsf{td}_{i,a})_{i,a}$
  $\hat{\tau} \leftarrow \mathsf{C_{tag}.Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_k)$
  $sk_\tau \leftarrow (\mathsf{td}_{1,\hat{\tau}_1}, \ldots, \mathsf{td}_{k,\hat{\tau}_k})$
  $\mathfrak{m} \leftarrow \mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}(sk_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

Clearly, **Game 0** is the IND-STAG-CCA2 experiment for $\mathsf{TBE}_{\mathsf{McE}}$, for which we have substituted $\mathsf{TBE.KeyGen}$, $\mathsf{TBE.Enc}$ and $\mathsf{TBE.Dec}$ with their implementations according to $\mathsf{TBE}_{\mathsf{McE}}$.

Game 1 is identical to **Game 0**, except for the following changes. The lines "$\hat{\tau}^* \leftarrow \mathsf{C_{tag}.Encode}(\tau^*)$" and "Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_k^*)$" have been moved further up. This however does not affect the experiment, since the tag τ^* does not change during the experiment. Moreover, the public and secret key components corresponding to the encoded challenge-tag $\hat{\tau}^*$ are computed seperately from the other public and secret key components and the secret key $\tilde{sk}$ is set to $\perp$ at locations corresponding to the encoded challenge-tag $\hat{\tau}^*$. Finally, the decryption oracle $\mathcal{O}_{\mathsf{Dec}}$ uses the alternative decryption algorithm Dec^* instead of $\mathsf{PKE}_{\mathsf{McE}}.\mathsf{Dec}$.

Game 1

$(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathtt{init}, 1^\lambda)$
$\hat{\tau}^* \leftarrow \mathsf{C_{tag}}.\mathsf{Encode}(\tau^*)$
Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_k^*)$
$(pk^*, sk^*) \leftarrow \mathsf{PKE_{McE}}.\mathsf{KeyGen}(1^\lambda)$
Parse $pk^* = (\mathbf{A}_1^*, \ldots, \mathbf{A}_k^*)$
For $i = 1, \ldots, k$ and $a \in \mathbb{F}_q$
  If $a = \hat{\tau}_i^*$
    $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow (\mathbf{A}_i^*, \perp)$
  Otherwise
    $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow \mathsf{McE}.\mathsf{Gen}(1^\lambda)$
$pk \leftarrow (\mathbf{A}_{i,a})_{i,a}$
$\tilde{sk} \leftarrow (\mathsf{td}_{i,a})_{i,a}$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \cdot, \cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{c}^* \leftarrow \mathsf{PKE_{McE}}.\mathsf{Enc}(pk_{\tau^*}, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \cdot, \cdot)}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \tau, \mathfrak{c})$
  If $\tau = \tau^*$
    Return $\perp$
  Parse $sk = (\mathsf{td}_{i,a})_{i,a}$
  $\hat{\tau} \leftarrow \mathsf{C_{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_k)$
  $\tilde{sk}_\tau \leftarrow (\mathsf{td}_{1,\hat{\tau}_1}, \ldots, \mathsf{td}_{k,\hat{\tau}_k})$
  $\mathfrak{m} \leftarrow \mathsf{Dec}^*(pk, \tilde{sk}_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

We claim that from the view of $\mathcal{A}$, **Game 0** and **Game 1** are identically distributed. First observe that the public keys pk are identically distributed in both experiments. As the code $\mathsf{C_{tag}}$ has minimum distance $l+1$, each encoded tag $\hat{\tau}$ has Hamming-distance at least $l+1$ from the encoded challenge-tag $\hat{\tau}^*$. Thus, for each $\tau \neq \tau^*$ the derived secret key $\tilde{sk}_\tau$ has at most $k-l-1$ erasures. By Lemma 5.2 it holds that $\mathsf{PKE_{McE}}.\mathsf{Dec}(sk_\tau, \mathfrak{c}) = \mathsf{Dec}^*(pk, \tilde{sk}_\tau, \mathfrak{c})$, where sk_τ is a derived secret key without erasures. Thus the decryption oracle has identical behavior in both experiments and we can conclude that **Game 1** and **Game 2** are identically distributed from the view of $\mathcal{A}$ and consequently $\mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A}) = \mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A}) = \epsilon$.

We can now construct the IND-CPA adversary $\mathcal{A}'$.

$\mathcal{A}'$ simulates **Game 1** faithfully, except that the public key pk^* and the challenge-ciphertext $\mathfrak{c}^*$ are provided by the IND-CPA experiment. However, the IND-CPA experiment computes pk^* and $\mathfrak{c}^*$ exactly in the same way as **Game 1** does, so the output of the IND-CPA experiment with $\mathcal{A}'$ is identically distributed to the output of **Game 1**. Therefore it holds that $\mathsf{Adv}_{IND-CPA}(\mathcal{A}') = \mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A}) = \epsilon$ which is non-negligible. This contradicts the IND-CPA security of $\mathsf{PKE_{McE}}$, which concludes the proof. □

5.5. Instantiation of the IND-CCA2 Scheme

We will now discuss how the parameters m, n, t, k, r, l and ρ should be chosen when instantiating $\mathsf{TBE_{McE}}$. We have to determine a parameter set that meets the following constraints.

1. The scheme $\mathsf{TBE_{McE}}$ should be complete.
2. The scheme $\mathsf{TBE_{McE}}$ should have security $2^{\Omega(\lambda)}$.

Adversary $\mathcal{A}'$
$\mathcal{A}'(\texttt{find}, pk^*)$
  $(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\texttt{init}, 1^\lambda)$
  $\hat{\tau}^* \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau^*)$
  Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_k^*)$
  Parse $pk^* = (\mathbf{A}_1^*, \ldots, \mathbf{A}_k^*)$
  For $i = 1, \ldots, k$ and $a \in \mathbb{F}_q$
    If $a = \hat{\tau}_i^*$
      $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow (\mathbf{A}_i^*, \perp)$
    Otherwise
      $(\mathbf{A}_{i,a}, \mathsf{td}_{i,a}) \leftarrow \mathsf{McE}.\mathsf{Gen}(1^\lambda)$
  $pk \leftarrow (\mathbf{A}_{i,a})_{i,a}$
  $\tilde{sk} \leftarrow (\mathsf{td}_{i,a})_{i,a}$
  $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \cdot, \cdot)}(\texttt{find}, \mathsf{st}_0, pk)$
  $\mathsf{st} \leftarrow (pk, \tilde{sk}, \tau^*, \mathsf{st}_1)$
  Return $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st})$
$\mathcal{A}'(\texttt{guess}, \mathsf{st}, \mathfrak{c}^*)$
  Parse $\mathsf{st} = (pk, \tilde{sk}, \tau^*, \mathsf{st}_1)$
  $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \cdot, \cdot)}(\texttt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
  Return b'

$\mathcal{O}_{\mathsf{Dec}}(pk, \tilde{sk}, \tau^*, \tau, \mathfrak{c})$
  If $\tau = \tau^*$
    Return $\perp$
  Parse $sk = (\mathsf{td}_{i,a})_{i,a}$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_k)$
  $\tilde{sk}_\tau \leftarrow (\mathsf{td}_{1,\hat{\tau}_1}, \ldots, \mathsf{td}_{k,\hat{\tau}_k})$
  $\mathfrak{m} \leftarrow \mathsf{Dec}^*(pk, \tilde{sk}_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

3. The size of the tag space $\mathfrak{T}_{\mathsf{McE},\lambda}$ should be 2^λ.

Under these constraints, we would like to minimize the size of the keys and the ciphertext expansion. We will now derive more specific constraints. There are three codes involved: The outer Reed Solomon Code RS, the inner scrambled Goppa codes and the code $\mathsf{C}_{\mathsf{tag}}$ encoding the tags. It will be convenient to consider these codes in terms of their respective rates. Let $R_{\mathsf{McE}} = \frac{n}{m}$ be the rate of the Goppa codes, $R_{\mathsf{RS}} = \frac{l+1}{k}$ be the rate of the Reed Solomon code and $R_{\mathsf{tag}} = \frac{r}{k}$ be the rate of $\mathsf{C}_{\mathsf{tag}}$. The size of the plaintexts of $\mathsf{TBE}_{\mathsf{McE}}$ is $l \cdot n$ and the size of the ciphertexts $k \cdot m$. Thus the ciphertext expansion is

$$E_{\mathsf{McE}} = \frac{k \cdot m}{l \cdot n} = \frac{k \cdot m}{(l+1)n - n} = \frac{k \cdot m}{R_{\mathsf{RS}} R_{\mathsf{McE}} k m - R_{\mathsf{McE}} m} = \frac{1}{R_{\mathsf{RS}} R_{\mathsf{McE}} - \frac{R_{\mathsf{McE}}}{k}}.$$

For a growing k, this is essentially $\frac{1}{R_{\mathsf{RS}} R_{\mathsf{McE}}}$. The size of the public keys is

$$|pk| = q \cdot k \cdot m \cdot n = q \cdot k \cdot R_{\mathsf{McE}} m^2.$$

Finally, the size of the tag space is

$$|\mathfrak{T}_{\mathsf{McE},\lambda}| = q^r = 2^{\log(q) R_{\mathsf{tag}} k}.$$

We will now derive the precise constraints. By Lemma 5.1, $\mathsf{TBE}_{\mathsf{McE}}$ is complete whenever

$$(1 + \beta)\rho m \leq t.$$

As $t = \frac{(1-R_{\mathsf{McE}})m}{\log(m)}$ this can be equivalently written as

$$\rho \leq \frac{1 - R_{\mathsf{McE}}}{(1+\beta)\log(m)}.$$

By Theorems 5.1 and 5.2 $\mathsf{TBE}_{\mathsf{McE}}$ is IND-STAG-CCA2 secure given that the decisional McEliece problem and the LPN problem $\mathsf{DLPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$ are hard and $\mathsf{C}_{\mathsf{tag}}$ has minimum distance at least $l + 1$. Thus, want both problems to have (conjectured) hardness $2^{\Omega(\lambda)}$. The decisional McEliece problem should have hardness $2^{\Omega(\lambda)}$ if $m = \Omega(\lambda)$ and R_{McE} is a constant smaller than 1. We assume that the best attack against $\mathsf{LPN}(n, m \cdot k, \mathsf{Ber}(m \cdot k, \rho))$ is not significantly better than (modified) brute force search, which in this case has complexity $2^{O(\rho n)} = 2^{O(m/\log(m))}$. Thus, in order to get $2^{\Omega(\lambda)}$ security we need to adjust the choice of m to $m = \Omega(\lambda \cdot \log(\lambda))$. We still need to choose the $\mathsf{C}_{\mathsf{tag}}$. The minimum distance of $\mathsf{C}_{\mathsf{tag}}$ must be at least $l + 1 = R_{\mathsf{RS}}k$. We end up with the constraints

1. $m = \Omega(\lambda \log(\lambda))$
2. $\rho \leq \frac{1-R_{\mathsf{McE}}}{(1+\beta)\log(m)}$
3. $\mathsf{d}(\mathsf{C}_{\mathsf{tag}}) \geq R_{\mathsf{RS}}k$
4. $\log(q)R_{\mathsf{tag}}k = \lambda$

The choice of R_{McE} is pretty much dictated by the McEliece assumption and leaves no opportunity to make adjustments. This also determines ρ. Moreover, the choice of R_{tag} follows directly from the choice of q and R_{RS}. The remaining degree of freedom between q, R_{RS} and k is now basically a parameter trade-off. We will discuss several options.

5.5.1. Minimizing the Ciphertext Expansion

The ciphertext expansion can be minimized by maximizing R_{RS}. Thus, we want to maximize R_{RS} under the constraints $\mathsf{d}(\mathsf{C}_{\mathsf{tag}}) \geq R_{\mathsf{RS}}k$ and $\log(q)R_{\mathsf{tag}}k = \lambda$. Maximizing R_{RS} is therefore equivalent to maximizing the minimum distance of $\mathsf{C}_{\mathsf{tag}}$. As q tends towards infinity we can let $\mathsf{d}(\mathsf{C}_{\mathsf{tag}})$ tend towards 1. If we let q grow with n, i.e. if we have an asymptotically growing alphabet, then Reed Solomon codes yield the best possible trade-off between rate and minimum distance, as they meet the Singleton bound. Thus, let $q \geq k$ and set $\mathsf{C}_{\mathsf{tag}}$ to be a $[k, R_{\mathsf{tag}}k, k - R_{\mathsf{tag}}k + 1]$ Reed Solomon code. Thus we have the constraint

$$1 - R_{\mathsf{tag}} \geq R_{\mathsf{RS}}$$

on R_{tag}, which is fulfilled by setting

$$R_{\mathsf{tag}} = 1 - R_{\mathsf{RS}}.$$

With this choice of R_{tag}, our remaining constraint is

$$\log(q)\,(1 - R_{\mathsf{RS}})\,k = \log(q)R_{\mathsf{tag}}k = \lambda$$

and we can thus set

$$R_{\mathsf{RS}} = 1 - \frac{\lambda}{k\log(q)} = 1 - o(1).$$

Thus, the ciphertext expansion $\frac{1}{R_{\mathsf{McE}}R_{\mathsf{RS}}}$ asymptotically tends to $\frac{1}{R_{\mathsf{RS}}}$ for $k = \Omega(\lambda)$ and $q = k$. The size of the keys is $k^2 R_{\mathsf{McE}} m^2$, which is by a factor k^2 larger than for the basic McEliece scheme.

5.5.2. Minimizing the size of the public key

The size of the public key can be minimized by minimizing $|pk| = q \cdot kR_{\mathsf{McE}}m^2$. Thus, we want to minimize $q \cdot k$ under the constraints $\mathsf{d}(\mathsf{C}_{\mathsf{tag}}) \geq R_{\mathsf{RS}}k$ and

$$\log(q)R_{\mathsf{tag}}k = \lambda.$$

Clearly, we can minimize $|pk|$ by minimizing q, therefore choosing $q = 2$. By the Gilbert Varshamov bound (Theorem 2.4), we can find a binary code $\mathsf{C}_{\mathsf{tag}}$ with relative minimum distance at least R_{RS} and rate at least $1 - H(R_{\mathsf{RS}})$. Thus, we have the remaining constraint

$$\log(q)(1 - H_q(R_{\mathsf{RS}})) \cdot k = \lambda.$$

which yields

$$k = \frac{\lambda}{1 - H(R_{\mathsf{RS}})}.$$

Thus, k can be minimized by choosing R_{RS} close to 0. Choosing R_{RS} as a small constant leads to a key growth of $\approx 2\lambda$ and a rather large, but constant factor ciphertext expansion of $\frac{1}{R_{\mathsf{McE}}R_{\mathsf{RS}}}$. The ciphertext expansion and the key expansion can be balanced by choosing R_{RS} such that

$$1 - H(R_{\mathsf{RS}}) = R_{\mathsf{RS}}$$

which is the case for $R_{\mathsf{RS}} \approx 0.77$. This leads to a ciphertext expansion of $\approx 1.3/R_{\mathsf{McE}}$ and a key growth of $\approx 2.6 \cdot \lambda$, which seems like a good trade-off.

5.5.3. IND-CCA2 scheme via CHK

We can now apply the CHK transformation (Theorem 4.1) and obtain the following Theorem to conclude this Chapter.

Theorem 5.3. *Let λ be a security parameter and $m = O(\lambda^2)$. There exists an IND-CCA2 secure public key encryption scheme $\mathsf{PKE}_{\mathsf{McE},CCA2}$ based on the hardness of the decisional McEliece assumption and the LPN problem $\mathsf{LPN}(\lambda, m, \mathsf{Ber}(m, O(\frac{1}{\log(\lambda)})))$. The scheme has a constant factor ciphertext expansion, plaintexts of size $\Theta(\lambda^2)$, and key sizes of $\Theta(\lambda^3)$.*

6. IND-CCA2 Secure Public Key Encryption from the LPN Assumption

> So seltsam es auch klingen mag, die Stärke der Mathematik beruht auf dem Vermeiden jeder unnötigen Annahme und auf ihrer großartigen Einsparung an Denkarbeit.
>
> Ernst Mach

6.1. Introduction

In Chapter 5 we have seen a construction of an IND-CCA2 secure public key encryption scheme based on the hardness of the decisional McEliece assumption and the LPN problem. A favorable feature of this scheme is that all its operations can be implemented using simple modulo 2 arithmetic. However, we have also seen that certain instantiations of the McEliece assumption give rise to algebraic attacks. This may lead to concerns regarding the hardness of the classical McEliece problem. For the LPN problem in contrast, no such structural attacks are known. Moreover, the LPN problem is arguably of a more combinatorial nature than the rather *algebraic* McEliece problem.

In this Chapter we will provide the construction of a standard model IND-CCA2 secure public key cryptosystem which is solely based on an LPN assumption. A preliminary version of this construction is due to Döttling, Müller-Quade and Nascimento [DMQN12]. We will need a low noise version of LPN such as used by Alekhnovich [Ale03]. More specifically, the hardness of our scheme will be based on the LPN problem $\mathsf{LPN}(n, m, \mathsf{Ber}(m, O(m^{-\frac{1}{2}})))$ for some $m = O(n)$.

As mentioned in Section 4.2, standard model IND-CCA2 security has been achieved for virtually all standard number-theoretic assumptions. Efforts to construct IND-CCA2 secure cryptosystems based on the hardness of lattice and coding assumptions succeeded rather recently. In the LWE realm, this was first achieved

by Peikert and Waters [PW08]. Improved constructions were provided by Peikert [Pei09] and Micciancio and Peikert [MP12]. The construction we will provide in this section was mostly inspired by McEliece scheme provided in Chapter 5, however, it also bears certain resemblances with the scheme of Micciancio and Peikert [MP12].

In an asymptotical sense, the scheme we construct in this Chapter is optimal regarding ciphertext expansion, key sizes and runtimes of the involved algorithms.

6.1.1. Outline

We will start with a rough outline of a simplified scheme that encrypts single bits and has a substantial decryption-error. On a technical level, this first building block resembles the schemes of Regev [Reg05] and the Dual-Regev Scheme of Gentry et al. [GPV08] (which both live in the LWE realm) and uses a slightly simpler trapdoor than [DMQN12], as proposed by Damgård and Park [DP12]. Public keys for this scheme are pairs $(\mathbf{a}, \mathbf{A})$, where $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m \times n}$ is chosen uniformly at random and $\mathbf{a} = \mathbf{A}^T\mathbf{w}$, where $\mathbf{w} \leftarrow_{\$} \mathsf{Ber}(m, \rho)$ is chosen by a Bernoulli distribution $\mathsf{Ber}(m, \rho)$. The secret key of this simplified scheme is $\mathbf{w}$. To encrypt a message $\mathfrak{m} \in \mathbb{F}_2$, sample $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$, $\mathbf{e} \leftarrow_{\$} \mathsf{Ber}(m, \rho)$, $e \leftarrow_{\$} \mathsf{Ber}(\rho)$ and set $\mathfrak{c} \leftarrow (\langle \mathbf{a}, \mathbf{s} \rangle + e + \mathfrak{m}, \mathbf{A}\mathbf{s} + \mathbf{e})$ and output $\mathfrak{c}$. To decrypt a ciphertext $\mathfrak{c} = (\mathfrak{c}_1, \mathfrak{c}_2) \in \mathbb{F}_2 \times \mathbb{F}_2^m$, compute $y = \mathfrak{c}_1 - \langle \mathbf{w}, \mathfrak{c}_2 \rangle$ and output y. The output y is a *noisy* version of the plaintext $\mathfrak{m}$, since it holds that

$$\begin{aligned} y &= \mathfrak{c}_1 - \langle \mathbf{w}, \mathfrak{c}_2 \rangle \\ &= \mathfrak{m} + e + \langle \mathbf{a}, \mathbf{s} \rangle - \langle \mathbf{w}, \mathbf{A}\mathbf{s} + \mathbf{e} \rangle \\ &= \mathfrak{m} + e + \langle \mathbf{A}^T\mathbf{w}, \mathbf{s} \rangle - \langle \mathbf{w}, \mathbf{A}\mathbf{s} \rangle - \langle \mathbf{w}, \mathbf{e} \rangle \\ &= \mathfrak{m} + e + \langle \mathbf{w}, \mathbf{A}\mathbf{s} \rangle - \langle \mathbf{w}, \mathbf{A}\mathbf{s} \rangle - \langle \mathbf{w}, \mathbf{e} \rangle \\ &= \mathfrak{m} + e - \langle \mathbf{w}, \mathbf{e} \rangle. \end{aligned}$$

As both $\mathbf{w}$ and $\mathbf{e}$ are chosen from $\mathsf{Ber}(m, \rho)$, $\langle \mathbf{w}, \mathbf{e} \rangle$ is approximately distributed by $\mathsf{Ber}(\rho^2 m)$. Thus, once ρ is sufficiently small (on the order of $O(m^{-\frac{1}{2}})$), the decryption error is low enough that it can be handled. Security of this simplified scheme follows by the decisional LPN assumption. First notice that the vector $\mathbf{a}$ of the public key is indistinguishable from uniformly random by the dual decisional LPN assumption. Next, by the decisional LPN assumption $(\langle \mathbf{a}, \mathbf{s} \rangle + e, \mathbf{A}\mathbf{s} + \mathbf{e})$ is indistinguishable from uniformly random and IND-CPA security follows.

Considering this scheme, the process of encrypting a message $\mathfrak{m}$ and decrypting the ciphertext $\mathfrak{c}$ acts like a noisy channel on the message $\mathfrak{m}$. The standard way of dealing with this noise is encoding longer messages $\mathfrak{m}$ using an error correcting code. Thus, assume that $\mathfrak{m} \in \mathbb{F}_2^n$ and that C_1 is a linear $[m, n]$ error correcting code. If $\mathbf{x} \leftarrow \mathsf{C}_1.\mathsf{Encode}(\mathfrak{m})$ is the encoding of $\mathfrak{m}$, we can just use the simplified scheme to transmit all the bits x_i of $\mathbf{x}$ and then use an efficient decoder for C_1 to remove the decryption error after decryption. However, there is room for optimization. The basic observation is that we can use the same vector $\mathbf{s}$ for all the encryptions of the bits of $\mathbf{x}$, i.e. we can batch the encryption of the x_i. For the modified scheme, the secret key is a matrix $\mathbf{W} \in \mathbb{F}_2^{m \times m}$ chosen according to $\mathsf{Ber}(m \times m, \rho)$, and the public key is $(\mathbf{A}_1, \mathbf{A}_3)$[1] where $\mathbf{A}_3 \leftarrow_{\$} \mathbb{F}_2^{m \times n}$ is chosen uniformly at random and

[1] It will become clear in a moment why we call the second matrix $\mathbf{A}_3$ instead of $\mathbf{A}_2$

$\mathbf{A}_1 = \mathbf{W}^T\mathbf{A}_3$. Messages $\mathfrak{m}$ are encrypted to ciphertexts $\mathfrak{c} = (\mathfrak{c}_1, \mathfrak{c}_3)$[2] by computing

$$\begin{aligned}\mathfrak{c}_1 &\leftarrow \mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathfrak{m})\\ \mathfrak{c}_3 &\leftarrow \mathbf{A}_3\mathbf{s} + \mathbf{e}_3,\end{aligned}$$

where as above $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ is chosen uniformly at random and $\mathbf{e}_1, \mathbf{e}_3 \leftarrow_\$ \mathsf{Ber}(m, \rho)$. There is a slightly better choice for the distribution of $\mathbf{e}_1$ and $\mathbf{e}_3$. By choosing $\mathbf{e}_1$ and $\mathbf{e}_3$ uniformly from the Hamming sphere $\mathsf{S}_{\lfloor \rho m \rfloor}$, we will be able to avoid decryption errors. Moreover, by Lemma 3.6 the corresponding LPN problem is at least as hard as LPN with Bernoulli errors.

Similar to the simplified scheme above, ciphertexts $(\mathfrak{c}_1, \mathfrak{c}_3)$ are decrypted by computing

$$\begin{aligned}\mathbf{x} &= \mathfrak{c}_1 - \mathbf{W}^T\mathfrak{c}_3\\ &= \mathsf{C}_1.\mathsf{Encode}(\mathfrak{m}) + \mathbf{A}_1\mathbf{s} + \mathbf{e}_1 - \mathbf{W}^T(\mathbf{A}_3\mathbf{s} + \mathbf{e}_3)\\ &= \mathsf{C}_1.\mathsf{Encode}(\mathfrak{m}) + \mathbf{W}^T\mathbf{A}_3\mathbf{s} + \mathbf{e}_1 - \mathbf{W}^T\mathbf{A}_3\mathbf{s} - \mathbf{W}^T\mathbf{e}_3\\ &= \mathsf{C}_1.\mathsf{Encode}(\mathfrak{m}) + \mathbf{e}_1 - \mathbf{W}^T\mathbf{e}_3.\end{aligned}$$

Since $\mathbf{W}$, $\mathbf{e}_1$ and $\mathbf{e}_3$ are chosen from low weight distributions, the error term $\mathbf{W}^T\mathbf{e}_3$ has also low weight and we can recover $\mathfrak{m}$ from $\mathbf{x}$ by using the efficient decoder of C_1, i.e. $\mathfrak{m} \leftarrow \mathsf{C}_1.\mathsf{Decode}(\mathbf{x})$. IND-CPA security follows as above.

Recall that for our McEliece based construction in Chapter 5 we required a witness recovering encryption scheme, i.e. decryption must be able to recover the noise terms $\mathbf{e}$ and test whether they are *valid*. We did this to facilitate the construction of an alternative decryption algorithm that accomplishes decryption using incomplete public keys. In this Chapter, we follow the same strategy. However with the above scheme, this is not yet possible, i.e. there is no obvious way to recover $\mathbf{e}_1$ and $\mathbf{e}_2$ given the secret key $\mathbf{W}$ and the ciphertext $\mathfrak{c}$. We circumvent this problem by encrypting the witness $\mathbf{s}$ instead of the message $\mathfrak{m}$. Thus, we will add an extra component $\mathfrak{c}_2$ to the ciphertext that encrypts the actual message $\mathfrak{m}$ encoded using a code C_2. We could actually encode $\mathfrak{m}$ together with $\mathbf{s}$. However, this would have a negative impact on the ciphertext expansion of the scheme, as the code C_1 needs to deal with *high noise* during decryption, whereas as separate code C_2 will only need to deal with a low amount of noise. Let therefore $\mathbf{C}_2$ be another $[m, n]$ error correcting code. The public key is extended by a randomly chosen matrix $\mathbf{A}_2^{m\times n}$, i.e. the public key is $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ where $\mathbf{A}_1 = \mathbf{W}^T\mathbf{A}_3$. Since we want to recover the error terms, the secret key now actually contains the matrices of the public key, i.e. the secret key is $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W})$. Messages $\mathfrak{m}$ are encrypted to ciphertexts $\mathfrak{c} = (\mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c}_3)$ by computing

$$\begin{aligned}\mathfrak{c}_1 &\leftarrow \mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})\\ \mathfrak{c}_1 &\leftarrow \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})\\ \mathfrak{c}_3 &\leftarrow \mathbf{A}_3\mathbf{s} + \mathbf{e}_3,\end{aligned}$$

where again $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \leftarrow_\$ \mathsf{S}_m(\lfloor \rho m \rfloor)$. To decrypt, we proceed as above and compute

$$\mathbf{x} = \mathfrak{c}_1 - \mathbf{W}^T\mathfrak{c}_3 = \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) + \mathbf{e}_1 - \mathbf{W}^T\mathbf{e}_3$$

[2] Same as for the matrix $\mathbf{A}_3$

to recover $\mathbf{s}$ by $\mathbf{s} \leftarrow \mathsf{C}_1.\mathsf{Decode}(\mathbf{x})$. We now recover $\mathfrak{m}$ by computing

$$\mathfrak{m} \leftarrow \mathsf{C}_2.\mathsf{Decode}(\mathfrak{c}_2 - \mathbf{A}_2 \cdot \mathbf{s}) = \mathsf{C}_2.\mathsf{Decode}(\mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}) + \mathbf{e}_2).$$

Once we have obtained $\mathbf{s}$ and $\mathfrak{m}$, we can recover $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ by

$$\begin{aligned}\mathbf{e}_1 &\leftarrow \mathfrak{c}_1 - \mathbf{A}_1\mathbf{s} - \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})\\ \mathbf{e}_2 &\leftarrow \mathfrak{c}_2 - \mathbf{A}_2\mathbf{s} - \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})\\ \mathbf{e}_3 &\leftarrow \mathfrak{c}_3 - \mathbf{A}_2\mathbf{s}.\end{aligned}$$

The decryption algorithm can now check whether $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ have appropriate Hamming weight (i.e. exactly $\lfloor \rho m \rfloor$), and if not reject decryption.

Basically, we have now constructed an IND-CPA secure encryption scheme that could take the role of the basic McEliece scheme in the constructions of Chapter 5. However, recall that the McEliece based scheme in Section 5.3 needed to use $\Omega(\lambda)$ McEliece trapdoors, which leads to rather large keys and ciphertexts. This blowup occurred because we don't know of any (sound) way of using the McEliece trapdoor in a non-monolithic way. Put differently, we basically used the McEliece algorithms $\mathsf{McE.Gen}$ and $\mathsf{McE.Decode}$ in an axiomatic way, without taking any advantage of their *internals*.

The situation is different now. Due to the very simple structure of the LPN problem and the trapdoors we have constructed, we will be able take advantage of their internal structure. Specifically, we do not need to use the above trapdoor monolithically. Recall that in order to construct a tag-based encryption scheme we want to be able to derive tag-specific keys from a master key. In the McEliece case we achieved this by using $O(\lambda)$ McEliece keys in the master key. Each of the McEliece secret keys is a full-fledged trapdoor that can be used to recover an entire block. Here, we take a different approach, instead of chopping up the messages in blocks and encrypting the blocks with monolithic keys we will chop up the keys and combine them in a tag-specific way. Specifically, we can split the trapdoor $\mathbf{W} \in \mathbb{F}_2^{m \times m}$ in k_1 *partial trapdoors* of block size k_2, i.e. if $m = k_1 \cdot k_2$ we split $\mathbf{W}$ into

$$\mathbf{W} = (\mathbf{W}_1 \| \ldots \| \mathbf{W}_{k_1}),$$

where each $\mathbf{W}_i$ is a matrix in $\mathbb{F}_2^{m \times k_1}$. This means in particular that we can combine a $\mathbf{W}$ by combining different choices of the $\mathbf{W}_i$. This decomposition of $\mathbf{W}$ into blocks also corresponds to a decomposition of the public keys and the ciphertexts into blocks. Specifically, we get

$$\mathbf{A}_1 = \mathbf{W}^T \cdot \mathbf{A}_3 = (\mathbf{W}_1 \| \ldots \| \mathbf{W}_{k_1})^T \cdot \mathbf{A}_3 = (\mathbf{A}_3^T\mathbf{W}_1 \| \ldots \| \mathbf{A}_3^T\mathbf{W}_{k_1})^T.$$

If $\mathfrak{c} = (\mathfrak{c}_1, \mathfrak{c}_2, \mathfrak{c}_3)$ is a ciphertext, we can decompose $\mathfrak{c}_1$ into

$$\mathfrak{c}_1 = (\mathbf{y}_1, \ldots, \mathbf{y}_{k_1})$$

and decrypt $\mathbf{y}_i$ by

$$\tilde{\mathbf{x}}_i = \mathbf{y}_i - \mathbf{W}_i^T\mathfrak{c}_3.$$

The block $\tilde{\mathbf{x}}_i$ is now a noisy block of the codeword $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_{k_1})$. If we have all the blocks $\mathbf{x}_1, \ldots, \mathbf{x}_k$ we can use the decoder $\mathsf{C}_1.\mathsf{Decode}$ to decode the bit-errors on $\tilde{\mathbf{x}}$. However, if several blocks are missing, we will need a decoder that jointly

decodes errors and erasures. We may even choose the blocks of length $k_2 = 1$ (as in [DMQN12]), but this leads to sub-optimal concrete performance.

We will again be faced with the task of simulating a decryption oracle using incomplete keys. We will therefore construct the our IND-CPA scheme in a way that allows decryption using incomplete secret keys. The alternative decryption algorithm Dec^* we construct for this task will have to handle bit errors and block erasures simultaneously. As the blocks for which Dec^* lacks the keys are not really erased, but rather inaccessible for Dec^*, we will be able to make use of a list-decoding advantage for Dec^*. We can design Dec^* to operate in noise regimes that do not allow unique decoding. However, given a short list of candidate secrets $\mathbf{s}$ will be sufficient, as we can check for each $\mathbf{s}$ if it is a valid solution by recovering the corresponding noise terms $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ and testing whether they are valid.

6.2. The Building Block IND-CPA Scheme

In this Section we will construct the underlying IND-CPA scheme for our IND-CCA scheme from LPN. The basic idea for this scheme follows the blueprint of the McEliece cryptosystem provided in Chapter 5 and Regev's LWE-based IND-CPA secure public key encryption scheme [Reg05].

Construction 6.1. *Let λ be a security parameter. Let $m_1, m_2, m_3, n_1, n_2, t_1, t_2 = \mathsf{poly}(\lambda)$ be positive integers. Let $\rho = \rho(\lambda) \in (0, 1)$. Let C_1 be a linear $[m_1, n_1]$ code with efficient encoding and decoding of up to t_1 errors. Let C_2 be an $[m_2, n_2]$ code with efficient encoding and decoding of up to t_2 errors. The encryption scheme $\mathsf{PKE}_{\mathsf{LPN}} = (\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{KeyGen}, \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}, \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec})$ is specified as follows.*

- $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{KeyGen}(1^\lambda)$*:*
  $\mathbf{A}_2 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
  $\mathbf{A}_3 \leftarrow_\$ \mathbb{F}_2^{m_3 \times n_1}$
  $\mathbf{W} \leftarrow_\$ \mathsf{Ber}(m_3 \times m_1, \rho)$
  $\mathbf{A}_1 \leftarrow \mathbf{W}^T \cdot \mathbf{A}_3$
  $pk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$
  $sk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W})$
  Return (pk, sk)

- $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}(pk, \mathfrak{m})$*:*
  Parse $pk = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$
  $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
  $\mathbf{e}_1 \leftarrow_\$ \mathsf{S}_{m_1}(\lfloor \rho \cdot m_1 \rfloor)$
  $\mathbf{e}_2 \leftarrow_\$ \mathsf{S}_{m_2}(\lfloor \rho \cdot m_2 \rfloor)$
  $\mathbf{e}_3 \leftarrow_\$ \mathsf{S}_{m_3}(\lfloor \rho \cdot m_3 \rfloor)$
  $\mathbf{y}_1 \leftarrow \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
  $\mathbf{y}_2 \leftarrow \mathbf{A}_2 \cdot \mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})$
  $\mathbf{y}_3 \leftarrow \mathbf{A}_3 \cdot \mathbf{s} + \mathbf{e}_3$
  $\mathfrak{c} \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
  Return $\mathfrak{c}$

- $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}(sk, \mathfrak{c})$*:*
  Parse $sk = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W})$ *and* $\mathfrak{c} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
  $\mathbf{z} \leftarrow \mathbf{y}_1 - \mathbf{W}^T \cdot \mathbf{y}_3$
  $\mathbf{s} \leftarrow \mathsf{C}.\mathsf{Decode}(\mathbf{z})$

$\mathbf{x} \leftarrow \mathbf{y}_2 - \mathbf{A}_2\mathbf{s}$
$\mathfrak{m} \leftarrow \mathsf{C}_2.\mathsf{Decode}(\mathbf{x})$
$\mathbf{e}_1 \leftarrow \mathbf{y}_1 - \mathbf{A}_1\mathbf{s} - \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
$\mathbf{e}_2 \leftarrow \mathbf{y}_1 - \mathbf{A}_2\mathbf{s} - \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})$
$\mathbf{e}_3 \leftarrow \mathbf{y}_3 - \mathbf{A}_3\mathbf{s}$
If $\mathsf{wgt}(\mathbf{e}_1) = \lfloor \rho m_1 \rfloor$ *and* $\mathsf{wgt}(\mathbf{e}_2) = \lfloor \rho m_2 \rfloor$ *and* $\mathsf{wgt}(\mathbf{e}_3) = \lfloor \rho m_3 \rfloor$
  Return $\mathfrak{m}$
Otherwise
  Return $\perp$

The plaintext space of $\mathsf{PKE}_{\mathsf{LPN}}$ *is* $\mathfrak{M}_{\mathsf{LPN},\lambda} = \mathbb{F}_2^{n_2}$ *and the ciphertext space is* $\mathfrak{C}_{\mathsf{LPN},\lambda} = \mathbb{F}_2^{m_1+m_2+m_3}$.

6.2.1. Completeness

We will first show that the scheme $\mathsf{PKE}_{\mathsf{LPN}}$ is complete.

Lemma 6.1. *Let $\beta > 0$ be a constant. Assume that the decoder $\mathsf{C}_1.\mathsf{Decode}$ can handle up to $t_1 \geq (1+\beta)\rho^2 m_1 m_3$ bit errors and $\mathsf{C}_2.\mathsf{Decode}$ can handle up to $t_2 \geq \rho m_2$ bit errors. Then the public key encryption scheme $\mathsf{PKE}_{\mathsf{LPN}}$ is complete.*

Proof. Let $pk = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ and $sk = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W})$ be a pair of public and secret keys generated by $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{KeyGen}(1^\lambda)$. Let $\mathfrak{c} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ be a ciphertext generated by $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}$. Since $\mathbf{W}$ is chosen from $\mathsf{Ber}(m_3 \times m_1, \rho)$, it holds by Lemma 2.11 that $\|\mathbf{W}^T\|_{\mathsf{wgt}} \leq (1+\beta-\frac{1}{\rho m_3})\rho m_1$, except with probability $m_3 \cdot e^{-\frac{1}{3}(\beta-\frac{1}{\rho m_3})^2 \rho m_1}$. Assume thus that $\|\mathbf{W}^T\|_{\mathsf{wgt}} \leq (1+\beta-\frac{1}{\rho m_3})\rho m_1$.

As $\mathfrak{c}$ was generated by $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}$, it holds that

$$\begin{aligned}
\mathbf{y}_1 &= \mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) = \mathbf{W}^T\mathbf{A}_3\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) \\
\mathbf{y}_2 &= \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}) \\
\mathbf{y}_3 &= \mathbf{A}_3\mathbf{s} + \mathbf{e}_3
\end{aligned}$$

Thus we get

$$\begin{aligned}
\mathbf{z} &= \mathbf{y}_1 - \mathbf{W}^T\mathbf{y}_3 \\
&= \mathbf{W}^T\mathbf{A}_3\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) - \mathbf{W}^T \cdot (\mathbf{A}_3\mathbf{s} + \mathbf{e}_3) \\
&= \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) + \mathbf{e}_1 - \mathbf{W}^T \cdot \mathbf{e}_3.
\end{aligned}$$

Since $\mathbf{e}_1 \in \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$ we have $\mathsf{wgt}(\mathbf{e}_1) = \lfloor \rho m_1 \rfloor$. Likewise we have $\mathsf{wgt}(\mathbf{e}_3) = \lfloor \rho m_3 \rfloor$ since $\mathbf{e}_3 \in \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$. As $\|\mathbf{W}^T\|_{\mathsf{wgt}} \leq (1+\beta-\frac{1}{\rho m_3})\rho m_1$ we get

$$\mathsf{wgt}(\mathbf{W}^T \cdot \mathbf{e}_3) \leq \|\mathbf{W}^T\|_{\mathsf{wgt}} \cdot \mathsf{wgt}(\mathbf{e}_3) \leq (1+\beta)\rho^2 m_1 m_3 - \rho m_1$$

and thus we can bound

$$\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}^T \cdot \mathbf{e}_3) \leq \rho m_1 + (1+\beta)\rho^2 m_1 m_3 - \rho m_1 = (1+\gamma)\rho^2 m_1 m_3.$$

The decoder $\mathsf{C}_1.\mathsf{Decode}$ will thus output $\mathbf{s}$, as it can handle up to $t_1 \geq (1+\beta)\rho^2 m_1 m_3$ errors. The decoder $\mathsf{C}_2.\mathsf{Decode}$ will then output $\mathfrak{m}$, as it can handle up to $t_2 \geq \rho m_2$ errors. Finally, the error terms $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ will be recovered correctly and the check will be passed. Thus $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$ will correctly output $\mathfrak{m}$, which concludes the proof. □

6.2.2. IND-CPA security

We will now show that the scheme $\mathsf{PKE}_{\mathsf{LPN}}$ is IND-CPA secure.

Theorem 6.1. *The public key encryption scheme* $\mathsf{PKE}_{\mathsf{LPN}}$ *is IND-CPA secure given that the* $\mathsf{DLPN}(n_1, m_1+m_2+m_3, M)$ *and* $\mathsf{DDMLPN}(n_1, m_1, m_2, \mathsf{Ber}(m_1 \times m_2, \rho))$ *are hard, where*

$$M = \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor) \times \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor) \times \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$$

Proof. Let $\mathcal{A}$ be an IND-CPA adversary against the IND-CPA security of $\mathsf{PKE}_{\mathsf{LPN}}$. Consider the experiments **Game 0**, **Game 1** and **Game 2**.

Game 0

$\mathbf{A}_2 \leftarrow_{\$} \mathbb{F}_2^{m_2 \times n_1}$
$\mathbf{A}_3 \leftarrow_{\$} \mathbb{F}_2^{m_3 \times n_1}$, $\mathbf{W} \leftarrow_{\$} \mathsf{Ber}(m_3 \times m_1, \rho)$
$\mathbf{A}_1 \leftarrow \mathbf{W}^T \cdot \mathbf{A}_3$
$pk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$
$sk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W})$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_{\$} \{0,1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
$\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$
$\mathbf{e}_1 \leftarrow_{\$} \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \leftarrow_{\$} \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$, $\mathbf{e}_3 \leftarrow_{\$} \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$
$\mathbf{y}_1 \leftarrow \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
$\mathbf{y}_2 \leftarrow \mathbf{A}_2 \cdot \mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*)$
$\mathbf{y}_3 \leftarrow \mathbf{A}_3 \cdot \mathbf{s} + \mathbf{e}_3$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Clearly, **Game 0** is the IND-CPA experiment for $\mathsf{PKE}_{\mathsf{LPN}}$, for which we have subsituted $\mathsf{PKE.KeyGen}$ and $\mathsf{PKE.Enc}$ with their implementations according to $\mathsf{PKE}_{\mathsf{LPN}}$.

Game 1

$\mathbf{A}_1 \leftarrow_{\$} \mathbb{F}_2^{m_1 \times n_1}$
$\mathbf{A}_2 \leftarrow_{\$} \mathbb{F}_2^{m_2 \times n_1}$
$\mathbf{A}_3 \leftarrow_{\$} \mathbb{F}_2^{m_3 \times n_1}$
$pk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_{\$} \{0,1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
$\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$
$\mathbf{e}_1 \leftarrow_{\$} \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \leftarrow_{\$} \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$, $\mathbf{e}_3 \leftarrow_{\$} \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$
$\mathbf{y}_1 \leftarrow \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
$\mathbf{y}_2 \leftarrow \mathbf{A}_2 \cdot \mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*)$
$\mathbf{y}_3 \leftarrow \mathbf{A}_3 \cdot \mathbf{s} + \mathbf{e}_3$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Game 1 is identical to **Game 0**, except that the matrix $\mathbf{A}_1$ is now chosen uniformly at random.

Game 2

$\mathbf{A}_1 \leftarrow_\$ \mathbb{F}_2^{m_1 \times n_1}$
$\mathbf{A}_2 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
$\mathbf{A}_3 \leftarrow_\$ \mathbb{F}_2^{m_3 \times n_1}$
$pk \leftarrow (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0,1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{e}_1 \leftarrow_\$ \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \leftarrow_\$ \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$, $\mathbf{e}_3 \leftarrow_\$ \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$
$\mathbf{y}_1 \leftarrow_\$ \mathbb{F}_2^{m_1}$
$\mathbf{y}_2 \leftarrow_\$ \mathbb{F}_2^{m_2}$
$\mathbf{y}_3 \leftarrow_\$ \mathbb{F}_2^{m_3}$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

Game 2 is identical to **Game 1**, except that the vectors $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$ are chosen uniformly at random. Clearly, $\mathcal{A}$'s advantage in **Game 2** is 0, as the challenge ciphertext $\mathfrak{c}^*$ is independent of the challenge message $\mathfrak{m}_b$. It remains to show that from the view of $\mathcal{A}$, **Game 0** and **Game 1** as well as **Game 1** and **Game 2** are indistinguishable.

Claim: It holds that $|\Pr[\mathbf{Game1}(\mathcal{A}) = 1] - \Pr[\mathbf{Game0}(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda)$, given that $\mathsf{DDMLPN}(n_1, m_3, m_1, \mathsf{Ber}(m_3 \times m_1, \rho))$ is hard.

Assume towards contradiction that $|\Pr[\mathbf{Game1}(\mathcal{A}) = 1] - \Pr[\mathbf{Game0}(\mathcal{A}) = 1]| > \epsilon$ for a non-negligible ϵ. We will construct a distinguisher $\mathcal{D}_1$ that distinguishes the distributions $(\mathbf{H}, \mathbf{H} \cdot \mathbf{E})$ and $(\mathbf{H}, \mathbf{U})$. The distinguisher $\mathcal{D}_1$ is given as follows.

First assume that $\mathcal{D}_1$'s input is of the form $(\mathbf{H}, \mathbf{H} \cdot \mathbf{E})$. Then the public key pk in $\mathcal{D}_1$'s simulation has the distribution $(\mathbf{E}^T\mathbf{H}^T, \mathbf{A}_2, \mathbf{H}^T)$, with a uniformly random $\mathbf{H}^T \in \mathbb{F}_2^{m_3 \times n_1}$ and an $\mathbf{E} \in \mathbb{F}_2^{m_3 \times m_1}$ that is distributed according to $\mathsf{Ber}(m_3 \times m_1, \rho)$. Therefore, the distribution of the public key pk is identical to the distribution of the public key pk in **Game 0**. Thus, from the view of $\mathcal{A}$, $\mathcal{D}_1$'s simulation and **Game 0** are identically distributed and we have

$$\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{H} \cdot \mathbf{E}) = 1] = \Pr[\mathbf{Game0}(\mathcal{A}) = 1].$$

If, on the other hand, $\mathcal{D}_1$'s input is of the form $(\mathbf{H}, \mathbf{U})$, then $\mathcal{A}$'s view in $\mathcal{D}_1$'s simulation is identical to $\mathcal{A}$'s view in **Game 1**. Thus we have

$$\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{U}) = 1] = \Pr[\mathbf{Game1}(\mathcal{A}) = 1].$$

This yields

$$\begin{aligned} \mathsf{Adv}_{\mathsf{DDMLPN}}(\mathcal{D}_1) &= |\Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{H} \cdot \mathbf{E}) = 1] - \Pr[\mathcal{D}_1(\mathbf{H}, \mathbf{U}) = 1] \\ &= |\Pr[\mathbf{Game0}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1]| \\ &> \epsilon, \end{aligned}$$

Distinguisher $\mathcal{D}_1$

Input $(\mathbf{H}, \mathbf{Y}) \in \mathbb{F}_2^{n_1 \times m_1} \times \mathbb{F}_2^{n_1 \times m_2}$
$\mathbf{A}_2 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
$\mathbf{A}_3 \leftarrow \mathbf{H}^T$
$\mathbf{A}_1 \leftarrow \mathbf{Y}^T$
$pk \leftarrow (\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_3)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
$\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
$\mathbf{e}_1 \leftarrow_\$ \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \leftarrow_\$ \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$, $\mathbf{e}_3 \leftarrow_\$ \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$
$\mathbf{y}_1 \leftarrow \mathbf{A}_1 \cdot \mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
$\mathbf{y}_2 \leftarrow \mathbf{A}_2 \cdot \mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*)$
$\mathbf{y}_3 \leftarrow \mathbf{A}_3 \cdot \mathbf{s} + \mathbf{e}_3$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

which contradicts the hardness of $\mathsf{DDMLPN}(n_1, m_3, m_1, \mathsf{Ber}(m_3 \times m_1, \rho))$, as ϵ is non-negligible.

Claim: It holds that $|\Pr[\mathbf{Game2}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda)$, given that $\mathsf{DLPN}(n_1, m_1 + m_2 + m_3, M)$ is hard.

Assume towards contradiction that $|\Pr[\mathbf{Game2}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1] > \epsilon$ for a non-negligible ϵ. We will construct a distinguisher $\mathcal{D}_2$ that distinguishes $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ from $(\mathbf{A}, \mathbf{u})$, where

$$\mathbf{e} \leftarrow_\$ M = \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor) \times \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor) \times \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$$

and $\mathbf{u} \in \mathbb{F}_2^{m_1+m_2+m_3}$ is chosen uniformly at random. Let $\mathbf{G} \in \mathbb{F}_2^{m_1 \times n_1}$ be the generator-matrix of C_1 used by $\mathsf{C}_1.\mathsf{Encode}$. The distinguisher $\mathcal{D}_2$ is given as follows.

Distinguisher $\mathcal{D}_2$

Input $(\mathbf{A}, \mathbf{y}) \in \mathbb{F}_2^{m_1+m_2+m_3 \times n_1} \times \mathbb{F}_2^{m_1+m_2+m_3}$
Parse $\mathbf{A}^T = (\mathbf{B}^T \| \mathbf{A}_2^T \| \mathbf{A}_3^T)$
Parse $\mathbf{y}^T = (\mathbf{y}_1^T \| \mathbf{z}^T \| \mathbf{y}_3^T)$
$\mathbf{A}_1 \leftarrow \mathbf{B} - \mathbf{G}$
$pk \leftarrow (\mathbf{A}_1, \mathbf{A}_3, \mathbf{A}_3)$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathtt{find}, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{m}^* \leftarrow \mathfrak{m}_b$
$\mathbf{y}_2 \leftarrow \mathbf{z} + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*)$
$\mathfrak{c}^* \leftarrow (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
$b' \leftarrow \mathcal{A}(\mathtt{guess}, \mathsf{st}, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

We will now analyze the distinguishing advantage of $\mathcal{D}_2$. First assume that $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{As} + \mathbf{e})$. When splitting up $\mathbf{A}^T = (\mathbf{B}_1^T \| \mathbf{A}_2^T \| \mathbf{A}_3^T)$ and

$\mathbf{y}^T = (\mathbf{y}_1^T \| \mathbf{z}^T \| \mathbf{y}_3^T)$ this yields

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{B}\mathbf{s} + \mathbf{e}_1 \\ \mathbf{z} &= \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 \\ \mathbf{y}_3 &= \mathbf{A}_3\mathbf{s} + \mathbf{e}_3 \end{aligned}$$

with uniformly random $\mathbf{A}_1 \in \mathbb{F}_2^{m_1 \times n}$, $\mathbf{B} \in \mathbb{F}_2^{m_2 \times n}$ and $\mathbf{A}_3 \in \mathbb{F}_2^{m_3 \times n}$ and with $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ chosen according to $\mathbf{e}_1 \leftarrow_\$ \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \leftarrow_\$ \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$ and $\mathbf{e}_3 \leftarrow_\$ \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$. As $\mathbf{B}$ is distributed uniformly random, so is $\mathbf{A}_2 = \mathbf{B} - \mathbf{G}$. The challenge ciphertext $\mathfrak{c}^*$ is of the form

$$\begin{aligned} \mathfrak{c}^* &= (\mathbf{y}_1, \mathbf{z} + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*), \mathbf{y}_2) \\ &= (\mathbf{B}\mathbf{s} + \mathbf{e}_1, \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*), \mathbf{A}_3\mathbf{s} + \mathbf{e}_3) \\ &= (\mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathbf{G}\mathbf{s}, \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*), \mathbf{A}_3\mathbf{s} + \mathbf{e}_3) \\ &= (\mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s}), \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*), \mathbf{A}_3\mathbf{s} + \mathbf{e}_3) \end{aligned}$$

Thus the distribution of pk and $\mathfrak{c}^*$ are the same as in **Game 1** and we get

$$\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) = 1] = \Pr[\mathbf{Game1}(\mathcal{A}) = 1].$$

Now assume that $\mathcal{D}_2$'s input is of the form $(\mathbf{A}, \mathbf{u})$. Again, splitting up $\mathbf{A}^T = (\mathbf{B}^T \| \mathbf{A}_2^T \| \mathbf{A}_3^T)$ and $\mathbf{y}^T = (\mathbf{y}_1^T \| \mathbf{z}^T \| \mathbf{y}_3^T)$ yields that since $\mathbf{B}$ is uniformly distributed and so is $\mathbf{A}_2 = \mathbf{B} - \mathbf{G}$. The ciphertext $\mathfrak{c}^*$ now has the form

$$\mathfrak{c}^* = (\mathbf{u}_1, \mathbf{u}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*), \mathbf{u}_3) = (\mathbf{u}_1, \mathbf{u}_2', \mathbf{u}_3).$$

As $\mathbf{u}_2$ is uniformly distributed, so is $\mathbf{u}_2' = \mathbf{u}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m}^*)$. Thus, pk and $\mathfrak{c}^*$ have the same distribution as in **Game 2**. We get

$$\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{u}) = 1] = \Pr[\mathbf{Game2}(\mathcal{A}) = 1].$$

We conclude

$$\begin{aligned} \mathsf{Adv}_{\mathsf{DLPN}}(\mathcal{D}_2) &= |\Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) = 1] - \Pr[\mathcal{D}_2(\mathbf{A}, \mathbf{u}) = 1]| \\ &= |\Pr[\mathbf{Game1}(\mathcal{A}) = 1] - \Pr[\mathbf{Game2}(\mathcal{A}) = 1]| \\ &> \epsilon, \end{aligned}$$

which contradicts the hardness of the $\mathsf{DLPN}(n_1, m_1 + m_2 + m_3, \rho)$ problem.

All together, we get that

$$\mathsf{Adv}_{IND-CPA}(\mathcal{A}) \leq \mathsf{Adv}_{\mathbf{Game2}}(\mathcal{A}) + \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda)$$

which concludes the proof. □

6.2.3. Alternative Decryption

As in Section 5.3.3, we will first provide an alternative decryption algorithm Dec^* for $\mathsf{PKE}_{\mathsf{LPN}}$. We will require Dec^* to produce the same output as $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$, even if Dec^* has only an incomplete secret key at its disposal. Assume therefore that the trapdoor $\mathbf{W} \in \mathbb{F}_2^{m_3 \times m_1}$ consists of k_1 column blocks of size k_2, i.e.$m_1 = k_1 \cdot k_2$ and

$$\mathbf{W} = (\mathbf{W}_1 \| \dots \| \mathbf{W}_{k_1}),$$

where $\mathbf{W}_i \in \mathbb{F}_2^{m_3 \times k_2}$. This is in direct analogy to the case of McEliece in Section 5.3.3, where instead of *partial trapdoors* $\mathbf{W}_i$ we have full McEliece trapdoors. Assume now that in an incomplete secret key $\tilde{sk}$ several $\mathbf{W}_i$ are missing. Consequently, Dec^* will not be able to recover the corresponding blocks from a ciphertext and has to declare an erasure. Thus, in order to recover the secret $\mathbf{s}$, we need a decoder for C_1 that deals with both bit-errors and block erasures in codewords. A block erasure deletes any of the k_1 blocks entirely, while bit-errors appear on bit-level. However, an important observation here is that we do not need unique decoding. Assume for a moment that we have a decoder that outputs a short list of candidates for $\mathbf{s}$.

We have crafted the encryption scheme $\mathsf{PKE}_{\mathsf{LPN}}$ such that, given a ciphertext $\mathfrak{c} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ and a possible secret $\mathbf{s}$, we can verify whether $\mathbf{s}$ was the secret used to encrypt $\mathfrak{c}$. More specifically, the decryption algorithm $\mathsf{PKE}_{\mathsf{LPN}}$ computes $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ and only generates an output if $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ have the proper Hamming weight. Therefore, we can use a list-decoder for the code C_1 to obtain a short list of candidates for $\mathbf{s}$ and then test which candidate is the right one.

Construction 6.2. *Let $m_1 = k_1 \cdot k_2$. Assume that there exists an efficient list decoder $\mathsf{C}_1.\mathsf{ListDecode}$ that list-decodes up to t_1 bit errors and l block erasures for blocks of length k_2. The alternative decryption algorithm Dec^* is given as follows.*

$\mathsf{Dec}^*(\tilde{sk}, \mathfrak{c})$*:*
  Parse $\tilde{sk} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_{k_1})$
  Parse $\mathfrak{c} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$
  Parse $\mathbf{y}_1^T = (\mathbf{y}_{1,1}^T \| \ldots \| \mathbf{y}_{1,k_1}^T)$
  For $i = 1, \ldots, k_1$
    If $\tilde{\mathbf{W}}_i \neq \perp$
      $\mathbf{z}_i \leftarrow \mathbf{y}_{1,i} - \tilde{\mathbf{W}}_i^T \mathbf{y}_3$
    Otherwise
      $\mathbf{z}_i \leftarrow \perp$
  $\mathbf{z} \leftarrow (\mathbf{z}_1, \ldots, \mathbf{z}_k)$
  $L \leftarrow \mathsf{C}_1.\mathsf{ListDecode}(\mathbf{z})$
  For $\mathbf{s} \in L$
    $\mathbf{x} \leftarrow \mathbf{y}_2 - \mathbf{A}_2\mathbf{s}$
    $\mathfrak{m} \leftarrow \mathsf{C}_2.\mathsf{Decode}(\mathbf{x})$
    $\mathbf{e}_1 \leftarrow \mathbf{y}_1 - \mathbf{A}_1\mathbf{s} - \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})$
    $\mathbf{e}_2 \leftarrow \mathbf{y}_1 - \mathbf{A}_2\mathbf{s} - \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})$
    $\mathbf{e}_3 \leftarrow \mathbf{y}_3 - \mathbf{A}_3\mathbf{s}$
    If $\mathsf{wgt}(\mathbf{e}_1) = \lfloor \rho m_1 \rfloor$ *and* $\mathsf{wgt}(\mathbf{e}_2) = \lfloor \rho m_2 \rfloor$ *and* $\mathsf{wgt}(\mathbf{e}_3) = \lfloor \rho m_3 \rfloor$
      Return $\mathfrak{m}$
  Return $\perp$

We will now show that Dec^* decrypts correctly.

Lemma 6.2. *Assume that the decoder $\mathsf{C}_1.\mathsf{ListDecode}$ can simultaneously handle t_1 errors and l block-erasures. Fix a public key pk together with a private key $sk = \mathbf{W}$ such that it holds for all $\mathbf{e}_1 \in \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$ and $\mathbf{e}_3 \in \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$ that $\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}^T\mathbf{e}_3) \leq t_1$. Split $\mathbf{W} = (\mathbf{W}_1 \| \ldots \| \mathbf{W}_{k_1})$. Let $\tilde{sk} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_{k_1})$ be such that $(\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_{k_1})$ contains at most l erasures but is otherwise identical to $(\mathbf{W}_1, \ldots, \mathbf{W}_{k_1})$. Then it holds for every $\mathfrak{c} \in \mathbb{F}_2^{m_1+m_2+m_3}$ (i.e. every $\mathfrak{c}$ that could syntactically be a ciphertext) that $\mathsf{Dec}^*(\tilde{sk}, \mathfrak{c}) = \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}(sk, \mathfrak{c})$.*

Proof. Let $\mathfrak{c} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$. One of the following two cases must occur.

1. There exist $\mathbf{s} \in \mathbb{F}_2^{n_1}$, $\mathfrak{m} \in \mathbb{F}_2^{n_2}$ and $\mathbf{e}_1 \in \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$, $\mathbf{e}_2 \in \mathsf{S}_{m_2}(\lfloor \rho m_2 \rfloor)$, $\mathbf{e}_3 \in \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$ such that

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{A}_1\mathbf{s} + \mathbf{e}_1 + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})\\ \mathbf{y}_2 &= \mathbf{A}_2\mathbf{s} + \mathbf{e}_2 + \mathsf{C}_2.\mathsf{Encode}(\mathfrak{m})\\ \mathbf{y}_3 &= \mathbf{A}_3\mathbf{s} + \mathbf{e}_3\end{aligned}$$

2. No such $\mathbf{s}$, $\mathfrak{m}$, $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ exist.

Case 2 is almost trivial, as both $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$ and Dec^* only output $\mathfrak{m}$ if they have found $\mathbf{s}$, $\mathfrak{m}$, $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ with the above property. If they do not exists, both $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$ and Dec^* will output $\perp$.

We will now consider case 1. Fix an index $i \in \{1, \ldots, k_1\}$. If $\tilde{\mathbf{W}}_i = \perp$, then $\mathbf{z}_i = \perp$. If $\tilde{\mathbf{W}}_i \neq \perp$, then we are guaranteed that $\tilde{\mathbf{W}}_i = \mathbf{W}_i$. Thus

$$\begin{aligned}\mathbf{z}_i &= \mathbf{y}_{1,i} - \tilde{\mathbf{W}}_i^T\mathbf{y}_3\\ &= \mathbf{A}_{1,i}\mathbf{s} + \mathbf{e}_{1,i} + \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})_i - \mathbf{W}_i^T(\mathbf{A}_3\mathbf{s} + \mathbf{e}_3)\\ &= \mathsf{C}_1.\mathsf{Encode}(\mathbf{s})_i + \mathbf{e}_{1,i} - \mathbf{W}_i^T\mathbf{e}_3.\end{aligned}$$

which states that $\mathbf{z}_i$ is the i-th block of $\mathsf{C}_1.\mathsf{Encode}(\mathbf{s}) + \mathbf{e}_1 + \mathbf{W}^T\mathbf{e}_3$. As $\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}^T\mathbf{e}_3) \leq t_1$, the vector $\mathbf{z}$ contains at most t_1 bit errors. Moreover, $\mathbf{z}$ contains at most l block erasures. Thus, as $\mathsf{C}_1.\mathsf{ListDecode}$ can list decode up to t_1 errors and l block erasures, the list L will contain $\mathbf{s}$ an we get that Dec^* and $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$ both output $\mathfrak{m}$. This concludes the proof. □

6.3. Tag-Based Encryption from the LPN Assumption

We will now construct a tag-based encryption scheme $\mathsf{TBE}_{\mathsf{LPN}}$ from the public key encryption scheme $\mathsf{PKE}_{\mathsf{LPN}}$ constructed in Section 6.2. The only extra ingredient for $\mathsf{TBE}_{\mathsf{LPN}}$ is a mechanism that derives tag-specific public and secret keys. Like in Section 5.4, the derivation mechanism will select several components of the full keys, obtaining tag-specific derived keys.

Construction 6.3. *Let λ be a security parameter, $m_2, m_3, n_1, n_2, k_1, k_2, r = \mathsf{poly}(\lambda)$, $m_1 = k_1k_2$, $\rho = \rho(\lambda) \in (0,1)$ and $q = \mathsf{poly}(n)$ be a prime-power. Let $\mathsf{C}_{\mathsf{tag}}$ be a q-ary linear $[k_1, r]$ code. The tag-based encryption scheme $\mathsf{TBE}_{\mathsf{LPN}} = (\mathsf{TBE}_{\mathsf{McE}}.\mathsf{KeyGen}, \mathsf{TBE}_{\mathsf{LPN}}.\mathsf{Enc}, \mathsf{TBE}_{\mathsf{LPN}}.\mathsf{Dec})$ is specified as follows.*

- $\mathsf{TBE}_{\mathsf{LPN}}.\mathsf{KeyGen}(1^\lambda)$*:*
  $\mathbf{A}_2 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
  $\mathbf{A}_3 \leftarrow_\$ \mathbb{F}_2^{m_3 \times n_1}$
  For $i = 1, \ldots, k_1$ and $a \in \mathbb{F}_q$
  $\mathbf{W}_{i,a} \leftarrow \mathsf{Ber}(m_3 \times k_2, \rho)$
  $\mathbf{A}_{1,i,a} \leftarrow \mathbf{W}_{i,a}^T\mathbf{A}_3$
  $pk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3)$
  $sk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
  Return (pk, sk)

- $\mathsf{TBE}_{\mathsf{LPN}}.\mathsf{Enc}(pk, \tau, \mathfrak{m})$:
  Parse $pk = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3)$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_{k_1}) \in \mathbb{F}_q^{k_1}$
  $\mathbf{A}_{1,\tau} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1}^T \| \ldots, \| \mathbf{A}_{1,k_1,\hat{\tau}_{k_1}}^T)^T$
  $pk_\tau \leftarrow (\mathbf{A}_{1,\tau}, \mathbf{A}_2, \mathbf{A}_3)$
  $\mathfrak{c} \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}(pk_\tau, \mathfrak{m})$
  Return $\mathfrak{c}$

- $\mathsf{TBE}_{\mathsf{LPN}}.\mathsf{Dec}(sk, \tau, \mathfrak{c})$:
  Parse $sk = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_{k_1}) \in \mathbb{F}_q^{k_1}$
  $\mathbf{A}_{1,\tau} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1}^T \| \ldots, \| \mathbf{A}_{1,k_1,\hat{\tau}_{k_1}}^T)^T$
  $\mathbf{W}_\tau \leftarrow (\mathbf{W}_{1,\hat{\tau}_1} \| \ldots \| \mathbf{W}_{k_1,\hat{\tau}_{k_1}})$
  $sk_\tau \leftarrow (\mathbf{A}_{1,\tau}, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W}_\tau)$
  $\mathfrak{m} \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}(sk_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

The plaintext-space of $\mathsf{TBE}_{\mathsf{LPN}}$ *is* $\mathfrak{M}_{\mathsf{LPN},\lambda} = \mathbb{F}_2^{n_2}$*, the ciphertext space is* $\mathfrak{C}_{\mathsf{LPN},\lambda} = \mathbb{F}_2^{m_1+m_2+m_3}$ *and the tag-space is* $\mathfrak{T}_{\mathsf{LPN},\lambda} = \mathbb{F}_q^r$*.*

6.3.1. Completeness

Before we show that $\mathsf{TBE}_{\mathsf{LPN}}$ is complete, we will first prove a lemma which states that with overwhelming probability over the choice of the secret key sk, all derived secret keys sk_τ will only account for an error that is bounded by $(1+\gamma)q\rho^2 mk$.

Lemma 6.3. *Let* $\beta > 0$*. Let* $sk = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$ *be a secret key generated by* $\mathsf{TBE}_{\mathsf{LPN}}.\mathsf{KeyGen}$*. Then it holds for all* $\mathbf{e}_1 \in \mathsf{S}_{m_1}(\lfloor \rho m_1 \rfloor)$ *and* $\mathbf{e}_3 \in \mathsf{S}_{m_3}(\lfloor \rho m_3 \rfloor)$ *and tags* τ *that* $\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}_\tau^T \mathbf{e}_3) \leq (1+\beta)q\rho^2 m_1 m_3$*, except with negligible probability over the coins used by* $\mathsf{TBE}_{\mathsf{LPN}}.\mathsf{KeyGen}$*.*

Proof. Let $\mathbf{W} = \|_{i\in\{1,\ldots,k\}, a\in\mathbb{F}_q} \mathbf{W}_{i,a} \in \mathbb{F}_2^{m_3 \times qm_1}$ be a matrix that is the horizontal concatenation of all $\mathbf{W}_{i,a}$. Then $\mathbf{W}$ is distributed according to $\mathsf{Ber}(m_3 \times qm_1, \rho)$. Now it holds by Lemma 2.11 that $\|\mathbf{W}^T\|_{\mathsf{wgt}} \leq (1 + \beta - \frac{1}{\rho q m_3})\rho q m_1$, except with negligible probability. The matrix $\mathbf{W}_\tau$ is a column-submatrix of $\mathbf{W}$, thus it holds for all τ that $\|\mathbf{W}_\tau^T\|_{\mathsf{wgt}} \leq \|\mathbf{W}^T\|_{\mathsf{wgt}} \leq (1 + \beta - \frac{1}{\rho q m_3})\rho q m_1$. As $\mathsf{wgt}(\mathbf{e}_3) = \rho m_3$ and $\mathsf{wgt}(\mathbf{e}_1) = \rho m_1$, it holds that

$$\begin{aligned}
\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}_\tau^T \mathbf{e}_3) &\leq \mathsf{wgt}(\mathbf{e}_1) + \|\mathbf{W}_\tau^T\|_{\mathsf{wgt}} \mathsf{wgt}(\mathbf{e}_3) \\
&\leq \rho m_1 + (1 + \beta - \frac{1}{\rho q m_3})\rho m_1 \cdot \rho m_3 \\
&= (1+\beta) q \rho^2 m_1 m_3.
\end{aligned}$$

□

A direct corollary from Lemma 6.3 is that $\mathsf{TBE}_{\mathsf{LPN}}$ fulfills the completeness requirement.

Corollary 6.4. $\mathsf{TBE}_{\mathsf{LPN}}$ *is complete, given that* $\mathsf{C}_1.\mathsf{Decode}$ *can handle at least* $t_1 \geq (1+\beta)q\rho^2 m_1 m_3$ *errors.*

6.3.2. IND-STAG-CCA2 Security

We are now ready to show that $\mathsf{TBE}_{\mathsf{LPN}}$ is IND-STAG-CCA2 secure.

Theorem 6.2. $\mathsf{TBE}_{\mathsf{LPN}}$ *is IND-STAG-CCA2 secure, given that* $\mathsf{PKE}_{\mathsf{LPN}}$ *is IND-CPA secure,* $\mathsf{C}_{\mathsf{tag}}$ *has minimum distance at least* $k_1 - l$ *and* $\mathsf{C}_1.\mathsf{ListDecode}$ *can efficiently simultaneously decode* t_1 *errors and* l *erasures.*

Proof. Let $\mathcal{A}$ be a PPT-adversary with advantage ϵ against the IND-STAG-CCA2 security of $\mathsf{TBE}_{\mathsf{LPN}}$. We will construct an adversary $\mathcal{A}'$ with advantage $\epsilon' \geq \epsilon - \mathsf{negl}(\lambda)$ against the IND-CPA security of $\mathsf{PKE}_{\mathsf{LPN}}$. Consider the following experiments **Game 0** and **Game 1**.

Game 0

$(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathtt{init}, 1^\lambda)$
$\mathbf{A}_2 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
$\mathbf{A}_3 \leftarrow_\$ \mathbb{F}_2^{m_2 \times n_1}$
For $i = 1, \ldots, k_1$ and $a \in \mathbb{F}_q$
  $\mathbf{W}_{i,a} \leftarrow \mathsf{Ber}(m_3 \times k_2, \rho)$
  $\mathbf{A}_{1,i,a} \leftarrow \mathbf{W}_{i,a}^T \mathbf{A}_3$
$pk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3)$
$sk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \cdot, \cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\hat{\tau}^* \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau^*)$
Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_{k_1}^*)$
$\mathbf{A}_{1,\tau^*} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1^*}^T \| \ldots, \| \mathbf{A}_{1,k_1,\hat{\tau}_{k_1}^*}^T)^T$
$pk_{\tau^*} \leftarrow (\mathbf{A}_{1,\tau^*}, \mathbf{A}_2, \mathbf{A}_3)$
$\mathfrak{c}^* \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}(pk_{\tau^*}, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \cdot, \cdot)}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(sk, \tau^*, \tau, \mathfrak{c})$
If $\tau = \tau^*$
  Return $\perp$
Parse $sk = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
$\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_{k_1})$
$\mathbf{A}_{1,\tau} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1}^T \| \ldots \| \mathbf{A}_{1,k_1,\hat{\tau}_{k_1}}^T)^T$
$\mathbf{W}_\tau \leftarrow (\mathbf{W}_{1,\hat{\tau}_1} \| \ldots \| \mathbf{W}_{k,\hat{\tau}_{k_1}})$
$sk_\tau \leftarrow (\mathbf{A}_{1,\tau}, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W}_\tau)$
$\mathfrak{m} \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}(sk_\tau, \mathfrak{c})$
Return $\mathfrak{m}$

Clearly, **Game 0** is the IND-STAG-CCA2 experiment for $\mathsf{TBE}_{\mathsf{LPN}}$, for which we have substituted $\mathsf{TBE.KeyGen}$, $\mathsf{TBE.Enc}$ and $\mathsf{TBE.Dec}$ with their implementations according to $\mathsf{TBE}_{\mathsf{LPN}}$.

In **Game 1**, the following changes have been made compared with **Game 0**. First, the generation of the keys corresponding to the challenge tag τ^* has been moved to the beginning of the experiment. Moreover, the secret key corresponding to the challenge tag is not included in the secret key $\tilde{sk}$, instead block-erasures are set in the corresponding blocks of the secret key. Finally, the decryption oracle $\mathcal{O}_{\mathsf{Dec}}$ now uses the alternative decryption algorithm Dec^* instead of $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$ to deal with the erasures in the secret key.

Claim: We claim that it holds that $|\Pr[\mathbf{Game0}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1]| \leq \mathsf{negl}(\lambda)$, given that $\mathsf{C}_{\mathsf{tag}}$ has minimum distance at least $k_1 - l$ and $\mathsf{C.Decode}$ can efficiently simultaneously decode t_1 errors and l erasures.

First observe that the distribution of the public key pk is identical in both experiments. Moreover, the challenge-ciphertext $\mathfrak{c}^*$ is also computed in the same way in both experiments. If we condition to the event that in both experiments the

Game 1

$(pk^*, sk^*) \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{KeyGen}(1^\lambda)$
Parse $pk^* = (\mathbf{A}_1^*, \mathbf{A}_2, \mathbf{A}_3)$
$(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\mathtt{init}, 1^\lambda)$
$\hat{\tau}^* \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau^*)$
Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_{k_1}^*)$
Parse $\mathbf{A}_1^* = ({\mathbf{A}_{1,1}^*}^T \| \ldots \| {\mathbf{A}_{1,k_1}^*}^T)^T$
For $i = 1, \ldots, k_1$ and $a \in \mathbb{F}_q$
  If $a = \tau_i^*$
    $\mathbf{W}_{i,a} \leftarrow \perp$
    $\mathbf{A}_{1,i,a} \leftarrow \mathbf{A}_{1,i}^*$
  Otherwise
    $\mathbf{W}_{i,a} \leftarrow \mathsf{Ber}(m_3 \times k_2, \rho)$
    $\mathbf{A}_{1,i,a} \leftarrow \mathbf{W}_{i,a}^T \mathbf{A}_3$
$pk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3)$
$\tilde{sk} \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
$(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \cdot, \cdot)}(\mathtt{find}, \mathsf{st}_0, pk)$
$b \leftarrow_\$ \{0, 1\}$
$\mathfrak{c}^* \leftarrow \mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Enc}(pk^*, \mathfrak{m}_b)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \cdot, \cdot)}(\mathtt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
Return 1 iff $b = b'$.

$\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \tau, \mathfrak{c})$
  If $\tau = \tau^*$
    Return $\perp$
  Parse $\tilde{sk} = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
  $\hat{\tau} \leftarrow \mathsf{C}_{\mathsf{tag}}.\mathsf{Encode}(\tau)$
  Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_{k_1})$
  $\mathbf{A}_{1,\tau} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1}^T \| \ldots \| \mathbf{A}_{1,k,\hat{\tau}_{k_1}}^T)^T$
  $\mathbf{W}_\tau \leftarrow (\mathbf{W}_{1,\hat{\tau}_1} \| \ldots \| \mathbf{W}_{k,\hat{\tau}_{k_1}})$
  $\tilde{sk}_\tau \leftarrow (\mathbf{A}_{1,\tau}, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W}_\tau)$
  $\mathfrak{m} \leftarrow \mathsf{Dec}^*(\tilde{sk}_\tau, \mathfrak{c})$
  Return $\mathfrak{m}$

decryption oracle behaves identically, then **Game 0** and **Game 1** are identically distributed from the view of $\mathcal{A}$.

Lemma 6.3 states that with overwhelming probability sk is such that it holds for all $\mathbf{e}_1$, $\mathbf{e}_3$ and τ that $\mathsf{wgt}(\mathbf{e}_1 - \mathbf{W}_\tau^T \mathbf{e}_3) \leq (1+\beta) q \rho^2 m_1 m_3 \leq t_1$. Moreover, since $\mathsf{C}_{\mathsf{tag}}$ has minimum-distance $k_1 - l$, it holds for every tag $\tau \neq \tau^*$ that $\mathsf{wgt}(\hat{\tau} - \hat{\tau}^*) \geq k_1 - l$ and thus $\hat{\tau}$ and $\hat{\tau}^*$ agree on at most l locations. But this means that $\tilde{sk}_\tau$ contains at most l erasures. Now we can conclude using Lemma 6.2 that the behavior of Dec^* is identical to the behavior of $\mathsf{PKE}_{\mathsf{LPN}}.\mathsf{Dec}$. Thus

$$|\Pr[\mathbf{Game0}(\mathcal{A}) = 1] - \Pr[\mathbf{Game1}(\mathcal{A}) = 1]| \leq \mathsf{negl}(\lambda)$$

follows.

Claim: It holds that $\mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A}) \leq \mathsf{negl}(\lambda)$, given that $\mathsf{PKE}_{\mathsf{LPN}}$ is IND-CPA secure.

We will now use $\mathcal{A}$ to construct an IND-CPA adversary $\mathcal{A}'$ against $\mathsf{PKE}_{\mathsf{LPN}}$. $\mathcal{A}'$ is given as follows.

We will now show that $\mathsf{Adv}_{IND-CPA}(\mathcal{A}')) = \mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A})$. Observe that the distribution of pk^* is identical in **Game 1** and the IND-CPA experiment. The only difference is that key generation has been moved into the IND-CPA experiment. The same is true for the challenge-ciphertext $\mathfrak{c}^*$, the computation of $\mathfrak{c}^*$ is moved into the IND-CPA experiment, but is otherwise identical. Thus the view of $\mathcal{A}$ is identical

Adversary $\mathcal{A}'$

$\mathbf{A}'(\texttt{find}, pk^*)$
- Parse $pk^* = (\mathbf{A}_1^*, \mathbf{A}_2, \mathbf{A}_3)$
- $(\tau^*, \mathsf{st}_0) \leftarrow \mathcal{A}(\texttt{init}, 1^\lambda)$
- $\hat{\tau}^* \leftarrow \mathsf{C_{tag}}.\mathsf{Encode}(\tau^*)$
- Parse $\hat{\tau}^* = (\hat{\tau}_1^*, \ldots, \hat{\tau}_{k_1}^*)$
- Parse $\mathbf{A}_1^* = ({\mathbf{A}_{1,1}^*}^T \| \ldots \| {\mathbf{A}_{1,k_1}^*}^T)^T$
- For $i = 1, \ldots, k_1$ and $a \in \mathbb{F}_q$
  - If $a = \tau_i^*$
    - $\mathbf{W}_{i,a} \leftarrow \perp$
    - $\mathbf{A}_{1,i,a} \leftarrow \mathbf{A}_{1,i}^*$
  - Otherwise
    - $\mathbf{W}_{i,a} \leftarrow \mathsf{Ber}(m_3 \times k_2, \rho)$
    - $\mathbf{A}_{1,i,a} \leftarrow \mathbf{W}_{i,a}^T \mathbf{A}_3$
- $pk \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3)$
- $\tilde{sk} \leftarrow ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
- $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st}_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \cdot, \cdot)}(\texttt{find}, \mathsf{st}_0, pk)$
- $\mathsf{st} \leftarrow (\tilde{sk}, \mathsf{st}_1)$
- Return $(\mathfrak{m}_0, \mathfrak{m}_1, \mathsf{st})$

$\mathbf{A}'(\texttt{guess}, \mathsf{st}, \mathfrak{c}^*)$
- Parse $\mathsf{st} = (\tilde{sk}, \mathsf{st}_1)$
- $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \cdot, \cdot)}(\texttt{guess}, \mathsf{st}_1, \mathfrak{c}^*)$
- Return b'

$\mathcal{O}_{\mathsf{Dec}}(\tilde{sk}, \tau^*, \tau, \mathfrak{c})$
- If $\tau = \tau^*$
  - Return $\perp$
- Parse $\tilde{sk} = ((\mathbf{A}_{1,i,a})_{i,a}, \mathbf{A}_2, \mathbf{A}_3, (\mathbf{W}_{i,a})_{i,a})$
- $\hat{\tau} \leftarrow \mathsf{C_{tag}}.\mathsf{Encode}(\tau)$
- Parse $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_{k_1})$
- $\mathbf{A}_{1,\tau} \leftarrow (\mathbf{A}_{1,1,\hat{\tau}_1}^T \| \ldots \| \mathbf{A}_{1,k,\hat{\tau}_{k_1}}^T)^T$
- $\mathbf{W}_\tau \leftarrow (\mathbf{W}_{1,\hat{\tau}_1} \| \ldots \| \mathbf{W}_{k,\hat{\tau}_{k_1}})$
- $\tilde{sk}_\tau \leftarrow (\mathbf{A}_{1,\tau}, \mathbf{A}_2, \mathbf{A}_3, \mathbf{W}_\tau)$
- $\mathfrak{m} \leftarrow \mathsf{Dec}^*(\tilde{sk}_\tau, \mathfrak{c})$
- Return $\mathfrak{m}$

in **Game 1** and $\mathcal{A}'$'s simulation. Consequently, it holds that $\mathsf{Adv}_{IND-CPA}(\mathcal{A}') = \mathsf{Adv}_{\mathbf{Game1}}(\mathcal{A}) = \epsilon - \mathsf{negl}(\lambda)$, which is non-negligible. Thus, $\mathcal{A}'$ breaks the IND-CPA security of $\mathsf{PKE_{LPN}}$ with non-negligible advantage, which yields the desired contradiction. This concludes the proof. □

6.4. Instantiations

We will now discuss instantiations of $\mathsf{TBE_{LPN}}$ for concrete choices of the codes involved. We will therefore discuss the performance of different codes and decoders. Let $n_1, n_2, m_1, m_2, m_3 = O(n)$ for a parameter n.

By Theorems 6.1 and 6.2, the security of $\mathsf{TBE_{LPN}}$ relies on the hardness of the problems $\mathsf{DLPN}(n_1, m_1 + m_2 + m_3, M)$ and $\mathsf{DDMLPN}(n_1, m_1, m_2, \mathsf{Ber}(m_1 \times m_2, \rho))$. Both problems in turn are based on $\mathsf{LPN}(n', m', \mathsf{Ber}(m', \rho))$ for some $n', m' = O(n)$. By the discussion in Section 3.6 we need to choose $n = \Omega((\lambda/\log(\lambda))^2)$ so that $\mathsf{LPN}(n', m', \mathsf{Ber}(m', \rho))$ has (conjectured) hardness $2^{\Omega(\lambda)}$.

Moreover, in order to achieve completeness of the scheme, we need to be able to decode errors of weight $O(\rho^2 m_1 m_3) = O(\rho^2 n^2)$ by Corollary 6.4. Thus, ρ must be in $\rho = O(1/\sqrt{n})$ for the errors to be sufficiently low to be decodable.

We now need to find suitable instantiations for the codes $\mathsf{C}_1, \mathsf{C}_2$ and $\mathsf{C_{tag}}$. Recall that our only constraint to the binary code C_2 was that it needs to be able to efficiently correct $\rho m_2 = O(\sqrt{n})$ errors, which is sub-linear in n. This allows us to choose C_2 to be a code of very high rate. A natural choice for C_2 are binary Goppa

codes (c.f. Section 2.5.6), as they are efficiently decodable and meet the Gilbert Varshamov bound. Thus, the rate of C_2 can be as high as $1 - O(\log(n)/\sqrt{n})$.

It remains to find suitable instantiations for C_1 and $\mathsf{C}_{\mathsf{tag}}$. Ideally, we would like to choose the codes C_1 and $\mathsf{C}_{\mathsf{tag}}$ in a way such that the noise rate ρ can be as high as possible, as higher noise means more security (in a concrete sense). It will be more convenient to discuss the codes C_1 and $\mathsf{C}_{\mathsf{tag}}$ with respect to their rates and relative minimum distances. Thus

- let C_1 be a binary $[m_1, R_1 m_1, \delta_1 m_1]$ code.
- let $\mathsf{C}_{\mathsf{tag}}$ be a q-ary $[k_1, \frac{R_{\mathsf{tag}}}{\log(q)} k_1, \delta_{\mathsf{tag}} k_1]$ code.

In the following discussion we will omit issues such as the constraints that $R_1 m_1$, $\delta_1 m_1$, $\frac{R_{\mathsf{tag}}}{\log(q)} k_1$ and $\delta_{\mathsf{tag}} k_1$ must be integers. As $\mathsf{C}_{\mathsf{tag}}$ may be a q-ary code (for a $q \geq 2$), we will consider $\mathsf{C}_{\mathsf{tag}}$ with respect to its *equivalent binary rate* $\mathsf{R}_{\mathsf{tag}}$, i.e. the rate of $\mathsf{C}_{\mathsf{tag}}$ is $R_{\mathsf{tag}}/\log(q)$. We do this to make the rates for different alphabet sizes q comparable. Recall that the size of the tag-space $\mathfrak{T}_{\mathsf{TBE}}$ is

$$|\mathfrak{T}_{\mathsf{LPN},\lambda}| = q^r = q^{\frac{R_{\mathsf{tag}}}{\log(q)} \cdot k_1} = 2^{R_{\mathsf{tag}} k_1}.$$

Thus, the equivalent binary rate R_{tag} represents the size of the tag-space, independent of the alphabet size q. We will consider different families of codes for C_1 and decoders for C_1 and each time proceed as follows. For such a choice of C_1 and a decoder for C_1, we want to set the *free* parameters of the construction in a way such that the tolerable noise rate ρ is maximal, as a function of the rate R_1 of C_1 and R_{tag} of $\mathsf{C}_{\mathsf{tag}}$. Recall that by Theorem 6.2 the list decoder $\mathsf{C}_1.\mathsf{ListDecode}$ needs to be able to handle $(1+\beta)q\rho^2 m_3 m_1$ bit errors and $l = k_1 - \delta_{\mathsf{tag}} k_1$ block erasures. Set

$$\eta = (1+\beta)q\rho^2 m_3$$

and

$$\sigma = 1 - \delta_{\mathsf{tag}}.$$

Thus, the decoder needs to handle an η fraction of bit errors and a σ fraction of block erasures. We will briefly consider the constraint $\delta_{\mathsf{tag}} = 1 - \sigma$ in more detail. If we want to choose $\mathsf{C}_{\mathsf{tag}}$ to be a binary code, then $\delta_{\mathsf{tag}} < \frac{1}{2}$. But this implies that $\sigma > \frac{1}{2}$. In other words, the list-decoder $\mathsf{C}_1.\mathsf{ListDecode}$ will have to deal with more than a $\frac{1}{2}$ fraction of block erasures. If we aspire for a block size $k_2 = 1$, then list decoding is strictly necessary, as no unique decoder can handle more than a $\frac{1}{2}$ fraction of (worst case) erasures.

Let $\delta_{max}(R, q)$ be the best (known) achievable minimum distance by an (efficiently constructible) q-ary code of equivalent binary rate R. Asymptotically, the best known (existential) bounds are the Gilbert Varshamov bound (Theorem 2.4) and the TVZ bound (Theorem 2.5). Combining both bounds yields that for every alphabet size q and every rate R there exists a q-ary code with minimum distance at least

$$\delta_{max}(R, q) = \max\left\{1 - R - \frac{1}{\sqrt{q} - 1}, H_q^{-1}(1 - R)\right\}.$$

We will choose the tag code $\mathsf{C}_{\mathsf{tag}}$ such that it has best possible minimum distance, i.e.

$$\delta_{\mathsf{tag}} = \delta_{max}(R_{\mathsf{tag}}/\log(q), q).$$

Recall that we do not need efficient decoding for $\mathsf{C}_{\mathsf{tag}}$.

6.4.1. Instantiation 1: Arbitrary Code C_1, Unique Decoder

We will start by providing the simple instantiation given in [DMQN12]. Let C_1 be a binary $[m_1, R_1 m_1, \delta_1 m_1]$ code for which we can efficiently decode any constant fraction $\alpha(R_1)$ of errors using a decoder $\mathsf{C}_1.\mathsf{Decode}$. We will not assume that C_1 is list decodable from errors and erasures. Instead, we will simply treat erasures as errors. We can thus also set the block size k_2 to $k_2 = 1$. As discussed above, this will make it necessary to choose the alphabet size q of $\mathsf{C}_{\mathsf{tag}}$ strictly larger than 2. For this instantiation, we will therefore treat q as a *free parameter*. The code C_1 may be an expander code (as suggested in [DMQN12], c.f. Section 2.5.8) or an asymptotically good concatenated code (c.f. Section 2.5.7). As we treat erasures as errors for this instantiation, we have the constraint

$$\eta + \sigma \leq \alpha(R_1).$$

As we want to maximize η, we set

$$\begin{aligned}\eta &= \alpha(R_1) - \sigma \\ &= \alpha(R_1) - 1 + \delta_{\mathsf{tag}} \\ &= \alpha(R_1) - 1 + \delta_{max}(q, R_{\mathsf{tag}}/\log(q)).\end{aligned}$$

Finding a best possible q analytically as a function of R_1 (or $\alpha(R_1)$) and R_{tag} is hopeless. We will thus first show that for every constants $\alpha(R_1) \in (0, 1/4)$ and $R_{\mathsf{tag}} \in (0, 1)$ there exists a q such that $\eta > 0$ is a constant. For large (but constant) q, it holds that

$$\delta_{max}(q, R_{\mathsf{tag}}/\log(q)) = 1 - \frac{R_{\mathsf{tag}}}{\log(q)} - \frac{1}{\sqrt{q} - 1},$$

as for such q the TVZ bound (Theorem 2.5) outperforms the Gilbert Varshamov bound (Theorem 2.4). Thus it holds that

$$\eta = \alpha(R_1) - \frac{R_{\mathsf{tag}}}{\log(q)} - \frac{1}{\sqrt{q} - 1}.$$

Consequently, by choosing q sufficiently large it holds that

$$\frac{R_{\mathsf{tag}}}{\log(q)} - \frac{1}{\sqrt{q} - 1} < \alpha(R_1),$$

which yields the desired, as η is now a positive constant. We have thus established that the scheme is instantiatable for every $\alpha(R_1) \in (0, 1/4)$ and $R_{\mathsf{tag}} \in (0, 1)$. However, choosing q as a large constant has severe side effects. For one, the size of the public and secret keys depends linearly on q. Even though q is a constant and thus asymptotically irrelevant, a large q would imply impractical key sizes. On the other hand, q also has a negative effect on the maximum tolerable noise rate ρ,

$$\rho = \sqrt{\frac{\eta}{(1+\beta) \cdot q \cdot m_3}}.$$

To get a handle on the effect of q on the noise rate, we can compare ρ with the maximum possible noise rate ρ' we could tolerate for the basic scheme $\mathsf{PKE}_{\mathsf{LPN}}$ when

instantiating it with C_1. Clearly, as $\mathsf{C}_1.\mathsf{Decode}$ can correct $\alpha(R_1)$ bit errors, it holds by Lemma 6.1 that

$$\rho' = \sqrt{\frac{\alpha(R_1)}{1+\beta} m_3}.$$

The fraction ρ/ρ' tells us by which factor we need to choose ρ smaller than ρ', or in other words, how much noise tolerance (and therefore concrete security) we lose compared to the basic scheme $\mathsf{PKE}_{\mathsf{LPN}}$. It holds that

$$\frac{\rho}{\rho'} = \frac{1}{\sqrt{q}} \cdot \sqrt{1 - \frac{\delta_{max}(q, R_{\mathsf{tag}}/\log(q))}{\alpha(R_1)}}.$$

Thus,

$$\rho \approx \frac{1}{\sqrt{q}} \rho',$$

which leads to unrealistically small ρ for very large q.

Figures 6.1 and 6.2 show plots of ρ/ρ' and the corresponding q as a function of $\alpha(R_1)$ and R_{tag}. The discontinuity in the plot of q (Figure 6.2) originates from the switch from the GV bound to the TVZ bound. A reasonable choice of parameters may be $\alpha(R_1) = 0.1$ and $R_{\mathsf{tag}} = 0.05$, for which $\rho/\rho' \approx 0.09$ and $q = 64$.

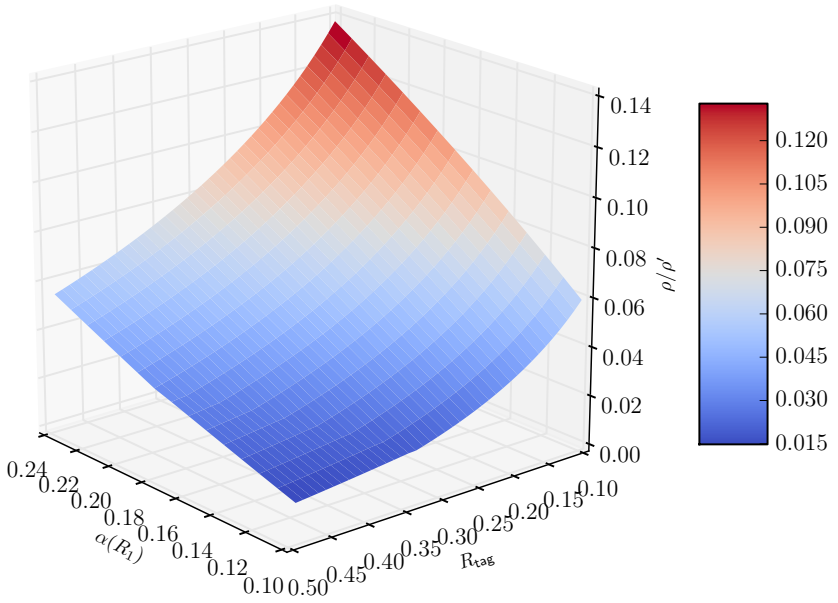

Figure 6.1.: Instantiation 1: The fraction ρ/ρ' as a function of $\alpha(R_1)$ and R_{tag}

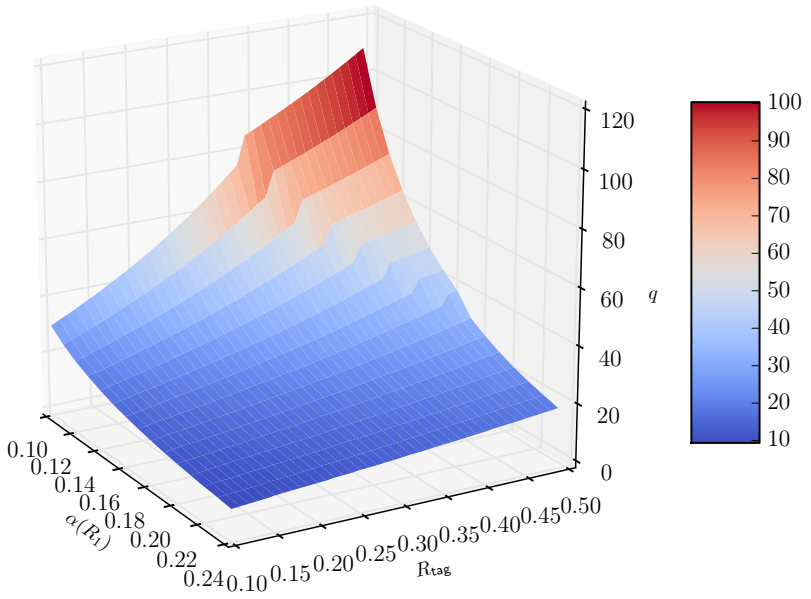

Figure 6.2.: Instantiation 1: The alphabet size q of $\mathsf{C}_{\mathsf{tag}}$ as a function of $\alpha(R_1)$ and R_{tag}

6.4.2. Instantiation 2: $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ with Unique Decoder, large Blocks

We have seen that a large (constant) q has several undesirable effects. We will therefore focus on the case $q = 2$ henceforth. This immediately implies that we can choose the code $\mathsf{C}_{\mathsf{tag}}$ by the Gilbert Varshamov ensemble, i.e. we have that

$$R_{\mathsf{tag}} = 1 - H(\delta_{\mathsf{tag}})$$

and therefore $\delta_{\mathsf{tag}} = H^{-1}(1 - R_{\mathsf{tag}})$. However, this will also require a more subtle treatment of erasures than just treating them like errors. In in the McEliece based construction of Chapter 5 we had a natural concatenated code structure with an outer Reed Solomon and an inner Goppa code. We will now take a similar approach by choosing the code C_1 as a concatenated code $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{C}_{out}$. As outer code we will choose a Reed Solomon code. As inner code we will choose a (short) binary code that meets the Gilbert Varshamov bound. We will now determine the block lengths. Let therefore $n_1 = c_1 c_2$ and $m_1 = k_1 k_2$ with $k_1 \geq c_1$ and $k_2 \geq c_2$. If we keep the block length $c_2 = O(\log \lambda)$, we can choose any binary linear code as C_{in} and still decode efficiently, as in this case $|\mathsf{C}_{\mathsf{in}}| \leq 2^{k_2} = \mathsf{poly}(\lambda)$. Thus, C_{in} can be chosen as a code that meets the Gilbert Varshamov bound and still be efficiently decoded (using brute force).

The size of the tag-space of this instantiation is

$$|\mathfrak{T}_{\mathsf{LPN},\lambda}| = 2^{R_{\mathsf{tag}} k_1} = 2^{R_{\mathsf{tag}} k_1} = 2^{R_{\mathsf{tag}} m_1 / k_1} = 2^{O(n/\log(n))}.$$

While this is asymptotically smaller than the tag space of instantiation 1, we still get a tag space of size $2^{\Omega(\lambda)}$ as $n = \Theta((\lambda/\log(\lambda))^2)$.

Let $R_{in} = k_2/c_2$ be the rate of C_{in} and δ_{in} its relative minimum distance. As C_{in} meets the Gilbert Varshamov bound it holds that $R_{in} = 1 - H(\delta_{in})$. Let R_{out} be the rate of the outer Reed Solomon code RS. The rate of $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ is

$$R_1 = R_{out} \cdot R_{in} = R_{out} \cdot (1 - H(\delta_{in})).$$

By Corollary 2.10 there exists an efficient decoder for C_1 decoding up to an η fraction of bit errors and a σ fraction of block erasures given that

$$\eta \leq \frac{1}{2} \delta_{in} (1 - R_{out} - \sigma).$$

Again, as we seek to maximize η, we set $\eta = \frac{1}{2}\delta_{in}(1 - R_{out})$. Solving the first equation for R_{out} and using that $\sigma = 1 - \delta_{\mathsf{tag}} = 1 - H^{-1}(1 - R_{\mathsf{tag}})$ yields

$$\eta = \frac{1}{2} \delta_{in} \left(H^{-1}(1 - R_{\mathsf{tag}}) - \frac{R_1}{1 - H(\delta_{in})} \right).$$

Moreover, it follows from $R_{out} < 1$ that $1 - H(\delta_{in}) > R_1$ and this yields the constraint $\delta_{in} < H^{-1}(1 - R_1)$. Now, δ_{in} is the only undetermined variable left on which η depends. Thus, we can maximize η as a function of δ_{in} subject to the constraint that $\delta_{in} < H^{-1}(1 - R_1)$. Clearly, also for this problem an analytical solution is hopeless. However, notice that for every R_{tag} we can find an R_1 and δ_{in} such that $\eta > 0$. Namely we can choose $\delta_{in} < \frac{1}{2}$ arbitrary and set

$$R_1 = \frac{H^{-1}(1 - R_{\mathsf{tag}}) \cdot (1 - H(\delta_{in}))}{2} > 0,$$

which yields

$$\eta = \frac{1}{4}\delta_{in}(H^{-1}(1 - R_{\mathsf{tag}}) > 0.$$

We will now discuss the numerical solution of the problem of maximizing

$$\eta = \frac{1}{2}\delta_{in}\left(H^{-1}(1 - R_{\mathsf{tag}}) - \frac{R_1}{1 - H(\delta_{in})}\right).$$

subject to $\delta_{in} < H^{-1}(1 - R_1)$.

Figure 6.3 shows a plot of η as a function of R_1 and R_{tag}. A reasonable choice of parameters for this instantiation is maybe $R_1 = 0.05$ and $R_{\mathsf{tag}} = 0.05$, for which $\eta \approx 0.02$ and $\rho/\rho' \approx 0.77$.

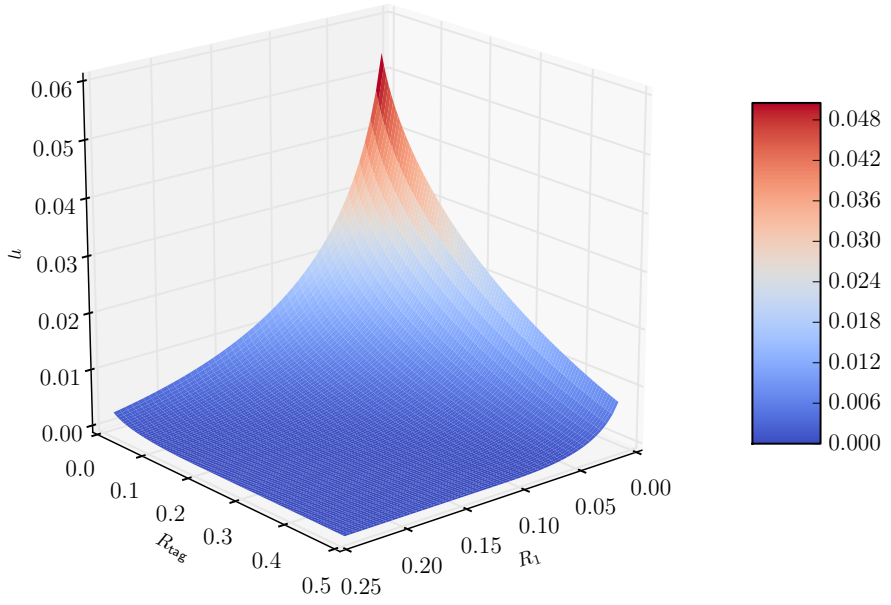

Figure 6.3.: Instantiation 2: The maximum toleratble noise rate η as a function of R_1 and R_{tag}

6.4.3. Instantiation 3: $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ with List Decoder, large Blocks

The third instantiation is very similar to the second, the only difference is that we use a list decoder instead of a unique decoder. Consequently, the constraints

$$\sigma = 1 - \delta_{\mathsf{tag}} = 1 - H^{-1}(1 - R_{\mathsf{tag}}),$$

$$R_{out} = \frac{R_1}{1 - H(\delta_{in})}$$

and

$$\delta_{in} < H^{-1}(1 - R_1)$$

remain the same. By Theorem 2.10 there exists a list decoder for $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ that can efficiently list decode an η fraction of bit errors and a σ fraction of block erasures given that

$$\eta \leq \frac{1}{2}(1 - \sigma) \cdot \left(1 - \sqrt{1 - 2\delta_{in}} - 2\sqrt{\frac{\delta_{in} R_{out}}{1 - \sigma}}\right).$$

Taking η to the maximum possible value and plugging in the expressions for σ and R_{out} yields

$$\eta = \frac{1}{2} H^{-1}(1 - R_{\mathsf{tag}}) \left(1 - \sqrt{1 - 2\delta_{in}} - 2\sqrt{\frac{\delta_{in} R_1}{H^{-1}(1 - R_{\mathsf{tag}}) \cdot (1 - H(\delta_{in}))}}\right).$$

A numerical maximization of η subject to $\delta_{in} < H^{-1}(1 - R_1)$ is given in the plot of Figure 6.4. It shows η as a function of R_1 and R_{tag}. The plot shows clearly that the *effect* of the list-decoder only kicks in for very small rates R_1. For higher rates, the performance of the decoder is worse than the unique decoder, used in instantiation 2. This however is not a shortcoming of the concept of list-decoding but rather of the specific decoder used here. It was also mentioned in 2.10 that this decoder only outperforms the unique decoder for very small rates.

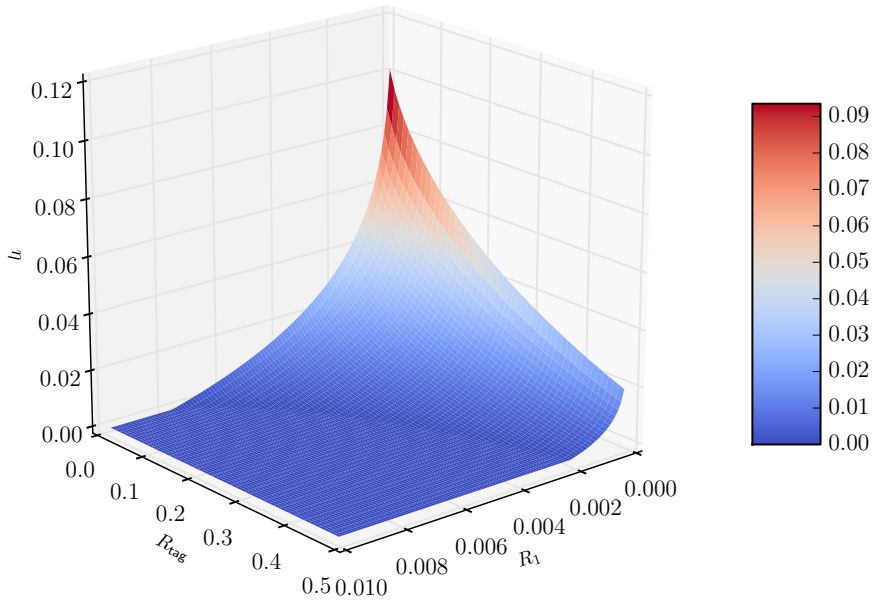

Figure 6.4.: Instantiation 3: The maximum tolerable noise rate η as a function of R_1 and R_{tag}

6.4.4. Instantiation 4: $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ with List Decoder, Blocks of size 1

We will now discuss a final instantiation. Again, we will choose C_1 as the concatenation of an outer Reed Solomon code and an inner code that meets the Gilbert Varshamov bound. Moreover, we will also use an alphabet size $q = 2$ for $\mathsf{C}_{\mathsf{tag}}$. However, this time we will choose blocks of size $k_2 = 1$, as in instantiation 1. As discussed above, this will make list decoding essential, as $q = 2$ implies that the fraction of bit erasures is greater than $1/2$. For this instantiation, the size of the tag-space is

$$|\mathfrak{T}_{\mathsf{LPN},\lambda}| = 2^{R_{\mathsf{tag}} k_1} = 2^{R_{\mathsf{tag}} m_1}.$$

Thus, let $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ be a concatenated code where RS is an outer Reed Solomon code with rate R_{out} and C_1 is a binary inner code with relative minimum distance δ_{in} and rate $R_{in} = 1 - H(\delta_{in})$. The constraints

$$\sigma = 1 - \delta_{\mathsf{tag}} = 1 - H^{-1}(1 - R_{\mathsf{tag}}),$$

$$R_{out} = \frac{R_1}{1 - H(\delta_{in})}$$

and

$$\delta_{in} < H^{-1}(1 - R_1)$$

are as before. By Theorem 2.11 there exists an efficient list decoder for $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$ that decodes an η fraction of bit errors and a σ fraction of bit erasures given that

$$\eta \leq \frac{1}{2} \cdot \left(1 - \sigma - \sqrt{(1+\epsilon)(1-2\delta_{in})} - \sqrt{\frac{(1-\sigma)R_{out}}{\epsilon \cdot (1-2\delta_{in})}} \right).$$

where $\epsilon > 0$ is arbitrary. Taking η to the maximum possible value and plugging in the expressions for σ and R_{out} yields

$$\eta = \frac{1}{2} \cdot \left(H^{-1}(1 - R_{\mathsf{tag}}) - \sqrt{(1+\epsilon)(1-2\delta_{in})} - \sqrt{\frac{H^{-1}(1 - R_{\mathsf{tag}})R_1}{\epsilon \cdot (1-2\delta_{in}) \cdot (1 - H(\delta_{in}))}} \right).$$

We will now directly consider the numerical maximization of η as a function of the free parameters δ_{in} and ϵ subject to $\epsilon > 0$ and $\delta_{in} < H^{-1}(1 - R_1)$. Figure 6.5 shows the maximum η as a function of R_1 and R_{tag}. As can be seen in Figure 6.5, a drawback of this decoder is that we only obtain reasonable values for η for unreasonably small rates $R_1 \approx 10^{-6}$.

6.4.5. IND-CCA2 scheme via CHK

We have seen that instantiation 1 requires large alphabet sizes q for the code $\mathsf{C}_{\mathsf{tag}}$, which leads to undesirably large public and private keys. In instantiations 3 and 4 we have examined the performance of list decoders for concatenated codes $\mathsf{C}_1 = \mathsf{C}_{in} \circ \mathsf{RS}$. It turns out that while this approach is interesting in theory, state of the art list decoders for concatenated codes only yield an advantage for very low rates R_1. This however, is also undesirable from a practical point of view. Instantiation 2 provides the most reasonable results from a practical point of view. Both the rate R_1 and the tolerable noise amount η can be set to realistic parameters. This comes at the price of a tag-space that is asymptotically smaller than that of instantiations 1 and 4. To conclude this Chapter, we will now apply the CHK transformation (Theorem 4.1) and obtain the following Theorem.

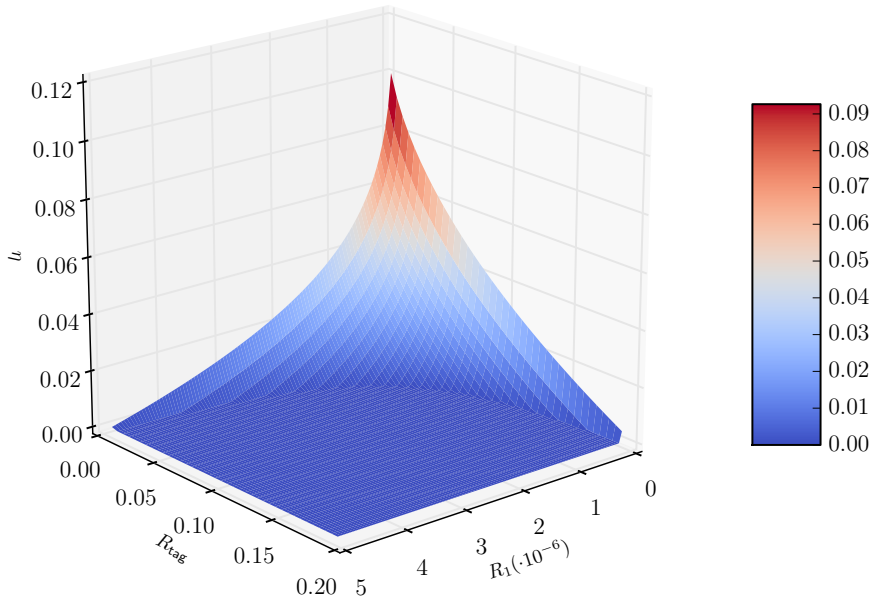

Figure 6.5.: Instantiation 4: The maximum toleratble noise rate η as a function of R_1 and R_{tag}

Theorem 6.3. *Let λ be a security parameter. Let $m, n = \Theta((\lambda/\log\lambda)^2)$. There exists an IND-CCA2 secure public key encryption scheme $\mathsf{PKE}_{\mathsf{LPN},CCA2}$ based on the hardness of the LPN problem $\mathsf{LPN}(n, m, \mathsf{Ber}(m, O(\frac{1}{\sqrt{m}})))$. The scheme has a constant factor ciphertext expansion, plaintexts of size $\Theta(m)$, and a key sizes of $\Theta(m^2)$.*

6.5. Further Developments

Very recently, Kiltz, Masny and Pietrzak [KMP14] proposed an alternative construction of an IND-CCA2 secure public key encryption scheme. Their scheme follows the same basic blue-print as ours by first constructing a tag-based encryption scheme and then applying the CHK transform (Theorem 4.1). The main difference between the scheme of [KMP14] and ours is that in their scheme the target-tag τ^* is only computationally hidden in the public key, while our scheme hides the target-tag statistically. The scheme of [KMP14] achieves slightly better performance than instantiation 2 of our scheme, while not depending on a specific choice of the code used to encode the secrets. Specifically, their public and secret keys are smaller by a factor of approximately $\frac{1}{2}$.

7. LWE with Uniform Errors

> Whoever wishes to become a philosopher must learn not to be frightened by absurdities
>
> Bertrand Russell [Rus36]

The following introduction and outline closely follow the introduction in the author's original publication of the results [DMQ13].

7.1. Introduction

As discussed in Chapter 3, the learning with errors (LWE) problem is a generalization of the learning parity with noise (LPN) problem to larger moduli and more general error distributions. The Learning-with-Errors (LWE) Problem asks to recover an unknown vector $\mathbf{s} \in \mathbb{Z}_q^n$, given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a *noisy codeword* $\mathbf{y} = \mathbf{As} + \mathbf{e}$, where $\mathbf{e} \in \mathbb{Z}_q^m$ is chosen from an error distribution χ^m.

This hardness assumption has had a significant impact in cryptography since its conception by Regev [Reg05] in 2005. Maybe the most intriguing feature of this problem, however, is its worst-to-average case connection [Reg05, Pei09]. This basically allows to transform an efficient adversary solving LWE on average, into an efficient (quantum) algorithm solving lattice problems in the worst case. Beyond this very strong hardness-guarantee, the problem has unmatched cryptographic versatility. It allows for IND-CPA and IND-CCA secure encryption [Reg05, GPV08, Pei09], lossy-trapdoor functions [PW08], (hierarchical) identity-based encryption [CHKP10, ABB10], fully homomorphic encryption [BV11, BGV12, Bra12] and many more. The worst-to-average-case reductions [Reg05, Pei09] crucially rely on gaussian error-distributions.

Gaussian distributions arise naturally in many different situations. This is mostly due to the central limit theorem, which roughly states that the sum of many independent (and bounded) random variables converges to a gaussian distribution. Consequently, the sum of several gaussian distributions is again gaussian.

However, this has the consequence that the cryptographic applications also need to use gaussian error-distributions. For the above-mentioned encryption-schemes, sampling from a gaussian error-distribution is usually the computationally heaviest

step (which occurs mostly during key-generation). It would thus be desirable to have a variant of the LWE problem enjoying the same worst-to-average-case connection, but that comes with an easier-to-sample error-distribution.

In this Chapter, we show that the the worst-case-connection of LWE with gaussian errors can be *transfered* to LWE with uniform errors from a small interval $[-r, r]$. A preliminary version of this result appeared in [DMQ13]. Our main-lever to obtain this result is a technique which we call *lossy codes*, reminiscent of the technique used by Peikert [Pei09] to establish the classical hardness of LWE (c.f. Section 3.7). Roughly speaking, lossy codes are pseudorandom codes that seem to be good codes. However, encoding messages with a lossy code and adding certain errors *provably* annihilates the message (on average). On the other hand, encoding the same message using a truly random code and adding the same type of error preserves the message, i.e. the message can be recovered *information theoretically* (yet not efficiently). Using a proof-strategy pioneered by Peikert and Waters [PW08], we conclude that recovering the message when encoding with a random code and adding noise must be computationally hard. Namely, if this was not the case, lossy codes could be efficiently distinguished from random codes, contradicting the pseudorandomness-property of lossy codes. The main-part of this Chapter is devoted to proving that a very simple construction of lossy codes for LWE *actually is lossy* for the uniform errors from $[-r, r]$. The key-insight for this construction is that the standard LWE problem with gaussian error-distribution allows us to implant many very short vectors into a random looking lattice. Our resulting worst-to-average case connection-factor for LWE with error-distribution $[-r, r]$ depends on the number of samples provided by LWE (while those for standard LWE [Reg05, Pei09] do not). We will therefore consider an m-bounded LWE problem $\mathsf{LWE}(n, m, q, [-r, r])$, where the number of samples m has a fixed $\mathsf{poly}(\lambda)$ upper bound (rather than being arbitrary $\mathsf{poly}(\lambda)$ depending on the adversary, like in the standard LWE problem). As lossy codes are basically an information-theoretical technique, this seems unavoidable. However, this drawback is still quite mild compared to the super-polynomial inapproximability assumptions made in other works [GKPV10, BV11, Bra12].

Applying the search-to-decision reduction of [MM11], we can conclude as a corollary that the decisional LWE problem $\mathsf{DLWE}(n, m, q, [-r, r])$ is also hard if $q = \mathsf{poly}(\lambda)$ is a prime modulus.

7.1.1. Outline

We will briefly outline the construction and the proof of our main results. As discussed in Chapter 3, the Learning With Errors Problem is basically the decoding problem for q-ary lattices: Given a randomly chosen generator-matrix $\mathbf{A}$ and a vector $\mathbf{y}$, find the nearest lattice point (or codeword) $\mathbf{As}$, under the promise that $\mathbf{y}$ was generated by drawing a random point from the lattice and adding an error by some specified distribution. We want to show that this decoding-problem is hard if the error is component-wise chosen by uniformly from $[-r, r]$. Assume that we knew that there exists a distribution of *lossy* matrices $\mathbf{A}'$ such that that the decoding-problem has no unique solution if the errors are drawn uniformly from $[-r, r]$, i.e., adding errors to a lattice-point $\mathbf{A}'\mathbf{s}$ loses information about $\mathbf{s}$. If distinguishing such matrices from truly random matrices is hard, we can conclude that the decoding-problem must be hard for truly random matrices. Otherwise, given a decoder for random matrices we can distinguish random matrices from lossy matrices. The distinguisher samples random challenges for the decoder. If the decoder succeeds

significantly often, i.e. if it outputs the same $\mathbf{s}$ that was used to sample the instance, then the given matrix must come from the random distribution, as this behavior is impossible for the lossy distribution. Thus, our task is to construct a distribution of lossy codes for uniform errors from $[-r, r]$. Our starting-point to find such a distribution is the observation that the standard LWE-problem allows us to construct pseudorandom matrices that generate lattices which contain many vectors that are significantly shorter than one would expect for lattices generated by truly random matrices. Let $\mathbf{G} \in \mathbb{Z}_q^{m \times n}$ be component-wise chosen according to a (short) discrete gaussian distribution $D_{\alpha q}$. We want to set the parameters α and r such that the lattice generated by $\mathbf{G}$ is *bad* on average against errors from $[-r, r]$. Put differently, if $\mathbf{y} = \mathbf{Gs}+\mathbf{e}$, where $\mathbf{s}$ is chosen uniformly at random and $\mathbf{e}$ is chosen uniformly from $[-r, r]^m$, we want that, with overwhelming probability, there exist at least one more "admissible" $\mathbf{s}' \neq \mathbf{s}$ and $\mathbf{e}' \in [-r, r]^m$ such that $\mathbf{y} = \mathbf{Gs}' + \mathbf{c}'$. As $\mathbf{e}$ is distributed uniformly on the volume $[-r, r]^m$, each $\mathbf{s}'$ will have the same posterior-probability given $\mathbf{G}$ and $\mathbf{y}$. If there is at least one such $\mathbf{s}'$, then $\mathbf{y}$ statistically hides at least one bit of $\mathbf{s}$ and we can implement the distinguisher sketched above. To make this lossy code pseudorandom, we *hide* the matrix $\mathbf{G}$ in a bigger matrix $\mathbf{A}$. This can be achieved in a pretty standard way. Let $\mathbf{A'} \in \mathbb{Z}_q^{m \times n}$ be chosen uniformly at random. Define $\mathbf{B} = (\mathbf{A'} \| \mathbf{G})$ as the concatenation of $\mathbf{A'}$ and $\mathbf{G}$. $\mathbf{B}$ now contains the $\mathbf{G}$ as a sub-matrix. Thus, $\mathbf{B}$ has a *lossy sub-code*. As having a lossy sub-code is sufficient to be lossy, $\mathbf{A}$ is also lossy. We can randomize the generator-matrix $\mathbf{B} = (\mathbf{A'} \| \mathbf{G})$ by applying the transformation

$$\mathbf{T} = \begin{pmatrix} \mathbf{I} & \mathbf{T'} \\ \mathbf{0} & \mathbf{I} \end{pmatrix},$$

for a $\mathbf{T'} \in \mathbb{Z}_q^{n \times n}$ chosen uniformly at random. This yields the randomized generator $\mathbf{A} = \mathbf{BT} = (\mathbf{A'} \| \mathbf{A'T'} + \mathbf{G})$ for the same code. By the decisional LWE-assumption (for specific parameters), the matrix $\mathbf{A}$ is pseudorandom.

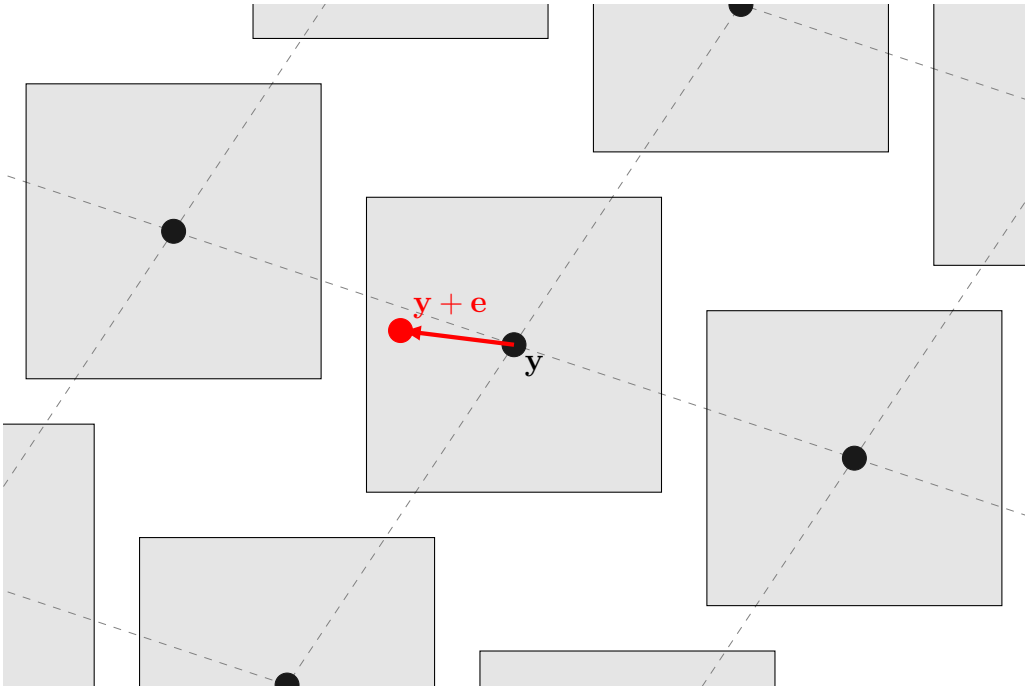

Figure 7.1.: The non-lossy case: The cubes around lattice points are non-intersecting

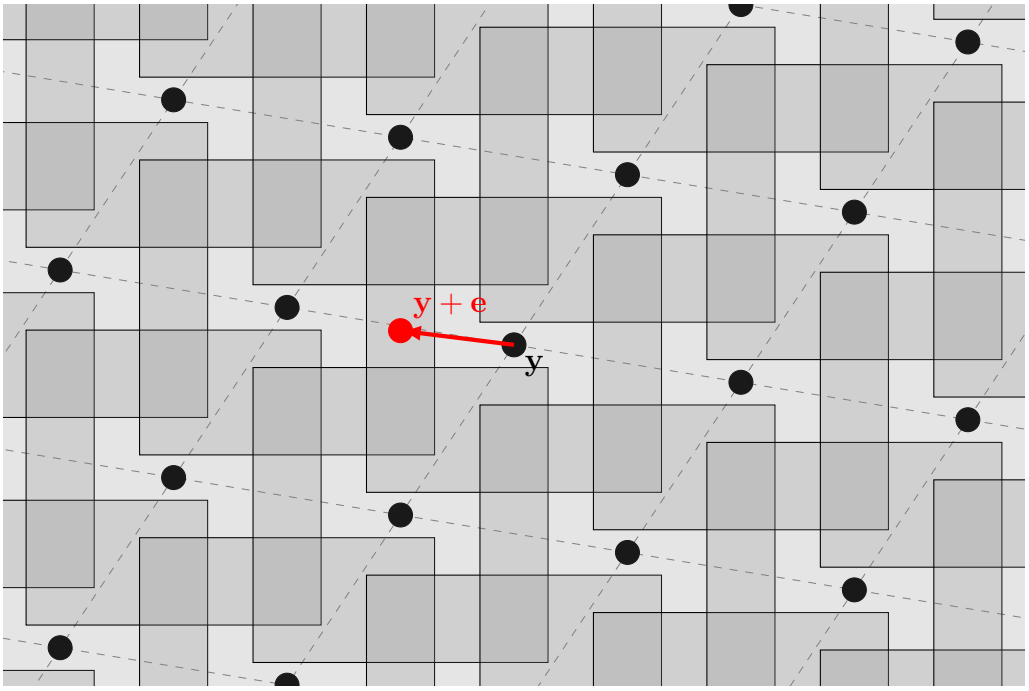

Figure 7.2.: The lossy case: The cubes around lattice points are highly intersecting

7.2. LWE with Non-Gaussian Errors for Superpolynomial Hardness

In this section we will discuss prior results on the hardness of LWE with non-gaussian errors. Applebaum, Ishai and Kushilevitz [AIK11] noted that if one is willing to assume that standard LWE is hard, even if the noise rate α is below $\frac{1}{\mathsf{poly}(\lambda)}$, then we can replace the gaussian error distribution by other error distributions. Specifically, let $D_{\alpha q}$ be the discrete gaussian distribution with noise rate α and let χ be any efficiently samplable distribution on $\mathbb{Z}$. Let $\chi' = \chi + D_{\alpha q}$ be the distribution obtained by adding independent samples of χ and $D_{\alpha q}$, i.e. $x \leftarrow_\$ \chi'$ is sampled by picking $z \leftarrow_\$ \chi$ and $e \leftarrow_\$ D_{\alpha q}$ and setting $x \leftarrow z + e$. χ' can be seen as a *smoothed* version of χ. Assume now that the distributions χ and χ' are statistically close. Then we claim that $\mathsf{DLWE}(n, q, \chi)$ is as hard as $\mathsf{DLWE}(n, q, D_{\alpha q})$. The reason for this is that samples from $\mathsf{DLWE}(n, q, D_{\alpha q})$ can be efficiently converted to samples from $\mathsf{DLWE}(n, q, \chi')$. Assume that $(\mathbf{a}, y)$ is a sample provided by $\mathsf{DLWE}(n, q, D_{\alpha q})$. We let $z \leftarrow_\$ \chi$. We claim that $(\mathbf{a}, y + z)$ is then a sample from $\mathsf{DLWE}(n, q, \chi')$. If $y = \langle \mathbf{a}, \mathbf{s} \rangle + e$, then

$$y + z = \langle \mathbf{a}, \mathbf{s} \rangle + e + z = \langle \mathbf{a}, \mathbf{s} \rangle + x$$

where x is a sample of χ'. On the other hand, if y is uniformly random, then $y + z$ is also uniformly random. Thus, the converted samples have the right distribution. However, since χ and χ' are statistically close, any distinguisher distinguishing $\mathsf{DLWE}(n, q, \chi)$ can also be used to distinguish $\mathsf{DLWE}(n, q, \chi')$. By the above conversion, such a distinguisher can be used to distinguish $\mathsf{DLWE}(n, q, D_{\alpha q})$, contradicting its hardness.

Thus, χ and χ' being statistically close is a sufficient condition for $\mathsf{DLWE}(n, q, \chi)$ to be hard. We will briefly consider the case that χ is the uniform distribution on an interval $[-r, r]$. We claim that χ and $\chi' = \chi + D_{\alpha q}$ are statistically close if and only if $\frac{\alpha q}{r} \leq \mathsf{negl}(\lambda)$. Given that $\frac{\alpha q}{r}$ is negligible, a sample $z \leftarrow_\$ [-r, r]$ is with overwhelming probability in the interval $[-r + 2\alpha q\lambda, r - 2\alpha q\lambda]$, as

$$\Pr_z[z \notin [-r + 2\alpha q\lambda, r - 2\alpha q\lambda]] = \frac{4\alpha q\lambda}{r} \leq \mathsf{negl}(\lambda).$$

Let $e \leftarrow_\$ D_{\alpha q}$. As e is $\alpha q\lambda$ bounded, it holds that

$$\begin{aligned}\Pr_e[z + e \in [-r + \alpha q\lambda, r - \alpha q\lambda]] &\leq \Pr_z[z \notin [-r + 2\alpha q\lambda, r - 2\alpha q\lambda]] + \Pr[|e| > \alpha q\lambda] \\ &\leq 1 - \mathsf{negl}(\lambda).\end{aligned}$$

Thus, both z and $z+e$ are in the interval $[-r+2\alpha q\lambda, r-2\alpha q\lambda]$, except with negligible probability. For $t \in [-r + 2\alpha q\lambda, r - 2\alpha q\lambda]$ it holds that

$$\begin{aligned}\Pr[z + e = t] &= \sum_{s \in [-r,r]} \Pr[s + e = t] \underbrace{\Pr[z = s]}_{=\Pr[z=t]} \\ &= \Pr[z = t] \cdot \underbrace{\Pr[t - e \in [-r, r]]}_{\geq 1 - \mathsf{negl}(\lambda)} \\ &\geq \Pr[z = t] \cdot (1 - \mathsf{negl}(\lambda)),\end{aligned}$$

from which follows that z and $z + e$ are statistically close.

If, on the other hand, $\frac{\alpha q}{r} > \frac{1}{\mathsf{poly}(\lambda)}$, then

$$\Pr[z \notin [-r + \alpha q, r - \alpha]] = \frac{2\alpha q}{r} \geq \frac{1}{\mathsf{poly}(\lambda)}.$$

Moreover, $\Pr[|e| > 2\alpha q] \geq \frac{1}{\mathsf{poly}(\lambda)}$ and thus

$$\Pr[z + e \notin [-r, r]] \geq \frac{1}{\mathsf{poly}(\lambda)}$$

which yields that z and $z + e$ are not statistically close.

7.3. Lossy Codes

In this section, we introduce the main technical tool of this chapter, which we call lossy codes. We will show that the existence of lossy codes implies that the associated decoding problems for random codes are hard.

Definition 7.1 (Families of Lossy Codes)**.** *Let λ be a security parameter, let $q = q(\lambda)$ be a modulus, let $m, n = \mathsf{poly}(\lambda)$ and $\gamma = \gamma(\lambda)$. Let $\mathcal{C}$ be a distribution on $\mathbb{Z}_q^{m \times n}$ and let $\boldsymbol{\chi}$ be a distribution on $\mathbb{Z}_q^m$. Let $\mathbf{L} \leftarrow_\$ \mathcal{C}$ and let $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m \times n}$ be chosen uniformly at random. We say that $\mathcal{C}$ is γ-lossy for the error-distribution $\boldsymbol{\chi}$, if the following 3 properties hold.*

1. **$\mathbf{L}$ *is pseudorandom:*** *It holds that $\mathbf{L} \approx_c \mathbf{U}$.*

2. **$\mathbf{L}$ *is lossy:*** *Let $\mathbf{y} = \mathbf{L} \cdot \tilde{\mathbf{s}} + \tilde{\mathbf{e}}$ (where $\tilde{\mathbf{s}}$ is chosen uniformly from $\mathbb{Z}_q^n$ and $\tilde{\mathbf{e}}$ is distributed according to $\boldsymbol{\chi}$), let $\mathbf{s}$ be chosen uniformly from $\mathbb{Z}_q^n$ and let $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$. Then it holds that $\Pr_{(\mathbf{L},\mathbf{y})}[H_\infty(\mathbf{s}|\mathbf{L}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq \gamma] \geq 1 - \mathsf{negl}(\lambda)$.*

3. **$\mathbf{U}$ *is non-lossy:*** *Let $\mathbf{y} = \mathbf{U} \cdot \tilde{\mathbf{s}} + \tilde{\mathbf{e}}$ (where $\tilde{\mathbf{s}}$ is chosen uniformly from $\mathbb{Z}_q^n$ and $\tilde{\mathbf{e}}$ is distributed according to $\boldsymbol{\chi}$), let $\mathbf{s}$ be chosen uniformly from $\mathbb{Z}_q^n$ and let $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$. Then it holds that $\Pr_{(\mathbf{U},\mathbf{y})}[H_\infty(\mathbf{s}|\mathbf{U}\mathbf{s} + \mathbf{e} = \mathbf{y}) = 0] \geq 1 - \mathsf{negl}(\lambda)$.*

Our main motivation for defining lossy codes is proving that the decoding-problem of recovering $\mathbf{s}$ given a matrix $\mathbf{A}$ and a noisy codeword $\mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{A}$ and $\mathbf{s}$ are chosen uniformly and $\mathbf{e}$ is chosen from $\boldsymbol{\chi}$, is computationally hard, even though $\mathbf{s}$ is information-theoretically (with overwhelming probability) uniquely defined.

Theorem 7.1. *Let λ be a security-parameter, let $m, n = \mathsf{poly}(\lambda)$ and let $q = q(\lambda)$ be a modulus. Let $\boldsymbol{\chi}$ be an efficiently samplable distribution on $\mathbb{Z}_q^m$.*

1. *Let $\boldsymbol{\chi}$ be a uniform distribution with efficiently decidable support. Then the problem $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ is hard, given that there exists a distribution of 1-lossy codes $\mathcal{C}$ on $\mathbb{Z}_q^{m \times n}$ for the error-distribution $\boldsymbol{\chi}$.*

2. *Let $\gamma = \gamma(n) = \omega(\log(n))$. Then $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ is hard, given that there exists a distribution of γ-lossy codes $\mathcal{C}$ on $\mathbb{Z}_q^{m \times n}$ for the error-distribution $\boldsymbol{\chi}$.*

Proof. First notice that due to the non-lossiness property of uniformly distributed $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ instances of $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ have a unique solution, except with negligible probability. For contradiction, let $\mathcal{A}$ be a PPT-adversary the solves the problem $\mathsf{LWE}(n, m, q, \boldsymbol{\chi})$ with non-negligible probability ϵ.

We will begin with the first statement of the theorem. Let $\mathbf{L} \leftarrow_\$ \mathcal{C}$ and let $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ be chosen uniformly at random. Using $\mathcal{A}$, we will construct a PPT-distinguisher $\mathcal{D}$ that distinguishes $\mathbf{L}$ and $\mathbf{U}$ with non-negligible advantage. Say that a solution $\mathbf{s}$ for an instance $(\mathbf{A}, \mathbf{y})$ is valid, if $\mathbf{y} - \mathbf{A} \cdot \mathbf{s}$ is in the support of the error-distribution χ.

There are two different behaviors that algorithm $\mathcal{A}$ could exhibit when receiving inputs of the form $(\mathbf{L}, \mathbf{y})$, where $\mathbf{L}$ is chosen from $\mathcal{C}$ and $\mathbf{y} = \mathbf{L}\mathbf{s} + \mathbf{e}$.

1. The probability that $\mathcal{A}$ outputs a valid solution $\mathbf{s}$ is negligible.
2. There exists a non-negligible $\epsilon'(\lambda)$ such that the probability that $\mathcal{A}$ outputs a valid solution $\mathbf{s}$ with probability at least ϵ'.

In the first case we can construct the distinguisher $\mathcal{D}_1$ as follows.

Distinguisher $\mathcal{D}_1$
Input $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$
$\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$
$\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$
$\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
$\mathbf{s}' \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{y})$
If $\mathbf{s}' = \mathbf{s}$
    Return `non-lossy`
Otherwise
    Return `lossy`

$\mathcal{D}_1$ basically generates an LWE sample and checks if $\mathcal{A}$ is successful in recovering the solution. Clearly, if $\mathbf{A}$ is chosen uniformly from $\mathbb{Z}_q^{m\times n}$, then $\mathcal{A}$ recovers $\mathbf{s}$ with probability at least ϵ. On the other hand, if $\mathbf{A}$ is chosen according from the lossy distribution $\mathcal{C}$, then $\mathcal{A}$ recovers $\mathbf{s}$ only with negligible probability. Thus it holds that

$$\mathsf{Adv}(\mathcal{D}_1) = |\Pr[\mathcal{D}_1(\mathbf{U}) = \texttt{lossy}] - \Pr[\mathcal{D}_1(\mathbf{L}) = \texttt{lossy}]| = \epsilon(\lambda) - \mathsf{negl}(\lambda),$$

which is non-negligible.

In the second case, we construct the distinguisher $\mathcal{D}_2$ differently.

First, observe that such a *collision* $\mathbf{s}' \neq \mathbf{s}$ cannot exist (except with negligible probability) if $\mathbf{A}$ is chosen according to the uniform distribution on $\mathbb{Z}_q^{m\times n}$. This is due to the non-lossiness property of $\mathbb{Z}_q^{m\times n}$. On the other hand, consider that $\mathbf{A}$ is chosen from the lossy distribution $\mathcal{C}$. Then it holds (with overwhelming probability) that $H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq 1$. Thus it holds (even for an unbounded $\mathcal{A}$) that $\mathcal{A}$ outputs the same $\mathbf{s}$ that was chosen by $\mathcal{D}$ with probability at most $1/2$, conditioned that $\mathcal{A}$ outputs a valid $\mathbf{s}$. Thus, conditioned that $\mathcal{A}$ gives a valid output, there is a chance of $1/2$ that $\mathcal{A}$ outputs a valid $\mathbf{s}' \neq \mathbf{s}$. As $\mathcal{A}$ gives a valid output with probability at least ϵ', $\mathcal{A}$ outputs a collision $\mathbf{s}'$ with probability at least $\epsilon'/2$. Thus $\mathcal{D}_2$ distinguishes $\mathbf{U}$ from $\mathbf{L}$ with advantage at least $\epsilon'/2$, which is non-negligible.

Distinguisher $\mathcal{D}_2$
Input $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
$\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$
$\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$
$\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
$\mathbf{s}' \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{y})$
$\mathbf{e}' \leftarrow \mathbf{y} - \mathbf{A}\mathbf{s}'$
If $\mathbf{s}' \neq \mathbf{s}$ and $\mathbf{e}' \in \mathsf{Support}(\boldsymbol{\chi})$
Return `lossy`
Otherwise
Return `non-lossy`

We now turn to the second statement of the theorem. In this case the construction of the distinguisher $\mathcal{D}$ is straightforward.

Distinguisher $\mathcal{D}$
Input $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
$\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$
$\mathbf{e} \leftarrow_\$ \boldsymbol{\chi}$
$\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}$
$\mathbf{s}' \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{y})$
If $\mathbf{s}' = \mathbf{s}$
Return `non-lossy`
Otherwise
Return `lossy`

Again, if $\mathbf{A}$ was chosen uniformly at random from $\mathbb{Z}_q^{m \times n}$, then $\mathcal{A}$ outputs $\mathbf{s}$ (which is in this case unique) with probability at least ϵ. On the other hand, if $\mathbf{A}$ comes from the lossy distribution $\mathcal{C}$, then $\mathcal{A}$ finds $\mathbf{s}$ with probability at most $2^{-H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s}+\mathbf{e}=\mathbf{y})} \leq 2^{-\gamma(n)}$ (this holds with overwhelming probability in the choice of $\mathbf{A}$ and $\mathbf{y}$), which is negligible (as $\gamma(n) = \omega(\log(n))$). All together, $\mathcal{D}$ distinguishes $\mathbf{U}$ from $\mathbf{L}$ with advantage at least $\epsilon - 2^{-\gamma}$, which is non-negligible. □

7.4. Construction of Lossy Codes for Uniform Errors from Standard-LWE

We will now provide the details of the construction outlined in Section 7.1.1.

Construction 7.1. *Let λ be a security parameter, let $q = q(\lambda)$ be a modulus, $m, n, k = \mathsf{poly}(\lambda)$ with $k \leq n$. Let $\alpha > 0$ and let $D_{\alpha q}$ be a discrete gaussian distribution on $\mathbb{Z}$. The distribution $\mathcal{C}_{\alpha,k}$ defined on $\mathbb{Z}_q^{m \times n}$ is specified by the following sampling procedure.*

We will now show, that for certain parameter choices, the distribution defined in Construction 7.1 is lossy for the errors chosen uniformly from $[-r, r]^m$. The

$\mathsf{SampleLossy}(1^\lambda)$
$\mathbf{A}' \leftarrow_\$ \mathbb{Z}_q^{m\times k}$
$\mathbf{T}' \leftarrow_\$ \mathbb{Z}_q^{k\times(n-k)}$
$\mathbf{G} \leftarrow_\$ D_{\alpha q}^{m\times(n-k)}$
$\mathbf{A} \leftarrow (\mathbf{A}'\|\mathbf{A}'\mathbf{T}' + \mathbf{G})$
Return $\mathbf{A}$

pseudorandomness of the distribution $\mathcal{C}_{\alpha,k}$ can be established directly assuming the hardness of $\mathsf{DMLWE}(k, m, n-k, q, D_{\alpha q})$, which in turn by Lemma 3.4 follows from the hardness of $\mathsf{DLWE}(k, m, n-k, q, D_{\alpha q})$.

Lemma 7.1. *Let λ be a security-parameter, let $q = q(\lambda)$ be a modulus, let $m, n, k = \mathsf{poly}(\lambda)$ with $k \leq n$ and let $\alpha = \alpha(\lambda) \in (0, 1)$. Assuming that $\mathsf{DLWE}(k, m, n-k, q, D_{\alpha q})$ is hard, the distribution $\mathcal{C}_{\alpha,k}$ is pseudorandom.*

The non-lossiness of truly random $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ can be established by the following Gilbert-Varshamov-type argument.

Lemma 7.2. *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$, $r = r(\lambda)$ and $q = q(\lambda)$ be a positive integers. Let $\epsilon > 0$ be an arbitrarily small constant. Assume $n \leq (1 - \log_q(4r+1) - \epsilon)m$ and let $\mathbf{A}$ be chosen uniformly at random from $\mathbb{Z}_q^{m\times n}$. Then the shortest vector of the lattice $\Lambda_q(\mathbf{A})$ has length (in the $\|\cdot\|_\infty$-norm) greater than $2r$, except with negligible probability.*

Proof. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be chosen uniformly at random. Fix a vector $\mathbf{s} \neq 0 \in \mathbb{Z}_q^n$. Then the vector $\mathbf{A}\cdot\mathbf{s}$ is distributed uniformly at random in $\mathbb{Z}_q^m$. Thus it holds that $\Pr_{\mathbf{A}}[\|\mathbf{A}\cdot\mathbf{s}\|_\infty \leq 2r] \leq \left(\frac{4r+1}{q}\right)^m$. Thus, a union-bound yields that

$$\Pr[\exists x \neq 0 \in \mathbb{Z}_q^n : \|\mathbf{A}\mathbf{s}\|_\infty \leq 2r] \leq \frac{(4r+1)^m}{q^{m-n}} = q^{n-(1-\log_q(4r+1))m} \leq q^{-\epsilon m},$$

as $n \leq (1 - \log_q(4r+1) - \epsilon)m$. This immediately yields

$$\Pr[\forall x \neq 0 \in \mathbb{Z}_q^n : \|\mathbf{A}\mathbf{s}\|_\infty \geq 2r] \geq 1 - q^{-\epsilon m},$$

which is overwhelming. □

Lemma 7.2 immediately yields that if $n \leq (1 - \log_q(4r+1) - \epsilon)m$, then uniformly chosen $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ are non-lossy for errors chosen uniformly from $[-r, r]^m$.

Corollary 7.3. *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$, $r = r(\lambda)$ and $q = q(\lambda)$ be a positive integers. Let $\epsilon > 0$ be an arbitrarily small constant. Assume $n \leq (1 - \log_q(4r+1) - \epsilon)m$. Let $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{m\times n}$ be chosen uniformly at random, let $\mathbf{y} = \mathbf{U}\cdot\tilde{\mathbf{s}} + \tilde{\mathbf{e}}$ (where $\tilde{\mathbf{s}}$ is chosen uniformly from $\mathbb{Z}_q^n$ and $\tilde{\mathbf{e}}$ is distributed according to $\boldsymbol{\chi}$), let $\mathbf{s}$ be chosen uniformly from $\mathbb{Z}_q^n$ and let $\mathbf{e}$ be chosen according to $\boldsymbol{\chi}$. Then it holds that*

$$\Pr_{(\mathbf{U},\mathbf{y})}[H_\infty(\mathbf{s}|\mathbf{U}\mathbf{s} + \mathbf{e} = \mathbf{y}) = 0] \geq 1 - \mathsf{negl}(\lambda),$$

i.e. $\mathbf{U}$ is non-lossy.

Proof. By Lemma 7.2, it holds for the shortest nonzero vector $\mathbf{v}$ of $\Lambda_q(\mathbf{U})$ that $\|\mathbf{v}\|_\infty > 2r$, except with negligible probability. Thus, $\mathbf{U}$ and $\mathbf{y} = \mathbf{U}\tilde{\mathbf{s}} + \tilde{\mathbf{e}}$ uniquely determine $\tilde{\mathbf{s}}$, as $\|\tilde{\mathbf{e}}\| \leq r$. Consequently, $H_\infty(\mathbf{s}|\mathbf{U}\mathbf{s} + \mathbf{e} = \mathbf{y}) = 0$. □

Definition 7.2. *We say that a vector $\mathbf{y} \in \mathbb{Z}_q^m$ is N-*ambiguous *for a matrix $\mathbf{A}$ and a distance r, if $|\{\mathbf{s} \in \mathbb{Z}_q^n | \|\mathbf{y} - \mathbf{A} \cdot \mathbf{s}\|_\infty \leq r\}| \geq N$. If $\mathbf{A}$ and r are clear by context, we just say that $\mathbf{y}$ is N-ambiguous.*

Notice that if $\mathbf{y}$ is N-ambiguous, then for every $\mathbf{z} \in \mathbb{Z}_q^n$ by linearity it holds that $\mathbf{y} + \mathbf{A}\mathbf{z}$ is also N-ambiguous.

Since we want to establish lossiness for errors uniformly distributed in $[-r, r]$, counting the number of possible preimages is sufficient, as each preimage is equally likely. This is formalized in the following lemma.

Lemma 7.4. *Let λ be a security parameter. Let $m, n = \mathsf{poly}(\lambda)$, let $q = q(\lambda)$ be a modulus and let $r = r(\lambda)$ and $N = N(\lambda)$ be positive integers. Fix a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$. Assume that $\mathbf{y} \in \mathbb{Z}_q^m$ is N-ambiguous for the matrix $\mathbf{A}$ and distance r. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be chosen uniformly at random and $\mathbf{e} \leftarrow_\$ [-r, r]^m$ be chosen uniformly from $[-r, r]^m$. Then it holds that $H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq \log(N)$.*

Proof. Since $\mathbf{s}$ and $\mathbf{e}$ are drawn from uniform distributions, $p := \Pr[\mathbf{s} = \tilde{\mathbf{s}}, \mathbf{e} = \tilde{\mathbf{e}}]$ is the same for all $\tilde{\mathbf{s}} \in \mathbb{Z}_q^n$ and $\tilde{\mathbf{e}} \in [-r, r]^m$. Let $X := \{\mathbf{z} \in \mathbb{Z}_q^n | \|\mathbf{y} - \mathbf{A}\mathbf{z}\| \leq r\}$. As $\mathbf{y}$ is N-ambiguous it holds that $|X| \geq N$, thus

$$\Pr[\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}] = \sum_{\mathbf{z} \in \mathbb{Z}_q^n} \Pr[\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}, \mathbf{s} = \mathbf{z}] = \sum_{\mathbf{z} \in X} \Pr[\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{z}, \mathbf{s} = \mathbf{z}] \geq p \cdot N.$$

Thus it holds for all $\mathbf{z} \in \mathbb{Z}_q^n$ that

$$\Pr[\mathbf{s} = \mathbf{z} | \mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}] = \frac{\Pr[\mathbf{s} = \mathbf{z}, \mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}]}{\Pr[\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}]} \leq \frac{1}{N}.$$

This immediately implies $H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq \log(N)$, which concludes the proof. □

Definition 7.3. *Let λ be a security parameter. Let $B = B(\lambda) > 0$. We say that a distribution χ is B-bounded, if*

$$\Pr[|x| > B] \leq \mathsf{negl}(\lambda),$$

where x is chosen according to χ.

The following lemma shows that if we sample $\mathbf{e}$ uniformly from $[-r, r]^m$, then with overwhelming probability $\mathbf{e}$ is such that if we add a sample $\mathbf{g}$ from an appropriately bounded distribution χ^m, then, with substantial probability over the choice of $\mathbf{g}$, $\mathbf{e} - \mathbf{g}$ is also in $[-r, r]^m$.

Lemma 7.5. *Let $n, m, B > 0$ be integers, let $r > (m+1)B$ and let $\epsilon < 1/2$. Let χ be a B-bounded symmetrical distribution on $\mathbb{Z}$. Let $\mathbf{e}$ be chosen uniformly at random from $[-r, r]^m$ and let $\mathbf{g}$ be distributed according to χ^m. Then it holds that*

$$\Pr_{\mathbf{e}}\left[\Pr_{\mathbf{g}}[\|\mathbf{e} - \mathbf{g}\|_\infty \leq r] \geq \epsilon\right] \geq 1 - m \cdot \epsilon^{\log(r/(m \cdot B))} - \mathsf{negl}(\lambda).$$

Proof. We will first bound the probability that it holds for more than $k = \lfloor -\log(\epsilon) \rfloor$ components e_i of $\mathbf{e}$ that $|e_i| > r - B$, i.e. that e_i is not in the interval $[-r+B, r-B]$. For $i = 1, \ldots, m$ let Z_i be a random-variable that is 1 if $|e_i| > r - B$ and 0 otherwise. As $e_1, \ldots, e_m$ are iid., $Z_1, \ldots, Z_m$ are also iid. Thus let

$$p = \Pr[Z_1 = 1] = \cdots = \Pr[Z_m = 1].$$

As $e_1 \leftarrow_\$ [-r, r]$ and $p = \Pr[Z_1 = 1] = \Pr[|e_1| > r - B]$, it holds that

$$(B-1)/r \leq p \leq B/r.$$

Set $Z = \sum_{i=1}^{m} Z_i$. Clearly, Z is the number of components of $\mathbf{e}$ that are not in the interval $[-r + B, r - B]$ and it is binomially distributed. We can bound the probability $\Pr[Z > k]$ by the following low deviation inequality.

$$\begin{aligned}\Pr[Z > k] = \sum_{i=k+1}^{m} \binom{m}{i} p^i (1-p)^{m-i} &\overset{(1)}{\leq} m \underbrace{\binom{m}{k+1}}_{\leq m^{k+1}} \underbrace{p^{k+1}}_{\leq (B/r)^{k+1}} \underbrace{(1-p)^{m-k-1}}_{\leq 1} \\ &\leq m \cdot \left(\frac{m \cdot B}{r}\right)^{k+1} \overset{(2)}{<} m \cdot \left(\frac{m \cdot B}{r}\right)^{-\log(\epsilon)} = m \cdot \epsilon^{\log(r/(m \cdot B))}.\end{aligned}$$

Inequality (1) holds, as $\binom{m}{i} p^i (1-p)^{m-i}$ is monotonically decreasing for $i \geq \lfloor (m+1)p \rfloor \geq \lfloor (m+1)(B-1)/r \rfloor = 0$. Inequality (2) holds as $m \cdot B/r < 1$ and $k + 1 > -\log(\epsilon)$. Now, fix an $\mathbf{e}$ and assume that it holds that it holds for at most k components $e_{i_1}, \ldots, e_{i_k}$ of $\mathbf{e}$ that $|e_{i_j}| > r - B$. Let $i \in \{i_1, \ldots, i_k\}$.

Let $\tilde{\chi}$ be the restriction of the distribution χ to $[-B, B]$, i.e. $\tilde{\chi}$ can be sampled by sampling χ and rejecting samples of outside of $[-B, B]$. As χ is symmetric and $[-B, B]$ is symmetric, $\tilde{\chi}$ is also symmetric. Moreover, as χ is B-bounded, χ and $\tilde{\chi}$ are statistically close. Therefore, $\mathbf{g} \leftarrow_\$ \chi^m$ is statistically close to a $\mathbf{g}' \leftarrow_\$ \tilde{\chi}^m$. Thus, let $\mathbf{g}' \leftarrow_\$ \tilde{\chi}^m$.

If $\mathsf{sgn}(g_i') = \mathsf{sgn}(e_i)$, then it holds that $|e_i - g_i'| = |e_i| - |g_i'| \leq |e_i| \leq r$. As $\tilde{\chi}$ is a symmetrical distribution, it holds that $\Pr[\mathsf{sgn}(g_i') = \mathsf{sgn}(e_i)] \geq \frac{1}{2}$. Therefore, it holds that $\Pr[|e_i - g_i'| \leq r] \geq \frac{1}{2}$. For all other indices $j \notin \{i_1, \ldots, i_k\}$ it holds that $|e_j| \leq r - B$. The triangle-inequality yields $|e_j - g_j'| \leq |e_j| + |g_j'| \leq r - B + B = r$. Therefore, we have that $\Pr[|e_j - g_j'| \leq r] = 1$. Putting this together, we get that

$$\Pr[\|\mathbf{e} - \mathbf{g}'\|_\infty \leq r] = \prod_{i=1}^{m} \Pr[|e_i - g_i'| \leq r] \geq 2^{-k} \geq \epsilon.$$

All together, it holds that

$$\begin{aligned}\Pr_{\mathbf{e}}[\Pr_{\mathbf{g}}[\|\mathbf{e} - \mathbf{g}\|_\infty \leq r] \geq \epsilon] &\geq \Pr_{\mathbf{e}}[\Pr_{\mathbf{g}'}[\|\mathbf{e} - \mathbf{g}'\|_\infty \leq r] \geq \epsilon] - \mathsf{negl}(\lambda) \\ &\geq 1 - m \cdot \epsilon^{\log(r/(m \cdot B))} - \mathsf{negl}(\lambda),\end{aligned}$$

which concludes the proof. □

We can now show that Construction 7.1 also fulfills the lossiness-condition for appropriate parameters.

Lemma 7.6. *Let λ be a security-parameter, let $m, n, k = \mathsf{poly}(\lambda)$ with $n - k \leq \lambda$ and let $q = q(\lambda)$ be a modulus. Let $B = B(\lambda)$ and let χ be a symmetric B-bounded distribution on $\mathbb{Z}$. Let $r \geq m \cdot B \cdot \kappa$ for some $\kappa = \kappa(\lambda) > 0$ with $\kappa = 2^{\sqrt{\omega(\log(\lambda))}}$. Let $\mathbf{G}$ be chosen according to $\chi^{m\times(n-k)}$, let the matrix $\mathbf{A}' \leftarrow_\$ \mathbb{Z}_q^{m\times k}$ be chosen uniformly from $\mathbb{Z}_q^{m\times k}$, let $\mathbf{T}' \leftarrow_\$ \mathbb{Z}_q^{k\times(n-k)}$ and let $\mathbf{A} = (\mathbf{A}' \| \mathbf{A}'\mathbf{T}' + \mathbf{G})$. Let $\mathbf{y} = \mathbf{A}\mathbf{s}' + \mathbf{e}'$, with $\mathbf{s}' \leftarrow_\$ \mathbb{Z}_q^n$ and $\mathbf{e}' \leftarrow_\$ [-r, r]^m$. Also let $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow_\$ [-r, r]^m$. Then it holds that*

$$\Pr_{(\mathbf{A},\mathbf{y})}[H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq 1] \geq 1 - \mathsf{negl}(\lambda).$$

Proof. We first need to set the parameter ϵ for Lemma 7.5 appropriately. Let $\kappa' \leq \kappa$ be such that $\kappa' = 2^{\sqrt{\omega(\log(\lambda))}}$ and $\kappa' = \lambda/\omega(\log(\lambda))$, i.e. κ' is also upper-bounded by $\lambda/\omega(\log(\lambda))$. Clearly, as $\kappa' \leq \kappa$, the distribution χ is also B'-bounded with

$$B' = B \cdot \frac{\kappa}{\kappa'} \geq B.$$

Moreover, it holds that

$$r \geq m \cdot B \cdot \kappa = m \cdot B' \cdot \kappa'.$$

We can now set $\epsilon = 1/\kappa'$ (for Lemma 7.5). Let $\mathbf{g}$ be distributed according to χ^m. It holds by Lemma 7.5 that

$$\begin{aligned}
\Pr_{\mathbf{e}'}\left[\Pr_{\mathbf{g}}[\|\mathbf{e}' - \mathbf{g}\|_\infty \leq r] \geq \frac{1}{\kappa'}\right] &\geq 1 - m \cdot {\kappa'}^{-\log(r/(m\cdot B'))} \\
&\geq 1 - 2^{-(\log(\kappa'))^2} \\
&\geq 1 - 2^{-\omega(\log(\lambda))} \\
&\geq 1 - \mathsf{negl}(\lambda),
\end{aligned}$$

as it holds that $\log(\kappa') = \sqrt{\omega(\log(\lambda))}$. Thus it holds that

$$\Pr_{\mathbf{g}}[\|\mathbf{e}' - \mathbf{g}\|_\infty \leq r] \geq \frac{1}{\kappa'},$$

except with negligible probability. Assume henceforth that $\Pr_{\mathbf{g}}[\|\mathbf{e}' - \mathbf{g}\|_\infty \leq r] \geq 1/\kappa'$. Let $\mathbf{g}_1, \ldots, \mathbf{g}_{n-k}$ be the columns of $\mathbf{G}$. As $\mathbf{G}$ is chosen according to $\chi^{m\times(n-k)}$, each $\mathbf{g}_i$ is independently distributed according to χ^m. As the $\mathbf{g}_i$ are chosen independently according to χ^m, the probability that it holds for all $i = 1, \ldots, n-k$ that $\|\mathbf{e}' - \mathbf{g}_i\|_\infty > r$ is at most

$$\begin{aligned}
\Pr[\forall i : \|\mathbf{e}' - \mathbf{g}_i\|_\infty > r] &= \prod_{i=1}^{n-k} \Pr[\|\mathbf{e}' - \mathbf{g}_i\|_\infty > r] \\
&\leq (1 - 1/\kappa')^{n-k} \\
&\leq e^{-(n-k)/\kappa'} \\
&\leq e^{-\lambda/\kappa'} \\
&\leq e^{-\omega(\log(\lambda))} \\
&\leq \mathsf{negl}(\lambda),
\end{aligned}$$

as $n - k \geq \lambda$ and $\kappa' = \lambda/\omega(\log(\lambda))$. Thus, with overwhelming probability there exists an $\mathbf{s} \neq \mathbf{0}$ such that $\|\mathbf{e}' - \mathbf{G}\mathbf{s}\|_\infty \leq r$, where we can choose $\mathbf{s}$ to be one of the unit vectors. Together, this yields that

$$\Pr_{\mathbf{G},\mathbf{e}'}[\mathbf{e}' \text{ 2-ambiguous for } \mathbf{G}] \geq 1 - \mathsf{negl}(\lambda).$$

The same holds for the matrix $\mathbf{A} = (\mathbf{A}' \| \mathbf{A}'\mathbf{T}' + \mathbf{G})$, as we can obtain $\mathbf{A}$ from $\mathbf{G}$ by appending extra columns and applying a basis-change. Both operations straightforwardly do not decrease the ambiguity. Therefore it holds that

$$\Pr_{\mathbf{A},\mathbf{e}'}[\mathbf{e}' \text{ 2-ambiguous for } \mathbf{A}] \geq 1 - \mathsf{negl}(\lambda).$$

By linearity, this also holds if we shift $\mathbf{e}'$ by $\mathbf{A}\mathbf{s}'$ for any $\mathbf{s}' \in \mathbb{Z}_q^n$. As $\mathbf{y} = \mathbf{A}\mathbf{s}' + \mathbf{e}'$ we get

$$\Pr_{\mathbf{A},\mathbf{y}}[\mathbf{y} \text{ 2-ambiguous for } \mathbf{A}] \geq 1 - \mathsf{negl}(\lambda).$$

By Lemma 7.4, if $\mathbf{y}$ is 2-ambiguous for $\mathbf{A}$, then $H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s}+\mathbf{e} = \mathbf{y}) \geq 1$. Thus we get

$$\Pr_{(\mathbf{A},\mathbf{y})}[H_\infty(\mathbf{s}|\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{y}) \geq 1] \geq \Pr_{\mathbf{A},\mathbf{y}}[\mathbf{y} \text{ 2-ambiguous for } \mathbf{A}] \geq 1 - \mathsf{negl}(\lambda).$$

This concludes the proof. □

We will summarize the statements of Lemma 7.1, Corollary 7.3 and Lemma 7.6 in the following theorem.

Theorem 7.2. *Let λ be a security-parameter. Let $q = q(\lambda)$ be a modulus and $\alpha = \alpha(\lambda) \in (0,1)$. Let $m, n, k = \mathsf{poly}(\lambda)$ and $r = r(\lambda)$ be positive integers such that*

- $n - k \geq \lambda$
- $r \geq m \cdot \alpha \cdot q \cdot \kappa$ *for some* $\kappa = \kappa(\lambda) > 0$ *with* $\kappa = 2\sqrt{\omega(\log(\lambda))}$.
- $n \leq (1 - \log_q(4r+1) - \epsilon)m$ *for an arbitrarily small constant* $\epsilon > 0$

Provided that $\mathsf{LWE}(k, m, q, D_{\alpha q})$ is hard, the distribution $\mathcal{C}_{\alpha,k}$ given in Construction 7.1 is 1-lossy for the errors uniformly distributed on $[-r, r]^m$,

Proof. Let $\mathbf{L}$ be distributed according to $\mathcal{C}_{\alpha,k}$ and $\mathbf{U}$ be chosen uniformly from $\mathbb{Z}_q^{m \times n}$.

1. By Lemma 7.1 it holds that $\mathbf{L} \approx_c \mathbf{U}$, as we assume that $\mathsf{LWE}(k, m, q, D_{\alpha q})$ is hard.
2. As $n \leq (1 - \log_q(4r+1) - \epsilon)m$, it holds by Corollary 7.3 that $\mathbf{U}$ is non-lossy.
3. We can set $\kappa' = \sqrt{\kappa}$. Then it holds that $\kappa' = \sqrt{\omega(\log(\lambda))}$ and $\kappa' = \sqrt{\kappa} = 2\sqrt{\omega(\log(\lambda))}$. By Lemma 2.15, the discrete gaussian $D_{\alpha q}$ is $B = \alpha q \kappa'$-bounded. Thus it holds that

$$\begin{aligned} r &\geq m \cdot \alpha \cdot q \cdot \kappa \\ &= m \cdot \alpha \cdot q \cdot \kappa'^2 \\ &= m \cdot B \cdot \kappa'. \end{aligned}$$

   As $n - k \geq \lambda$, $\kappa' = 2\sqrt{\omega(\log(\lambda))}$, Lemma 7.6 yields that $\mathbf{L}$ is 1-lossy.

Consequently, $\mathcal{C}_{\alpha,k}$ is a distribution of 1-lossy codes for uniformly distributed errors on $[-r, r]^m$. □

7.5. Putting it all together

Using Theorems 7.1 and 7.2 we can translate the worst-case connection for standard LWE (Theorem 3.5) to LWE with uniform errors.

Theorem 7.3. *Let λ be a security parameter. Let $q = q(\lambda)$ be a modulus and $\alpha = \alpha(\lambda) \in (0,1)$. Let $m, n = \mathsf{poly}(\lambda)$ and $r = r(\lambda)$ be positive integers such that $n \geq 2\lambda$ and $r \geq m \cdot \alpha \cdot \kappa$ for $\kappa = \kappa(\lambda) = 2^{\sqrt{\omega(\log(\lambda))}}$. If $\mathsf{LWE}(n/2, q, D_{\alpha q})$ is hard, then $\mathsf{LWE}(n, m, q, [-r, r]^m)$ is also hard.*

Notice that the term $\kappa = 2^{\sqrt{\omega(\log(\lambda))}}$ can be chosen *sub-polynomial*. For instance, we can set

$$\kappa = 2^{(\log(\lambda))^{\frac{1}{2}+\epsilon}} = \lambda^{(\log(\lambda))^{-(\frac{1}{2}-\epsilon)}} = \lambda^{o(1)}$$

for a small constant $\epsilon \in (0, 1/2)$.

Proof of Theorem 7.3. Let $\epsilon > 0$ be an arbitrarily small integer and m^* be such that

$$m^* \geq \frac{n}{1 - \log_q(4r+1) - \epsilon}.$$

Assume that $m < m^*$ and that there exists an efficient PPT adversary $\mathcal{A}$ that solves $\mathsf{LWE}(n, m, q, [-r, r]^m)$ with non-negligible probability ϵ. Then there exists an efficient PPT adversary $\mathcal{A}'$ that solves $\mathsf{LWE}(n, m^*, q, [-r, r]^{m^*})$ with probability ϵ. $\mathcal{A}'$ basically discards the last $m - m'$ equations of its input instance and runs $\mathcal{A}$ on the so truncated instance. As the components of the error distribution $[-r, r]^{m^*}$ are independent, this this is a proper instance of $\mathsf{LWE}(n, m, q, [-r, r]^m)$, $\mathcal{A}$'s success probability is at least ϵ. Thus, it is sufficient to consider the case $m \geq m^* \geq \frac{n}{1-\log_q(4r+1)-\epsilon}$. Setting $k = n/2$, we get that $n - k \geq n/2 \geq \lambda$. By a search-to-decision reduction (Theorem 3.1 or 3.2), the hardness of $\mathsf{LWE}(n/2, q, D_{\alpha q})$ implies the hardness of $\mathsf{DLWE}(n/2, q, D_{\alpha q})$, which in turn immediately implies the hardness of $\mathsf{DLWE}(n/2, m, q, D_{\alpha q}^m)$. By Theorem 7.2 this implies that $\mathcal{C}_{\alpha, n/2}$ is 1-lossy. Theorem 7.1 then implies that $\mathsf{LWE}(n, m, q, [-r, r]^m)$ is also hard. This concludes the proof. □

Using the search-to-decision reduction of Theorem 3.1, we can establish the hardness of the decisional LWE problem with error-distribution $[-r, r]$, given that q is a polynomially small prime integer.

Corollary 7.7. *Let λ be a security parameter. Let $q = \mathsf{poly}(\lambda)$ be a prime modulus and $\alpha = \alpha(\lambda) \in (0,1)$. Let $m, n, r = \mathsf{poly}(\lambda)$ be positive integers such that $n \geq 2\lambda$ and $r \geq m \cdot \alpha \cdot \kappa$ for $\kappa = \kappa(\lambda) = 2^{\sqrt{\omega(\log(\lambda))}}$. If $\mathsf{LWE}(n/2, q, D_{\alpha q})$ is hard, then $\mathsf{DLWE}(n, m, q, [-r, r]^m)$ is also hard.*

By the worst-to-average case reductions (Theorem 3.5 and 3.6) for LWE with gaussian errors, we can conclude that $\mathsf{LWE}(n, m, q, [-r, r]^m)$ and $\mathsf{DLWE}(n, m, q, [-r, r]^m)$ (for prime $q = \mathsf{poly}(\lambda)$) are as hard as standard worst case lattice problems.

7.6. Further Developments

Concurrently and independently of our work, Micciancio and Peikert [MP13] established a worst-case connection for LWE with very short uniform errors. Specifically, [MP13] shows that a family of instantiations of LWE with short uniform errors, at most linear number of samples and polynomial modulus are as hard as approximating standard worst-case lattice problems to withing a factor of $\tilde{O}(\sqrt{n}q)$. For instance, their result can be instantiated with binary errors and $n \cdot (1+\Omega(1/\log(n)))$ samples or polynomial errors (n^ϵ for some small ϵ) and a linear number of samples ($m = (1+\epsilon/3)n$).

Moreover, Alwen, Krenn, Pietrzak and Wichs [AKPW13] independently provided a hardness reduction for the *learning with rounding* (LWR) problem [BPR12] and LWE with uniform errors which is, on a technical level, similar to ours presented in this Chapter. While their technique yields slightly worse connection factors than ours, it has stronger implications, yielding that LWE and LWR remain secure if the secret $\mathbf{s}$ is *weak and leaky*.

8. Conclusion and Prospects

We will conclude this thesis by reflecting on problems left open by the results of Chapters 5, 6 and 7, but also on directions to which our results point.

Chosen Ciphertext Security from McEliece.

In Chapter 5 we presented a chosen ciphertext secure McEliece cryptosystem which has a constant factor ciphertext expansion. However, this scheme does not provide a tight reduction to the McEliece problem. More specifically, the basic McEliece scheme has keys of size $O(\lambda^2)$ and ciphertexts of size $O(\lambda)$, while our scheme has keys of size $O(\lambda^3)$ and ciphertexts of size $O(\lambda^2)$. The reason for this quality-loss is that our tag-based encryption scheme uses $O(\lambda)$ basic McEliece instances in a monolithic way. This poses the question whether there is a construction of chosen ciphertext secure McEliece cryptosystem that uses at most $O(1)$ basic McEliece instances. We believe that in order to reach this goal, one will have to make *non-monolithic* use of the McEliece trapdoor, i.e. use structural properties of binary Goppa codes in an essential way.

Chosen Ciphertext Security from LPN

In Chapter 6 we have constructed a chosen ciphertext secure public key cryptosystem based on low noise LPN. The public and secret keys of this scheme have size $\tilde{O}(\lambda^4)$ and the ciphertexts have size $\tilde{O}(\lambda^2)$. This seems optimal, as even the best known constructions of semantically secure public key cryptosystems from this assumption have the same performance. The reason for this rather large dependence on the security parameter lies in the computational assumption we use. As we use LPN with a square-root amount of noise, we need to scale up the security parameter accordingly by a quadratic amount to ensure that the best known attack (in this case brute forcing) has complexity at least 2^λ. Thus, the quadratic dependence of the ciphertext size on the security parameter seems unavoidable if the underlying hardness assumption uses a square-root amount of noise.

Recently several alternatives to the standard LPN problem have been proposed, most notable the Ring-LPN problem [HKL+12] and the *linear time encoding*[1] LPN problem [DI14]. Both problems replace the multiplication with $\mathbf{A}$ by a more efficient operation in an algebraically richer structure. This also results in a more

[1] In [DI14] this problem is called CODE

efficient representation of the LPN instances, which reduces the size from $O(\lambda^4)$ to $O(\lambda^2)$. Constructions of semantically secure public cryptosystems can immediately be translated from standard LPN to these new LPN variants. However, our construction of a chosen ciphertext secure cryptosystem from LPN[2] makes explicit use of the *matrix structure* of the LPN problem and therefore does not translate to the new LPN variants. We thus regard the construction of efficient chosen ciphertext secure public cryptosystems from the new LPN variants as a problem of high practical relevance. First implementations [DP12] indicate that cryptosystems based on the new LPN variants have competitive performance, compared to state-of-the-art number theory based cryptosystems.

LWE with Uniform Errors

In Chapter 7 we showed that under certain conditions, LWE with uniform errors enjoys the same hardness guarantees as LWE with gaussian errors. However, in comparison to LWE with gaussian errors our guarantees for LWE with uniform errors are unsatisfactory in two ways.

1. While we can establish the hardness of the LWE search problem with uniform errors for any modulus, our reduction for decisional LWE basically relies on the search-to-decision reduction of Micciancio and Mol [MM11] (Theorem 3.1), which essentially restricts the modulus q to small powers of polynomially large primes. The search-to-decision reductions for LWE with gaussian errors in turn, allow for *hardware friendly* moduli that are powers of 2. To take full advantage of LWE with uniform errors, decisional LWE with moduli 2^k would be highly desirable.

2. From a theoretical point of view a more important drawback is that our hardness guarantee for LWE with uniform errors deteriorates with the number of samples. We stress that this is not the case for LWE with gaussian errors. We thus consider it an important question whether this limitation arises from an insufficient proof technique or whether this limitation is inherent. Compare this to the case of LWE with *binary* errors, i.e. the errors are chosen uniformly from $\{0, 1\}$. We immediately face a strict upper bound of $m = O(n)$ samples, as the algorithm of Arora and Ge [AG11] (c.f. Section 3.8.2) yields an efficient attack given more samples. For the case of binary errors, this is well in line with a recent result of Micciancio and Peikert [MP13] who showed that LWE with binary errors has a worst case hardness guarantee given that the number of samples is limited by $n + o(n)$.

[2] As well as a subsequent work [KMP14]

Bibliography

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[ACPS09] 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.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP (1)*, pages 403–415, 2011.

[AGHS13] Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete gaussian leftover hash lemma over infinite domains. In *ASIACRYPT (1)*, pages 97–116, 2013.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. In *CRYPTO*, pages 92–110, 2007.

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In *FOCS*, pages 120–129, 2011.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *In Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 99–108. ACM, 1996.

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. In *ICALP*, pages 1–9, 1999.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *CRYPTO (1)*, pages 57–74, 2013.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307, 2003.

[AS92] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; a new characterization of np. In *FOCS*, pages 2–13, 1992.

[ASP12] Jacob Alperin-Sheriff and Chris Peikert. Circular and kdm security for identity-based encryption. In *Public Key Cryptography*, pages 334–352, 2012.

[Bab85] László Babai. On loász' lattice reduction and the nearest lattice point problem (shortened version). In *Proceedings of the 2Nd Symposium of Theoretical Aspects of Computer Science*, STACS '85, pages 13–20, London, UK, UK, 1985. Springer-Verlag.

[BBP04] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT*, pages 171–188, 2004.

[BBS82] Lenore Blum, Manuel Blum, and Mike Shub. Comparison of two pseudo-random number generators. In *CRYPTO*, pages 61–78, 1982.

[BCGM07] Marco Baldi, Franco Chiaraluce, Roberto Garello, and Francesco Mininni. Quasi-cyclic low-density parity-check codes in the mceliece cryptosystem. In *ICC*, pages 951–956, 2007.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *EUROCRYPT*, pages 37–51, 1997.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.

[Ber73] Elwyn R. Berlekamp. Goppa codes. *IEEE Transactions on Information Theory*, 19(5):590–592, 1973.

[BFKL93] 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.

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. In *FOCS*, pages 16–25, 1990.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991.

[BFM88a] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.

[BFM88b] Manuel Blum, Paul Feldman, and Silvio Micali. Proving security against chosen cyphertext attacks. In *CRYPTO*, pages 256–268, 1988.

[BG84] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *CRYPTO*, pages 289–302, 1984.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003.

[BL12] Daniel J. Bernstein and Tanja Lange. Never trust a bunny. In *RFIDSec*, pages 137–148, 2012.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs #1. In *CRYPTO*, pages 1–12, 1998.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.

[BMVT78] E. Berlekamp, R.J. McEliece, and H. C A Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *Information Theory, IEEE Transactions on*, 24(3):384–386, May 1978.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, pages 719–737, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111, 1994.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. *IACR Cryptology ePrint Archive*, 2012:78, 2012.

[BRC60] R. C. Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, March 1960.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based fhe as secure as pke. In *ITCS*, pages 1–12, 2014.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *ASIACRYPT*, pages 157–174, 2001.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In *TCC*, pages 40–57, 2004.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *IACR Cryptology ePrint Archive*, 2010:591, 2010.

[CKS08] David Cash, Eike Kiltz, and Victor Shoup. The twin diffie-hellman problem and applications. In *EUROCRYPT*, pages 127–145, 2008.

[COT14] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild mceliece over quadratic extensions. *IACR Cryptology ePrint Archive*, 2014:112, 2014.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[Dam91] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *CRYPTO*, pages 445–456, 1991.

[DDMQN12] Nico Döttling, Rafael Dowsley, Jörn Müller-Quade, and Anderson C. A. Nascimento. A cca2 secure variant of the mceliece cryptosystem. *IEEE Transactions on Information Theory*, 58(10):6672–6680, 2012.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552, 1991.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DI14] Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the gilbert-varshamov bound and their cryptographic applications. In *ITCS*, pages 169–182, 2014.

[Din06] Irit Dinur. The pcp theorem by gap amplification. In *STOC*, pages 241–250, 2006.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009.

[DLZW13] Guanyang Deng, Hui Li, Ying Zhang, and Jun Wang. Tree-lshb+: An lpn-based lightweight mutual authentication rfid protocol. *Wireless Personal Communications*, 72(1):159–174, 2013.

[DMQ13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *EUROCRYPT*, pages 18–34, 2013.

[DMQN09] 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.

[DMQN12] 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.

[DORS08] 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.

[DP12] Ivan Damgård and Sunoo Park. Is public-key encryption based on lpn practical? *IACR Cryptology ePrint Archive*, 2012:699, 2012.

[FGK+09] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. *IACR Cryptology ePrint Archive*, 2009:590, 2009.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996.

[FGUO+13] Jean-Charles Faugère, Valérie Gauthier-Umaña, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate mceliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

[FM08] C. Faure and L. Minder. Cryptanalysis of the mceliece cryptosystem over hyperelliptic curves. *Proceedings of the eleventh International Workshop on Algebraic and Combinatorial Coding Theory*, page 99–107, June 2008.

[FOPT10] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of mceliece variants with compact keys. In *EUROCRYPT*, pages 279–298, 2010.

[For66] G. Forney. Generalized minimum distance decoding. *Information Theory, IEEE Transactions on*, 12(2):125–131, 1966.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[Gab05] P. Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryp- tography*, page 81–91, Bergen, Norway, March 2005.

[Gal63] Robert G. Gallager. Low-density parity-check codes, 1963.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[Gau66] C.F. Gauss. *Disquisitiones arithmeticae*. Yale paperbound. Yale University Press, 1966.

[GGH96] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems, 1996.

[GGH97a] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating decryption errors in the ajtai-dwork cryptosystem. In *CRYPTO*, pages 105–111, 1997.

[GGH97b] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '97, pages 112–131, London, UK, UK, 1997. Springer-Verlag.

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GHGKN06] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *CRYPTO*, pages 112–130, 2006.

[Gil52] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522, 1952.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS*, pages 102–113, 2003.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, pages 230–240, 2010.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377, 1982.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51, 2008.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, New York, NY, USA, 2004.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219, 1996.

[GS99] V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *Information Theory, IEEE Transactions on*, 45(6):1757–1767, Sep 1999.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92, 2013.

[Gur01] Venkatesan Guruswami. *List decoding of error correcting codes*. PhD thesis, 2001.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, pages 99–108, 2011.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[HB98] W. C. Huffman and Richard A. Brualdi. *Handbook of Coding Theory*. Elsevier Science Inc., New York, NY, USA, 1998.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *ASIACRYPT*, pages 52–66, 2001.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[HJ86] Roger A. Horn and Charles R. Johnson, editors. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1986.

[HK09] Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

[HKL+12] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An efficient authentication protocol based on ring-lpn. In *FSE*, pages 346–365, 2012.

[HLAWW13] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *EUROCRYPT*, pages 160–176, 2013.

[Hoc59] A. Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–158, 1959.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.

[Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[HW10] Susan Hohenberger and Brent Waters. Constructing verifiable random functions with large input spaces. In *EUROCRYPT*, pages 656–672, 2010.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 12–24, New York, NY, USA, 1989. ACM.

[JK03] J. Jonsson and B. Kaliski. Public-key cryptography standards (pkcs) #1: Rsa cryptography specifications version 2.1, 2003.

[JKPT12] 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.

[JM96] Heeralal Janwa and Oscar Moreno. Mceliece public key cryptosystems using algebraic-geometric codes. *Des. Codes Cryptography*, 8(3):293–307, 1996.

[JOP+01] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, pages 426–442, 2004.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[Kir11] Paul Kirchner. Improved generalized birthday attack. *IACR Cryptology ePrint Archive*, 2011:377, 2011.

[KLC+00] Ki Hyoung Ko, Sangjin Lee, Jung Hee Cheon, Jae Woo Han, Ju-Sung Kang, and Choonsik Park. New public-key cryptosystem using braid groups. In *CRYPTO*, pages 166–183, 2000.

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise lpn. In *Public Key Cryptography*, pages 1–18, 2014.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[KPC+11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In *EUROCRYPT*, pages 7–26, 2011.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the hfe public key cryptosystem by relinearization. In *CRYPTO*, pages 19–30, 1999.

[KS06] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the hb and hb$^+$ protocols. In *EUROCRYPT*, pages 73–87, 2006.

[KSS10] 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.

[Lam79] Leslie Lamport. Constructing digital signatures from one-way functions. *SRI intl. CSL-98*, 1979.

[LATV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234, 2012.

[LF06] Éric Levieil and Pierre-Alain Fouque. An improved lpn algorithm. In *SCN*, pages 348–359, 2006.

[Lin03] Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. In *EUROCRYPT*, pages 241–254, 2003.

[LLL82] H.W. jr. Lenstra, A.K. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*, pages 577–594, 2009.

[LMS+97] Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. Practical loss-resilient codes. In *STOC*, pages 150–159, 1997.

[LMSS98] Michael Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, and Daniel A. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *STOC*, pages 249–258, 1998.

[LP11] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA*, pages 319–339, 2011.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

[LS01] Pierre Loidreau and Nicolas Sendrier. Weak keys in the mceliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.

[Luc02] Stefan Lucks. A variant of the cramer-shoup cryptosystem for groups of unknown order. In *ASIACRYPT*, pages 27–45, 2002.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM*, pages 378–389, 2005.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. In *DSN Progress Report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA*, pages 114–116, 1978.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In *EUROCRYPT*, pages 419–453, 1988.

[Mic98] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *FOCS*, pages 92–98, 1998.

[Mic01] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In Joseph Silverman, editor, *Cryptography and Lattices Conference — CaLC 2001*, volume 2146 of *Lecture Notes in Computer Science*, pages 126–145, Providence, Rhode Island, 29–30March 2001. Springer-Verlag.

[Mic04] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. *SIAM J. Comput.*, 34(1):118–169, 2004.

[Min10] H. Minkowski. *Geometrie der Zahlen*. Number Bd. 1 in Geometrie der Zahlen. B.G. Teubner, 1910.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of lwe search-to-decision reductions. In *CRYPTO*, pages 465–484, 2011.

[MN73] Marshall McLuhan and Barrington Nevitt. The argument: Causality in the electric world. *Technology and Culture*, 14(1):1–18, 1973.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of sis and lwe with small parameters. In *CRYPTO (1)*, pages 21–39, 2013.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *FOCS*, pages 372–381, 2004.

[MR08] Daniele Micciancio and Oded Regev. Lattice-based cryptography, 2008.

[MRS86] Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems. In *CRYPTO*, pages 381–392, 1986.

[MRS00] C. Monico, J. Rosenthal, and A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT 2000*, page 215, 2000.

[MRY04] Philip D. MacKenzie, Michael K. Reiter, and Ke Yang. Alternatives to non-malleability: Definitions, constructions, and applications (extended abstract). In *TCC*, pages 171–190, 2004.

[MS07] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the sidelnikov cryptosystem. In *EUROCRYPT*, pages 347–360, 2007.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *ISIT*, pages 2069–2073, 2013.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.

[Nie85] Harald Niederreiter. A public-key cryptosystem based on shift register sequences. In *EUROCRYPT*, pages 35–39, 1985.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. In *Problems of Control and Information Theory. Problemy Upravlenija i Teorii Informacii. 15*, page 159–166, 1986.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, pages 111–126, 2002.

[NIKM08] Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Des. Codes Cryptography*, 49(1-3):289–305, 2008.

[NS00] Phong Q. Nguyen and Jacques Stern. Lattice reduction in cryptology: An update. In *ANTS*, pages 85–112, 2000.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

[OTD10] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. Cryptanalysis of two mceliece cryptosystems based on quasi-cyclic codes. *Mathematics in Computer Science*, 3(2):129–140, 2010.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In *STOC*, pages 109–118, 2011.

[Pat96] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342, 2009.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In *CRYPTO*, pages 80–97, 2010.

[PR07] Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *STOC*, pages 478–487, 2007.

[PS08] Krzysztof Pietrzak and Johan Sjödin. Weak pseudorandom functions in minicrypt. In *ICALP (2)*, pages 423–436, 2008.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[RS60] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[RS10] Markus Rückert and Michael Schneider. Estimating the security of lattice-based cryptosystems. *IACR Cryptology ePrint Archive*, 2010:137, 2010.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[RST07] Dima Ruinskiy, Adi Shamir, and Boaz Tsaban. Cryptanalysis of group-based key agreement protocols using subgroup distance functions. In *Public Key Cryptography*, pages 61–75, 2007.

[RSW96] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996.

[RU01] Thomas J. Richardson and Rüdiger L. Urbanke. Efficient encoding of low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):638–656, 2001.

[Rus36] B. Russell. *The Problems of Philosophy*. Home university library of modern knowledge. T. Butterworth limited, 1936.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

[SCO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, pages 566–598, 2001.

[Sen00] Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.

[Sha48] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[Sho94] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In *ANTS*, page 289, 1994.

[Sid94] V.M. Sidelnikov. A public-key cryptosytem based on reed-muller codes. *Discrete Mathematics and Applications*, 4:191–207, 1994.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions. pages 334–345. Springer-Verlag, 1998.

[Sin64] Richard C. Singleton. Maximum distance q-nary codes. *Information Theory, IEEE Transactions on*, 10(2):116–118, Apr 1964.

[Spi95] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. In *STOC*, pages 388–397, 1995.

[SS92] V. M. Sidelnikov and S. O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4), 1992.

[SS94] Michael Sipser and Daniel A. Spielman. Expander codes. In *FOCS*, pages 566–576, 1994.

[SS11] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47, 2011.

[TVZ82] M. A. Tsfasman, S. G. Vlădut, and Th. Zink. Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982.

[Val84] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984.

[Var57] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Acad. Nauk SSSR*, 117:739–741, 1957.

[vL99] J.H. van Lint. *Introduction to Coding Theory*. Graduate Texts in Mathematics. New York University Press, 1999.

[Wei] Eric W. Weisstein. Mean-value theorem. From MathWorld—A Wolfram Web Resource.

[WM84] Neal R. Wagner and Marianne R. Magyarik. A public key cryptosystem based on the word problem. In *CRYPTO*, pages 19–36, 1984.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.

[yCN97] Jin yi Cai and Ajay Nerurkar. An improved worst-case to average-case connection for lattice problems. In *FOCS*, pages 468–477, 1997.

List of Figures

Appendix

A. Lattices

Definition 8.1. *A discrete G subgroup of $(\mathbb{R}^n, +)$ is a a group such that for all distinct $\mathbf{x}, \mathbf{x}' \in G$ it holds that*

$$\|\mathbf{x} - \mathbf{x}'\|_2 \geq \lambda_1.$$

The notions of discrete subgroups of $\mathbb{R}^n$ and lattices in $\mathbb{R}^n$ are identical.

Lemma 8.1. *Every discrete subgroup G of $(\mathbb{R}^n, +)$ is a lattice, i.e. there exists a $\mathbf{B} \in \mathbb{R}^{n \times k}$ such that*

$$G = \Lambda(\mathbf{B}).$$

The converse follows directly by Lemma 2.13.

Proof. We will construct the basis $\mathbf{B}$ together with $r(\mathbf{B})$ inductively. To this end, we will define a sequence of bases $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_k$ and radii $r_0, \ldots, r_k$ with the property that $G \cap \mathsf{span}(\mathbf{B}_i) = \Lambda(\mathbf{B}_i)$ and forall $\mathbf{y} \in \mathsf{span}(\mathbf{B}_i)$ it holds that $\mathsf{d}(\mathbf{y}, \Lambda(\mathbf{B}_i)) \leq r_i$. Set $\mathbf{B}_0 = \emptyset$ and $r_0 = 0$, thus we have $G \cap \mathsf{span}(\mathbf{B}_0) = \Lambda(\mathbf{B}_0) = \{0\}$. We will now prove that the inductive step. So let $\mathbf{y} \in G \backslash \Lambda(\mathbf{B}_{i-1})$. Thus $\mathbf{y} \notin \mathsf{span}(\mathbf{B}_{i-1})$ and $\mathbf{y}$ has non-zero distance $\mathsf{d}(\mathbf{y}, \mathsf{span}(\mathbf{B}_{i-1}))$ from $\mathbf{B}_{i-1}$. Set $r^* = r_{i-1} + \mathsf{d}(\mathbf{y}, \mathsf{span}(\mathbf{B}_{i-1}))$. We claim that for every $\mathbf{x} \in G$ with $\mathsf{d}(\mathbf{x}, \mathsf{span}(\mathbf{B}_{i-1})) \leq \mathsf{d}(\mathbf{y}, \mathsf{span}(\mathbf{B}_{i-1}))$, there exists an $\mathbf{x}' \in G$ with $\mathsf{d}(\mathbf{x}', \mathsf{span}(\mathbf{B}_{i-1})) = \mathsf{d}(\mathbf{y}', \mathsf{span}(\mathbf{B}_{i-1}))$ and $\|\mathbf{x}'\| \leq r^*$. Let $\bar{\mathbf{x}}$ be the orthogonal projection of $\mathbf{x}$ on $\mathsf{span}(\mathbf{B}_{i-1})$, i.e. it holds that $\bar{\mathbf{x}} \in \mathsf{span}(\mathbf{B}_{i-1})$ and $\|\mathbf{x} - \bar{\mathbf{x}}\| = \mathsf{d}(\mathbf{x}, \mathsf{span}(\mathbf{B}_{i-1}))$. Now, by assumption it holds that $\mathsf{d}(\bar{\mathbf{x}}, \Lambda(\mathbf{B}_{i-1})) \leq r_{i-1}$, i.e. there exists a $\mathbf{z} \in \Lambda(\mathbf{B}_{i-1})$ such that $\|\bar{\mathbf{x}} - \mathbf{z}\| \leq r_{i-1}$. Set $\mathbf{x}' = \mathbf{x} - \mathbf{z} \in G$. By the triangle inequality it holds that

$$\begin{aligned}
\|\mathbf{x}'\| &= \|\mathbf{x} - \mathbf{z}\| \\
&\leq \|\mathbf{x} - \bar{\mathbf{x}}\| + \|\bar{\mathbf{x}} - \mathbf{z}\| \\
&\leq \mathsf{d}(\mathbf{x}, \mathsf{span}(\mathbf{B}_{i-1})) + r_{i-1} \\
&\leq \mathsf{d}(\mathbf{y}, \mathsf{span}(\mathbf{B}_{i-1})) + r_{i-1} \\
&= r^*
\end{aligned}$$

Now, there are only finitely many $\mathbf{x}' \in G \backslash \Lambda(\mathbf{B}_{i-1})$ with $\|\mathbf{x}'\| \leq r^*$. This is true because two distinct $\mathbf{x}_1, \mathbf{x}_2 \in G$ are separated at least by distance λ_1, i.e. $\|\mathbf{x}_1 - \mathbf{x}_2\| \geq \lambda_1$. Therefore, the spheres of radius $\lambda_1/4$ around $\mathbf{x}_1$ and $\mathbf{x}_2$ are disjoint. However, we can only pack a finite number of spheres of radius $\lambda_1/4$ in the sphere of radius r^*. Thus, there can only be a finite number of lattice points $\mathbf{x}'$ in the sphere of radius r^*. But from this follows immediately that the number of $\mathbf{x}' \in G \backslash \Lambda(\mathbf{B}_{i-1})$ with $\|\mathbf{x}'\| \leq r^*$ is finite. Therefore, there exists an $\mathbf{x}_{min} \in G \backslash \Lambda(\mathbf{B}_{i-1})$ with $\|\mathbf{x}_{min}\| \leq r^*$ such that for all $\mathbf{x} \in G \backslash \Lambda(\mathbf{B}_{i-1})$ it holds that $\mathsf{d}(\mathbf{x}_{min}, \mathsf{span}(\mathbf{B}_{i-1})) \leq \mathsf{d}(\mathbf{x}, \mathsf{span}(\mathbf{B}_{i-1}))$. We can now set $\mathbf{B}_i = (\mathbf{B}_{i-1} \| \mathbf{x}_{min})$ and set $r_i = \sqrt{r_{i-1}^2 + \mathsf{d}(\mathbf{x}_{min}, \mathsf{span}(\mathbf{B}_{i-1}))^2}$.

It remains to show that $G \cap \mathsf{span}(\mathbf{B}_i) = \Lambda(\mathbf{B}_i)$ and and for all $\mathbf{y} \in \mathsf{span}(\mathbf{B}_i)$ that $\mathsf{d}(\mathbf{y}, \Lambda(\mathbf{B}_i)) \leq r_i$. Let $\mathbf{x} \in \mathsf{span}(\mathbf{B}_i)$, i.e. let $\mathbf{x} = \mathbf{v} + \alpha \mathbf{x}_{min}$ with $\mathbf{v} \in \mathsf{span}(\mathbf{B}_{i-1})$ and $\alpha \in \mathbb{R}$. By adding an appropriate integer multiple of $\mathbf{x}_{min}$ to $\mathbf{x}$ we can obtain an $\mathbf{x}' \in G \cap \mathsf{span}(\mathbf{B}_i)$ with $\mathbf{x}' = \mathbf{v} + \alpha' \mathbf{x}_{min}$ with $\alpha' \in [-\frac{1}{2}, \frac{1}{2}]$.

First assume that $\mathbf{x}' \in G$. Then it holds that

$$\mathsf{d}(\mathbf{x}', \mathsf{span}(\mathbf{B}_{i-1})) \leq \alpha' \|\mathbf{x}_{min}\| \leq \frac{1}{2} \|\mathbf{x}_{min}\|,$$

which implies that $\alpha' = 0$, since otherwise $\mathbf{x}'$ would have smaller distance from $\mathsf{span}(\mathbf{B}_{i-1})$ than $\mathbf{x}_{min}$, contradicting the minimality of $\mathbf{x}_{min}$. Thus $\mathbf{x}_{min} = \mathbf{v} \in G \cap \mathsf{span}(\mathbf{B}_{i-1})$. But by the inductive assumption it holds that $\mathbf{v} \in \Lambda(\mathbf{B}_{i-1})$ and this yields $\mathbf{x} \in \Lambda(\mathbf{B}_i)$, proving the first part of the statement.

Now assume that $\mathbf{x}' \notin G$. Let $\bar{\mathbf{x}}'$ be the orthogonal projection of $\mathbf{x}'$ onto $\mathsf{span}(\mathbf{B}_{i-1})$. By the inductive assumption it holds that $\mathsf{d}(\bar{\mathbf{x}}', \Lambda(\mathbf{B}_{i-1})) \leq r_{i-1}$. Thus it holds that

$$\begin{aligned}
\mathsf{d}(\mathbf{x}, \Lambda(\mathbf{B}_i)) &= \mathsf{d}(\mathbf{x}', \Lambda(\mathbf{B}_i)) \\
&= \sqrt{\|\mathbf{x}' - \bar{\mathbf{x}}'\|^2 + \mathsf{d}(\bar{\mathbf{x}}', \Lambda(\mathbf{B}_{i-1})^2)} \\
&\leq \sqrt{\alpha'^2 \mathsf{d}(\mathbf{x}_{min}, \mathsf{span}(\mathbf{B}_{i-1}))^2 + r_{i-1}^2} \\
&\leq \sqrt{\mathsf{d}(\mathbf{x}_{min}, \mathsf{span}(\mathbf{B}_{i-1}))^2 + r_{i-1}^2} \\
&= r_i.
\end{aligned}$$

Thus, $\mathbf{x}$ is within distance r_i of $\Lambda(\mathbf{B}_i)$, which concludes the proof. □

Lebenslauf

02.11.1982	Geboren in Heilbronn als Sohn von Gustav und Erika Döttling
1989 - 1993	Besuch der Grundschule Affaltrach
1993 - 2002	Besuch des Hohenlohe Gymnasiums Öhringen
2002 - 2008	Studium der Informatik an der Universität Karlsruhe
2004	Vordiplom
2008	Diplom
seit 2008	Wissenschaftlicher Mitarbeiter am KIT