Institute of Information Security

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

A Tool for the Estimation of Lattice Parameters

Nicolai Krebs

Course of Study:	Informatik, B.Sc.
Examiner:	Prof. Dr. Ralf Küsters
Supervisor:	Marc Rivinius, M.Sc.
Commenced:	April 22, 2021
Completed:	October 22, 2021

Abstract

<Short summary of the thesis>

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms

1 Introduction

- rise of quantum computing (short history)

* conceptual

* reality

- problem: some hard classical problems no longer hard

* Shor's Algorithm (Peter Shor, 1994) => quantum computers can solve the factoring and the discrete logarithm problem in polynomial time

* application to encryption

* overview of current encryption methods that will become insecure

- one solution (among hash-based, code-based, isogeny-based, and multivariate): lattice crypto

* overview over history and capability of lattice crypto

* advantages: good (quasilinear) asymptotic key sized, good concrete runtimes and key sizes, worst-case secure instantiations, advanced cryptographic primitives previously infeasible

* including intro to LWE/SIS and applications to build crypto systems

. SIS: signature schemes, hash functions

. LWE: "cryptomania" applications (PKE, ...), signature schemes, lines:

- cryptographic applications

- establishing theoretical and asymptotic hardness [Reg05] [BLP+13; MP13] - concrete hardness of LWE: attacks, runtime estimates,

* briefly outline concept and benefits of hard-case to average-case reductions

- purpose of this thesis

* building schemes: need realistic hardness estimates of schemes for given parameter settings

* lack in the past: no unified/easy to use tool => thesis aims to solve this problem tool we call *Lattice Parameter Estimation* LWE instances are estimated by calling various estimation functions from the LWE Estimator [APS15], which we will refer to as *Estimator*.

- overview of chapters/how to read

2 Preliminaries

2.1 Notation

In the following, we denote vectors by bold lower-case letters like $\mathbf{v}$ and matrices by bold upper-case letters $\mathbf{M}$. We interchangably use matrix notation and sets of column vectors $[\mathbf{v}_1 \cdots \mathbf{v}_n] = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. Unless specified otherwise, by $\| \cdot \|$ or simply *norm* we refer to the Euclidean norm. By $[n]$ we denote the set $\{1, \ldots, n\}$ for $n \in \mathbb{Z}^+$.

2.2 Math

2.2.1 Lattices

- background and history: example from lecture -> change

* Birhoff [Bir40]

* cryptoanalysis [LLL82]

* cryptosystems [Ajt96, HPS98] SIS introduced Ajtai [Ajt96]

* [MR04]

* LWE, assumption: worst-case lattice problems are hard [Reg05]

* fully homomorphic [Gen09]

* BGV scheme [BV11, BGV12]

* tools [LPR10, LPR13] ideal latties, RLWE

Other Notes: - PKE [AD97; Reg03; Reg05], CCA security [Pei09; PW08], identity-based encryption [ABB10; CHKP10; GPV08], fully homomorphic [Gen09] - , LWE introduced by [Reg05] "provably as hard as certain lattice problems in worst case, appear to require time exponential in main security parameter to solve NTRU [HPS98] - q-ary lattice: modulus $q \geq 2$

- math * lattice Λ

- discrete additive subgroup of $\mathbb{R}^m$

- Let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ be a set of linearly independent basis vectors and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ be the corresponding basis with column vectors $\mathbf{b}_i$

- n is the dimension of the Lattice

- $\Lambda(\mathbf{B})$ defined by all integer combinations of elements of $\mathbf{B}$:

$$(2.1)\quad \Lambda(\mathbf{B}) = \left\{ x \in \mathbb{R}^m \mid \exists \alpha_1, \dots, \alpha_n \in \mathbb{Z} : \mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{b}_i \right\}$$

- show example plot

- full-ranked lattice: dimension is maximal, m

- basis $\mathbf{B}$ is not unique -> let $\mathbf{U} \in \mathbb{Z}^{n \times n}$ be a modular matrix (determinant is ±1), then $\mathbf{B} \cdot \mathbf{U}$ is also a basis of the Λ ($\mathbf{U} \cdot \mathbb{Z}^n = \mathbb{Z}^n$) -> different basis for the same lattice Λ

- lattice coset: quotient group $\mathbb{R}^n/\Lambda$ of cosets

$$\mathbf{c} + \Lambda = \mathbf{c} + \mathbf{v} \mid v \in \Lambda$$

with $\mathbf{c} \in \mathbb{R}^n$

- fundamental domain: subset of $\mathbb{R}^m$ containing exactly one representative of every coset

- (shifted) fundamental parallelipiped : $\mathcal{P}(\mathbf{B}) = \mathbf{B} \cdot [-1/2, 1/2)^n = \left\{ \mathbf{x} \in \mathbb{R}^m \mid \mathbf{x} \sum_{i=1}^{n} \gamma_i \mathbf{x}_i, \gamma_i \in [-1/2, 1/2) \right\}$ every coset has representative

- determinant of lattice $\Lambda(\mathbf{B})$: $\sqrt{\det(\mathbf{B}^\top \mathbf{B})}$. For a full-ranked lattice the determinant is

$$(2.2)\quad \det(\Lambda(\mathbf{B})) = |\det(\mathbf{B})|$$

is well-defined (independent from basis) => volume of fundamental domain can be generalized to not full-ranked => $\det(\Lambda(\mathbf{A})) = \sqrt{\det(\mathbf{A}^\perp \mathbf{A})}$

* minimum distance of $\lambda_1(\Lambda)$ of a lattice is the length of its shortest nonzero vector, i.e. $\lambda_1(\Lambda) \min_{v \in \Lambda \setminus \{0\}}$ * ith successive minimum $\lambda_i(\Lambda)$ - Let $r \in \mathbb{R}$ and $\mathbf{c} \in \mathbb{R}^m$, then we define $\mathcal{B}(\mathbf{c}, r)$ as the ball of radius r with center $\mathbf{c}$.

- smallest radius r such that the ball $\mathcal{B}(\mathbf{0}, r)$ centered at the origin of Λ contains i linearly independent lattice vectors.

- in general hard to calculate $\lambda_i(\Lambda(\mathbf{B}))$ for a given basis

* modular integer (or q-ary) lattices

- full-ranked lattice Λ such that $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$ given $q \in \mathbb{N}$ => if $\mathbf{x} \in \mathbb{Z}^m$ in Λ then $\mathbf{x} \mod q$ also in Λ.

- can be specified in two ways by matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$:

$$(2.3)\quad \Lambda_q(\mathbf{A}) = \{x \in \mathbb{Z}^m \mid \exists y \in \mathbb{Z}^n : \mathbf{x} = \mathbf{A}\mathbf{y} \mod q\}$$

or

$$(2.4)\quad \Lambda_q^\perp(\mathbf{A}) = \{x \in \mathbb{Z}^m \mid \mathbf{A}^\top \mathbf{x} = 0 \mod q\}$$

- finding a short vector in $\Lambda_q(\mathbf{A})$ corresponds to LWE

- finding short vectors in $\Lambda_q^\perp(\mathbf{A})$ corresponds to SIS

- easy to find basis of $\Lambda_q(\mathbf{A})$ [AFG13]

- with high probability determinant of q-ary lattice is $\det(\Lambda_q(\mathbf{A})) = q^{m-n}$ if $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$

* Gram-Schmidt basis

- set of column vectors $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$, $\pi_{\text{span}(\mathbf{B})}(\mathbf{t})$ for projection of vector $\mathbf{t}$ unto span of vectors of $\mathbf{B}$

- $\pi_{\text{span}(\mathbf{B})}(\mathbf{t}) = \mathbf{B}(\mathbf{B}^{\perp}\mathbf{B})^{-1}\mathbf{B}^{\intercal} \cdot \mathbf{t}$

- Gram-Schmidt orthogonalization $\tilde{\mathbf{B}} = \left[\tilde{\mathbf{b}}_1 \cdots \tilde{\mathbf{b}}_n\right]$ of basis $\mathbf{B}$: $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \pi_{\text{span}(\mathbf{b}_1,\ldots,\mathbf{b}_{i-1})}(\mathbf{b}_i)$ for $i \in \{1, \ldots, n\}$

- Gram-schmidt coefficients $\mu_{i,j} = \frac{\langle \tilde{\mathbf{b}}_j, \mathbf{b}_i \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$

Alternative: Let $\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_n], \mathbf{b}_i \in \mathbb{Z}_q^m$ be a basis. Define $\tilde{\mathbf{b}}_i$ as follows: $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$. For $i \in \{2, \ldots, n\}$ let $\tilde{\mathbf{b}}_i$ be the component of $\mathbf{b}_i$ that is orthogonal to the span of $\{\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\}$. Then, $\tilde{\mathbf{B}} = \left[\tilde{\mathbf{b}}_1 \cdots \tilde{\mathbf{b}}_n\right]$ is called the Gram-Schmidt orthogonalization of basis $\mathbf{B}$ where $\|\tilde{\mathbf{b}}_i\| \leq \|\mathbf{b}_i\|$.

* dual of a lattice is "the set of points whose inner products with the vectors in the lattice are integers"Λ: $\Lambda^{\perp} := \{\mathbf{y} \in \mathbb{R}^m \mid \forall \mathbf{v} \in \Lambda : \langle \mathbf{y}, \mathbf{v} \rangle \in \mathbb{Z}\}$ scaled-by-q dual lattice: $\{\mathbf{y} \in \mathbb{Z}^m \mid \forall \mathbf{v} \in \Lambda : \langle \mathbf{y}, \mathbf{v} \rangle = 0 \mod q\}$ basis of the dual of a lattice with basis $\mathbf{B}$ is $\mathbf{B}' = \mathbf{B}(\mathbf{B}^{\intercal}\mathbf{B})^{-1}$

* smoothing lemma

* Voronoi region The fundamental Voronoi region $\mathcal{V}$ is defined as

$$\mathcal{V} = \{\mathbf{x} \in \mathbb{R}^n \mid \forall \mathbf{y} \in \Lambda : \|\mathbf{x}\| \leq \|\mathbf{x} - \mathbf{y}\|\} \tag{2.5}$$

* Linear Code [Van12] Let $\mathbb{F}_q^n$ be the n-dimensional vector space over the field $\mathbb{F}_q$. A q-ary linear code C or $[n, k]$-code is a k-dimensional linear subspace of $\mathbb{F}_q^n$ such that

- $\mathbf{0} \in C$,
- if $\mathbf{x}, \mathbf{y} \in C$, then $\mathbf{x} + \mathbf{x} \in C$,
- and if $\mathbf{x} \in C$ and $\gamma \in \mathbb{F}_q$, then $\gamma\mathbf{x} \in C$.

There are q^k different codewords in C.

Let C be a q-ary linear $[n, k]$-code. The lattice over C is defined as

$$\Lambda(C) = \{\mathbf{x} \in \mathbb{R}^n \mid \exists \mathbf{y} \in C : \mathbf{x} = \mathbf{y} \mod q\}. \tag{2.6}$$

Similarly, for a lattice $\Lambda(\mathbf{B})$ a lattice code C defined by $\Lambda(\mathbf{B})$ and a shaping region $\mathcal{V} \subset \mathbb{R}^n$ (e.g. the Voronoi region) is a subspace of $\mathbb{R}^n$ such that all codewords are lattice vectors in $\Lambda(\mathbf{B})$ within the region $\mathcal{V}$ [SFS08]:

$$C' = \{x \in \Lambda(\mathbf{B}) \mid x \in \mathcal{V}\}. \tag{2.7}$$

We define $\text{dist}(\mathbf{t}, \Lambda(\mathbf{B}))$ where $\Lambda(\mathbf{B}) \subset \mathbb{R}^m$ as the distance of a vector $\mathbf{t} \in \mathbb{R}^m$ to the closest lattice vector $\mathbf{v} \in \Lambda(\mathbf{B})$, i.e.

$$\text{dist}(\mathbf{t}, \Lambda(\mathbf{B})) = \min_{\mathbf{v} \in \Lambda(\mathbf{B})} \|\mathbf{t} - \mathbf{v}\|. \tag{2.8}$$

- Lattice problems

* Minkowski theorem: Let Λ be a lattice of dimension n, then $\lambda_1 \leq \sqrt{n} \cdot (\det \Lambda)^{\frac{1}{n}}$

* Lattice reduction: find short basis compared to $\lambda_1(\Lambda)$...

* SVP: given a basis $\mathbf{B}$ of lattice Λ find shortest nonzero lattice vector => $v \in \Lambda$ s.t. $\|v\| = \lambda_1(\Lambda)$

Definition 2.2.1 (γ-approximate Shortest Vector Problem (SVP$_\gamma$))
Given a basis $\mathbf{B}$ *of lattice* Λ*, find a short lattice vector* $v \in \Lambda$ *such that* $0 < \|v\| \leq \gamma\lambda_1(\Lambda)$

Definition 2.2.2 (κ-approximate Hermite Shortest Vector Problem (HSVP$_\kappa$))
Given a basis $\mathbf{B}$ *of a lattice* $\Lambda(\mathbf{B}) \in \mathbb{R}^m$*, find a nonzero lattice vector* $\mathbf{v} \in \Lambda$ *such that* $\|\mathbf{v}\| \leq \kappa \cdot det(\Lambda)^{\frac{1}{n}}$.

* GapSVP$_\gamma$ (decision version of SVP): "given basis $\mathbf{B}$ of n-dimensional lattice Λ with either $\lambda_1\Lambda \leq 1$ or $\lambda_1\Lambda \geq \gamma(n)$, decide which is the case"NP hard for any constant γ fastest algorithm for $1 \leq \gamma \leq \text{poly}(n)$ has runtime complexity of $2^{O(n)}$

* γ-unique Shortest Vector Problem (uSVP$_\gamma$) [LM09]: given lattice Λ such that $\lambda_2(\Lambda) > \gamma\lambda_1(\Lambda)$, find shortest nonzero vector in $\mathbf{v} \in \Lambda$ with $\|\mathbf{v}\| = \lambda_1(\Lambda)$

* CVP$_\gamma$: given basis $\mathbf{B}$ of n-dimensional lattice Λ and target $\mathbf{t} \in \mathbb{R}^n$ find point in lattice that is close to $\mathbf{t}$ => find $\mathbf{v} \in \mathbb{R}^n$ with $\|\mathbf{t} - \mathbf{v}\| < \gamma \min_{\mathbf{v}' \in \Lambda} \|\mathbf{v}' - \mathbf{v}\|$

* SIVP (shortest independent vector problem): given basis $\mathbf{B}$ of n-dimensional lattice Λ, find n linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \Lambda(\mathbf{B})$ such that $\max_i \|\mathbf{v}_i\|$ for $i \in \{1, \ldots, n\}$ is minimal

* γ-Bounded Distance Decoding (BDD$_\gamma$): Given a lattice Λ and a target vector $\mathbf{t}$ such that $\text{dist}(\mathbf{t}, \Lambda) < \gamma\lambda_1(\Lambda)$, find the closest lattice vector $\mathbf{v} \in \Lambda$, i.e. find $\mathbf{v} = \min_{\mathbf{v}' \in \Lambda} \|\mathbf{v}' - \mathbf{t}\|$ minimal [LM09]

* ideal lattice (do I need that?)

* ...?

* eher die Sachen für LWE/SIS als die Sachen für Algorithmen (analog Vorlesung), evtl.

Intuition für die anderen Sachen...

2.2.2 Distributions

- Gaussian, def, component-wise, trafo to bound

* definition: discrete Gaussian distribution over q-ary lattice Λ with Gaussian width parameter $s > 0$ and center $\mathbf{c}$, denoted by $D_{\Lambda,s,\mathbf{c}}$: probability of sampling a vector $\mathbf{x} \in \Lambda$ is proportional to $e^{-\pi\|\mathbf{x}-\mathbf{c}\|^2/s^2}$. If $\mathbf{c} = \mathbf{0}$ we simply write $D_{\Lambda,s}$ In order to avoid confusion, throughout this work and in the *Lattice Parameter Estimation* we use σ to denote the standard deviation, where $\sigma = \frac{s}{\sqrt{2\pi}}$, and define $\alpha := \frac{s}{q} = \frac{\sqrt{2\pi}\sigma}{q}$.

* better definition in GPV08 => different definition needed for LWE??? * how to do this? => variant of Babai's "nearest-plane" algorithm, see [GPV08]

* component-wise

* smoothing factor here?

2.3 Two Important Problems

Applications: SIS can be used for one-way functions and collision-resistant hasing. LWE can be used to build pseudo-random number generators, public-key encryption schemes and oblivious transfer and secure MPC. Lattice Trapdoors (trapdoor functions, digital signatures)? Punctured Trapdoors (identity-based encryption, attribute-based encryption, predicate encryption)?

2.3.1 Learning with Errors (LWE)

Following based on [Reg10]:

Introduced by Regev in [Reg09] Origin: work of Ajtai and Dwork [AD97], first public-key cryptosystem based on worst-case lattice problems, simlifications/improvements [GGH97; Reg03] imply hardness result for LWE. Early work: hardness based on unique-SVP, Peikert [Pei09] and Lyubashevsky and Micciancio [LM09] show that unique-SVP is essentially equivalent to GapSVP.

- ´cryptomania´ applications: public-key encryption schemes under chosen-plaintext attacks [KTX07; PVW08; Reg05], and chosen-ciphertext attacks [Pei09; PW08], oblivious transfer protocoles [PVW08], identity-based encryption (IBE) schemes [ABB10; CHKP10; GPV08], leakage-resilient encryption [ACPS09; AGV09; DGK+10; GKPV10], and more

- most important: fully homomorphic encryption schemes [Bra12; BV11; Gen09; GSW13]

Intuition: - "recover $\mathbf{s} \in \mathbb{Z}_q^n$ given sequence of ´approximate´ random linear equations on $\mathbf{s}$ public matrix $\mathcal{A} \in \mathbb{Z}^{n\times m}$, secret vector $\int \in \mathbb{Z}^n$, given $\ddagger = \mathcal{A}^\intercal \int$ we can find $\mathbf{s}$ by linear algebra when we add a small error vector $\mathbf{e} \in \mathbb{Z}^m$, solving $\ddagger' = \mathcal{A}^\intercal \int + \mathbf{e}$ for $\mathbf{s}$ or distinguishing $\mathbf{z}'$ from uniform becomes hard

Formal Definition:

Definition 2.3.1 (LWE Distribution [Reg10])
For $n \geq 1$, modulus $q \geq 2$, error distribution χ on $\mathbb{Z}_q$, and a fixed secret vector $\mathbf{s}$, let $\mathcal{A}_{\mathbf{s},\chi}$ be the probability distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ by choosing a vector $\mathbf{a}_i \in \mathbb{Z}_q^n$ uniformly at random, $e_i \in \mathbb{Z}_q$ according to χ and returning pairs of $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i \mod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Additions are performed in $\mathbb{Z}_q$. We say that an algorithm solves LWE with modulus q and error distribution χ if, for any $\mathbf{s} \in \mathbb{Z}_q^n$, given an arbitrary number of independent samples from $\mathcal{A}_{\mathbf{s},\chi}$ it outputs $\mathbf{s}$ (with high probability). For $q = 2$ corresponds to *learning parity with noise* (LPN) problem.

Definition 2.3.2 (Search-LWE$_{n,q,m,\chi}$)
Search-LWE$_{n,q,m,\chi}$ asks for the recovery of the secret vector $\mathbf{s}$ *given m independent samples* $(\mathbf{a}_i, z_i) \leftarrow \mathcal{A}_{\mathbf{s},\chi}$

Definition 2.3.3 (Decision-LWE$_{n,q,m,\chi}$)
Given m samples, Search-LWE$_{n,q,m,\chi}$ asks to distinguish whether the samples were drawn from $\mathcal{A}_{\mathbf{s},\chi}$ or from a uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

LWE as a Decoding Problem

We request m samples $(\mathbf{a}_1, z_1), \ldots, (\mathbf{a}_m, z_m)$ where $z_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \in \mathbb{Z}_q$. Let $A = [\mathbf{a}_1 \cdots \mathbf{a}_m]$, $\mathbf{z} = [z_1, \ldots, z_m]^\intercal$ and $e = [e_1, \ldots, e_n]^\intercal$. Hence, we can reformulate LWE as a decoding problem as in [GJS15]:

(2.9) $\mathbf{z} = \mathbf{A}^\intercal \mathbf{s} + \mathbf{e}$

with generator matrix $\mathbf{A}$ for a linear code over $\mathbb{Z}_q$ and $\mathbf{z}$ as the received word. Finding the secret vector $\mathbf{z}$ is equivalent to finding the codeword $\mathbf{y} = \mathbf{A}^\intercal \mathbf{s}$ with minimum distance $\|\mathbf{y} - \mathbf{z}\|$.

An LWE$_{n,q,m,\chi}$ instance with a secret vector $\mathbf{s}$ chosen according to a uniform distribution can be transformed into an LWE$_{n,q,m-n,\chi}$ instance with a secret vector $\hat{\mathbf{s}}$ chosen according to the error distribution χ at a loss of n samples as follows: Let $\mathbf{A}_0 = [\mathbf{a}_1 \cdots \mathbf{a}_n]$ where $\mathbf{a}_1, \ldots, \mathbf{a}_n$ are the first n columns of $\mathbf{A}$. We introduce new variables $\hat{\mathbf{s}} = \mathbf{A}_0^\intercal \mathbf{s} - [z_1, \ldots, z_n]^\intercal = [e_0, \ldots, e_n]^\intercal$ and $\hat{\mathbf{A}} = \mathbf{A}_0^{-1} \mathbf{A} = [\mathbf{I}\ \hat{\mathbf{a}}_{n+1} \cdots \hat{\mathbf{a}}_m]$ and compute $\hat{\mathbf{z}} = \mathbf{z} - \hat{\mathbf{A}}^\intercal\ [z_1, \ldots, z_n]^\intercal = [\mathbf{0}, \hat{z}_{n+1} \cdots \hat{z}_m]^\intercal$.

LWE as a BDD Problem

Solving LWE also corresponds to solving the *Bounded Distance Decoding problem* (BDD) in the lattice $\Lambda(\mathbf{A}^\intercal) = \{\mathbf{x} \in \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}_q^n : \mathbf{x} = \mathbf{A}^\intercal \mathbf{s} \mod q\}$, where the m columns of $\mathbf{A}$ correspond to the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ of m independent LWE samples $(\mathbf{a}_i, z_i) \leftarrow \mathcal{A}_{\mathbf{s},\chi}$ and the components z_i correspond to a perturbed lattice point in $\Lambda(\mathbf{A}^\intercal)$.

Best algorithm to solve LWE: Blum, Kalai, and Wasserman [BKW03] with $2^{O(n)}$ samples and time.

Hardness: best algorithm exponential, extension of LPN (LPN believed to be hard), hard assuming worst-case hardness of GapSVP and SIVP [Pei09; Reg05]. More details? Different cases for q exponential/polynomial, approximation factors... Hardness based on worst-case lattice problems => strong security guarantees, such as conjectured security against quantum computers...

Search to decision reduction => distiguishing is LWE samples from uniform samples sufficient, worst-case to average-case reduction => sufficient to solve distinguishing for uniform secret

2.3.2 Short Integer Solution (SIS)

The dual problem to LWE is the *Short Integer Solution problem* (SIS).

- principle: given a set of set of uniformly random vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{Z}_q^n$ find a subset of them or combination with small coefficients that sums to zero (modulo q).

- introduced in [MR04], origins in [Ajt96], used for ´minicrypt´ primitives: one-way functions [Ajt96], collision resistant hash functions [GGH96], digital signature schemes [CHKP10; GPV08], and identification schemes [KTX07; Lyu08; MV03]

Definition 2.3.4 (SIS Problem (Adapted from [[LS15], Definition 3.1)
)] The problem $SIS_{n,q,m,\beta}$ is defined as follows: Given a uniformly random matrix $\mathbf{A}^{n\times m}$, find a vector $\mathbf{s} \in \mathbb{Z}_q^m$ such that $\mathbf{A} \cdot \mathbf{s} = 0$ mod q and $0 < \|\mathbf{s}\| \leq \beta$.

Finding such a vector corresponds to finding a short lattice vector in costets of the lattice $\Lambda^\perp(\mathbf{A}) = \{y \mid \mathbf{A} \cdot y \mod q\}$

Hardness: for any poly-bounded m, β and for "large enough" prime q: $\mathrm{SIS}_{n,q,m,\beta}$ is as hard as worst-case approx-SIVP (and GapSVP) to within $\beta \cdot \tilde{O}(\sqrt{n})$ factor

2.3.3 Ring and Module Variants

- problem key sizes in LWE/SIS in $O(n^2)$ (matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$), where $m \in \Omega(n)$)

- idea: introduce some sort of a structure in samples: n power of two, $\mathbf{a}$ vectors in groups of size n, for each group $\mathbf{a}_1 = [a_1, \ldots, a_n]^\top$, a_i are uniformly random in $\mathbb{Z}_q$, and $\mathbf{a}_i = [a_i, \ldots, a_n, -a_1, \ldots, -a_{i-1}]^\top$. Hence, n vectors only need $O(n)$ memory, also speedups in operations by using FFT

- formally: vectors are elements of the ring $\mathbb{Z}_q\,[x]\,/\langle x^n + 1\rangle$ which we call $\mathcal{R}_q$ instead of the group $\mathbb{Z}_q^n$, n power of two ensures that $x^n + 1$ is irreducible over the rationals

- add more?

Definition 2.3.5 (Ring-SIS Problem [[LS15], Definition 3.3)
)] The problem $RSIS_{n,q,m,\beta}$ is defined as follows: Given $a_1, \ldots, a_n \in \mathcal{R}_q$ chosen independently from the uniform distribution, find $s_1, \ldots, s_n \in \mathcal{R}$ such that $\sum_{i=1}^m a_i \cdot s_i = 0 \mod q$ and $0 < \|\mathbf{s}\| \leq \beta$, where $\mathbf{s} = [s_1, \ldots, s_m]^\top \in \mathcal{R}^m$.

Definition 2.3.6 (Module-SIS Problem [[LS15], Definition 3.3)
)] The problem $MSIS_{n,d,q,m,\beta}$ is defined as follows: Given $a_1, \ldots, a_n \in \mathcal{R}_q^d$ chosen independently from the uniform distribution, find $s_1, \ldots, s_n \in \mathcal{R}$ such that $\sum_{i=1}^m a_i \cdot s_i = 0 \mod q$ and $0 < \|\mathbf{s}\| \leq \beta$, where $\mathbf{s} = [s_1, \ldots, s_m]^\top \in \mathcal{R}^m$.

3 Algorithms and Estimates

3.1 Lattice Basis Reduction

Problem: usually ugly basis (long vectors...), we want a better basis with shorter and more orthogonal basis vectors... - improve lattice basis quality => measure by hermite factor (compare shortest vector in basis to lattice volume) or approximation factor (compare shortest vector in basis to shortest lattice vector) - algorithm finding vector with approximation factor γ can be used to solve uSVP with gap $\lambda_2(\Lambda)/\lambda_1(\Lambda) > \gamma$ - best known theoretical bound by Slide reduction [GN08a], BKZ better in practice

- measure quality of basis: Hermite factor

* basis $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, n-dimensional lattice $\Lambda(\mathbf{B})$ has root Hermite factor δ if

(3.1) $||\mathbf{b}_1|| \approx \delta^n \det(\Lambda)^{1/n}$

* use Geometric Series Assumption (GSA) [Sch03] to obtain estimates for $\mathbf{b}_i$:

(3.2) $||\tilde{\mathbf{b}}_i|| \approx \alpha^{i-1} ||\mathbf{b}_1||$

for $0 < \alpha < 1$ Equation (3.1) into Equation (3.2) -> $||\tilde{\mathbf{b}}_i|| \approx \alpha^{i-1}\delta^m \det(\Lambda)^{1/m}$ with $\prod_{i-1}^m ||\tilde{\mathbf{b}}_i|| = \det(\Lambda)$ we get

$$\begin{aligned} & \prod_{i-1}^{m} ||\tilde{\mathbf{b}}_i|| \approx \prod_{i-1}^{m} \alpha^{i-1}\delta^m \det(\Lambda)^{1/m} \\ \iff \quad & \det(\Lambda) \approx \delta^{2m} \det(\Lambda) \prod_{i-1}^{m} \alpha^{i-1} \\ \iff \quad & \delta^{-m^2} \approx \alpha^{\frac{m(m-1)}{2}} \\ \iff \quad & \delta^{-2} \approx \alpha^{(m-1)/m} \end{aligned}$$

Hence, $alpha \approx \delta^{-2}$ and

(3.3) $||\tilde{\mathbf{b}}_i|| \approx \delta^{-2(i-1)+m} \det(\Lambda)^{1/m}$

* good basis -> first Gram-Schmidt vectors become shorter (latter longer)

* $\delta = 1.01$ feasible, $\delta = 1.007$ seems infeasible for now

* gap between provable and experimental cost estimate to reach some hermite δ => provable results only give upper bounds, for practical security we need lower bound => combine theoretical results with experimental results

* well-established estimate [LP11]

In the following, we will focus on two related methods for lattice reduction. First, we define some reduction criterias following [ABLR21]. A basis $\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_n]$ is *size-reduced* if its Gram-Schmidt coefficients (see Section 2.2.1) satisfy $|\mu_{i,j}| \leq \frac{1}{2}$ for all $0 \leq j < i < n$. If the first basis vector $\mathbf{b}_1$ of $\mathbf{B}$ is the shortest lattice vector, i.e. $\|\mathbf{b}_1\| = \lambda_1(\Lambda(\mathbf{B}))$, we call $\mathbf{B}$ *SVP-reduced*. If a basis $\mathbf{B}$ is size-reduced and in addition each block $\{\mathbf{b}_i, \ldots, \mathbf{b}_n\}$ for $i = 1, \ldots, n$ of basis vectors is SVP-reduced then $\mathbf{B}$ is *HKZ-reduced*. We will see in the next section that size-reduction is closely related to the LLL algorithm and a special case of the BKZ reduction outputs an HKZ-reduced basis.

3.1.1 The LLL Algorithm

The LLL algorithm was proposed by Lenstra, Lenstra and Lovász [LLL82] and can be considered as a generalization of the two dimensional Lagrange reduction. The lagrange reduction reduces a basis of two basis vectors such that output basis satisfies $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$ and $\frac{|\langle \mathbf{b}_1, \mathbf{b}_2 \rangle|}{\|\mathbf{b}_1\|} = |\mu_{2,1}| \leq \frac{1}{2}$). Intuitively, a multiple of the shorter vector $\mathbf{b}_1$ is subtracted from the longer vector $\mathbf{b}_2$ such that the resulting vector $\mathbf{b}_2'$ is as orthogonal to $\mathbf{b}_0$ as possible, i.e. $\mathbf{b}_1' = \mathbf{b}_1 - \lfloor \mu_{1,0} \rceil \mathbf{b}_0$. We set $\mathbf{b}_2 = \mathbf{b}_2'$ and repeat until nothing changes.

A δ-LLL reduced basis ensures two criterias [LLL82]:

1. Size reduced: $|\mu_{i,j}| \leq \frac{1}{2}$ for $1 \leq i \leq n$ and $j < i$
2. Lovász condition: $\delta \|\tilde{\mathbf{b}}_i\|^2 > \|\mu_{i+1,i} \tilde{\mathbf{b}}_i + \tilde{\mathbf{b}}_{i+1}\|^2$ for $1 \leq i < n$

Recall the definition of the Gram-Schmidt coefficients $\mu_{i,j} = \frac{\langle \tilde{\mathbf{b}}_j, \mathbf{b}_i \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$. The LLL algorithm shown in Algorithm 1 follows the notation in [Reg04]. We start by computing the Gram-Schmidt orthogonalization of the input basis (Line 2) and continue with a reduction step in which we update every basis vector $\mathbf{b}_i$ by pairwisely comparing and subtracting lower indexed basis vector just as in the Lagrange reduction (Line 5) to ensure Criteria 1. Finally, vectors violating the Lovász condition are swapped (Line 7ff) and the process is repeated until nothing changes. The LLL algorithm can be used to find short vectors of at most $2^{n/2}\lambda_1(\Lambda)$ in polynomial time. Several floating-point variants have been suggested that can significantly speed up the runtime of LLL. For example, L^2 runs in $O(n^2 \log^2 B)$, where B is a bound on the norm of the input basis vectors [NS05].

3.1.2 The BKZ Algorithm

The Block Korkin-Zolotarev (BKZ) algorithm was proposed by Schnorr in 1987 and adapted by Schnorr and Euchner in [SE91] and represents a family of lattice reduction algorithm. Essentially, BKZ iteratively divides the input basis into blocks of a lower dimension k and calling an SVP oracle on each block. The output of the oracle is then used to obtain a basis of improved quality.

Algorithm 1: The δ-LLL Algorithm [LLL82]

```
function $\delta$-LLL($\mathbf{B} \in \mathbb{Z}^{m \times n}$)
    Compute $\tilde{\mathbf{B}}$
    for $i = 2, \ldots, n$ do
        for $j = i - 1, \ldots, 1$ do
            $\mathbf{b}_i = \mathbf{b}_i - \lfloor \mu_{i,j} \rceil \mathbf{b}_j$
    if $\exists i$ such that $\delta \|\tilde{\mathbf{b}}_i\|^2 > \|\mu_{i+1,i} \tilde{\mathbf{b}}_i + \tilde{\mathbf{b}}_{i+1}\|^2$ then
        Swap $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$
        Return $\delta$-LLL($\mathbf{B}$)
    else
        Return $\mathbf{B}$
```

Algorithm 2 presents the main concept of BKZ and follows the description in [CN11] with some adjustments. Initially, we run an LLL reduction on the input basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ and update the basis. In each jth iteration, we consider a block of k basis vectors $\mathbf{b}_j, \ldots, \mathbf{b}_{j+k-1}$. The vectors of the current block are projected onto the orthogonal complement of the span of vectors from previous iterations span $(\{\mathbf{b}_i \mid i \in [j-1]\})$ (Line 6 - 9, we skip this step if the span is empty). Note that the orthogonal complement $A^\perp$ of a subspace A is defined as the set of all vectors that are orthogonal to every vector in A. We then run an SVP oracle on the projected block to obtain a shortest vector $\mathbf{b}'_{\text{new}}$ in the projected lattice (Line 12) and reconstruct a lattice vector $\mathbf{b}_{\text{new}}$ of which $\mathbf{b}'_{\text{new}}$ is a projection Line 13. Note that in practice, the SVP oracle should include this step. If $\mathbf{b}_{\text{new}}$ is a new vector we insert it in our list of basis vectors before $\mathbf{b}_j$. Otherwise as nothing changed, we increment a counter z. Finally, we run LLL on all basis vectors up to index $j + i$ (including the possibly newly added vector). If no new lattice vectors can be found in n iterations, the reduction terminates. After n iterations, j is reset to start over at the first block. The ouput of the algorithm is a BKZ$_k$-reduced basis. For $k = 2$ we obtain an LLL-reduced basis in polynomial time and for $k = n$ an optimally HKZ-reduced basis in at least exponential time.

Several improvements have been suggested. The total number of rounds until termination is unknown and can be quite large. Hanrot *et al.* [HPS11] show an *early termination* of BKZ still yields a very good output basis quality and propose $\frac{n^2}{k^2} \log n$ rounds as a bound.

Local preprocessing increases the quality of the current block basis by recursively calling BKZ with smaller block size. A variant known as *progressive BKZ* applies the recursion globally [AWHT16].

If enumeration is used as an SVP oracle, the size of the search space can be reduced by means of *pruning* techniques. Nodes closer to the edges of the enumeration tree are less likely to represent in short lattice vectors. For more details on enumeration, we refer to Section 3.1.3. Gamma *et al.* show that applying a variant of this known as *extreme pruning* can reduce the running time by a much larger factor than the success probability. Repeating the search yields the desired speedup [GNR10].

In addition, [CN11] optimizes the *enumeration radius* by using experimental results. BKZ 2.0 incorporates a number of these techniques [CN11].

Algorithm 2: The BKZ Algorithm [SE91]

```
**function** BKZ($\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}, k \in [n]\backslash\{1\}$)
    z = 0; j = 0
    $\mathbf{B}$ = LLL($\mathbf{B}$)
    **while** $z < n - 1$ **do**
        $j = (j \mod (n-1)) + 1; l = \min(j + k - 1, n); h = \min(l + 1, n)$
        $A = \mathrm{span}\left(\{\mathbf{b}_i | i \in [j-1]\}\right)$
        **for** $i \in \{j, ..., l\}$ **do**
            **if** $A \neq \emptyset$ **then**
                $\mathbf{b}'_i = \pi_{A^\perp}(\mathbf{b_i})$
            **else**
                $\mathbf{b}'_i = \mathbf{b}_i$
        $\mathbf{b}'_{\text{new}}$ = SVP-Oracle($\mathbf{b}'_j, \ldots, \mathbf{b}'_l$)
        Reconstruct $\mathbf{b}_{\text{new}} = \sum_{i=j}^{l} \alpha_i \mathbf{b}_i$ with $\alpha_i \in \mathbb{Z}$ such that $\mathbf{b}'_{\text{new}} = \pi_{\left(\text{span}\left(\mathbf{b}_j,\ldots,\mathbf{b}_l\right)\right)^\perp}(\mathbf{b}_{\text{new}})$
        **if** $\mathbf{b}'_{new} \neq \tilde{\mathbf{b}}_j$ **then**
            $z = 0$; $\left\{\mathbf{b}_j, \ldots, \mathbf{b}_h\right\} = \mathrm{LLL}(\left\{\mathbf{b}_j, \ldots, \mathbf{b}_{j-1}, \mathbf{b}_{\text{new}}, \mathbf{b}_j, \ldots, \mathbf{b}_h\right\})$
        **else**
            $z = z + 1$; $\left\{\mathbf{b}_j, \ldots, \mathbf{b}_h\right\} = \mathrm{LLL}(\left\{\mathbf{b}_j, \ldots, \mathbf{b}_h\right\})$
```

It is difficult to find hard runtime bounds for BKZ. The upper bound on the number of rounds is superexponential in the dimension n for a fixed block size [GN08b; HPS11] before BKZ terminates given that no early termination strategy is used. In addition, calls to the SVP oracle in all dimensions $k' \leq k$ must be taken into account. Albrecht *et al.* ignore these intricacies and estimate the cost of BKZ in clock cycles as $\rho \cdot n \cdot t_k$ where ρ is the number of rounds needed and t_k is the cost (in block cycles) of calling the SVP oracle on a block of dimension k Albrecht et al. [APS15]. The value ρ is set to 8 in the *Estimator* and is derived from experiments in [Che13] that indicate that the most significant progress happens in the first $7 - 9$ rounds.

3.1.3 Cost Models for Lattice Reduction

In this section, we will look at various high level ideas to realize an SVP solver that can be used as a subroutine in BKZ and present up-to-date cost models from the literature. SVP is known to be NP-complete even for large constant approximation factors [Ajt98; Kho05]. An exponential approximation factor can be achieved in polynomial time but is mostly insufficient for practical purposes [LLL82]. We will mainly focus on two classes of (nearly) exact SVP solvers, namely, enumeration algorithms and sieving algorithms. Enumeration algorithms can solve SVP in a lattice of dimension k in $2^{O(k \log k)}$ time and polynomial space. Sieving algorithms only need $2^{O(k)}$ time, however, at the cost of exponential memory complexity. Only recently, progress in sieving strategies has given rise to BKZ implementations relying on sieving (e.g. the General Sieve Kernel (G6K) impelmentation [ADH+19; DSW21]) that outperform enumeration implementations already in relatively small dimensions $\gtrsim 70$ in the classical setting [ABLR21]. On the other hand, quantum speedups for enumeration are greater than for sieving. Aono *et al.* show a quadratic cost reduction for enumeration Aono et al. [ANS18], while the cost sieving only decreases by a factor of $2^0.027$ with idealized assumptions [Laa16]. The authors of [ADPS16] argue that due of a lower bound $2^0.2075k$ on the required size of the building lists future quantum sieving algorithms are not expected to achieve an asymptotic runtime below $2^0.2075k$.

A selection of the most relevant cost models for cryptographic purposes is shown Table 3.2. All these cost models are supported in our tool.

Enumeration

Enumeration aims to find the shortest vector by enumerating all lattice vectors within some bounded region. In general, we start with reducing the lattice basis to improve the basis quality. We then define a bound and iteratively project the lattice to the span of its Gram-Schmidt vectors beginning from $\tilde{\mathbf{b}}_n$ until we arrive at the lowest level of a one-dimensional subspace. We continue by enumerating all vectors of norm less than r in the projected lattice and "lift" each of these vectors to the level above and repeat this process until we arive at the level from which we started. The search space can be thought of as a large tree of (projected) vectors on which we apply depth-first search. Note that the root of the tree here is at the lowest level and the leafs are the lattice vectors in our target lattice. The low memory cost of enumeration is due to its similarities to depth-first search.

A very early but very efficient variant was suggested by Kannan [Kan83] with a proven worst-case runtime of $2^{O(k \log k)}$. BKZ$_k$ using Kannan's enumeration algorithm as SVP oracle yields a short lattice vector of norm approximately $\left(k^{\frac{1}{2k}}\right)^n \cdot \mathrm{Vol}(\Lambda)^{1/n}$ or equivalently achieves a root Hermite factor of $k^{\frac{1}{2k}}$ [ABF+20; HS07].

In [ABF+20] we find a more concrete experimental cost model of $\mathrm{poly}(n) \cdot 2^{\frac{k \log k}{2e} - 0.995k + 16.25}$ for BKZ 2.0 (see Section 3.1.2), where $\mathrm{poly}(n)$ is the number of calls to the enumeration subroutine. BKZ 2.0 achieves a root Hermite factor of $\left(\frac{k}{2\pi e} \cdot (\pi k)^{1/k}\right)^{\frac{1}{2(k-1)}}$ [Che13].

The FastEnum algorithm in Albrecht et al. [ABF+20] incorporates an idea called "extended preprocessing" and simulations achieve a root Hermite factor of $k^{\frac{1}{2k}(1+o(1))}$ in $\mathrm{poly}(n) \cdot 2^{0.125k \log k - 0.050k + 56}$ time. The corresponding quantum algorithm reduces the runtime from $2^{\frac{k \log k}{8} + o(k)}$ to $2^{\frac{k \log k}{16} + o(k)}$. In extended preprocessing, instead of preprocessing the current projected basis block of size k, the BKZ-reduction is applied to a block of higher dimension $\lceil (1 + c) \cdot k \rceil$ for some constant c. Enumeration is faster on the first basis vectors as their Gram-Schmidt norms closely follow the Geometric Series Assumption [MW16].

A tradeoff of runtime and success probability for "relaxing" the approximation and extreme pruning turns out to exponentially speed up the search [LN20] and was combined with extended preprocessing by Albrecht *et al.* to further reduce the experimental runtime of BKZ to $\mathrm{poly}(n) \cdot 2^{\frac{k \log k}{8} - 0.654k + 25.84}$ for a root Hermite factor of $k^{\frac{1}{2k0}}$ [ABF+20].

Sieving

The second group of SVP solvers are sieving algorithms [ADH+19; BDGL16; BGJ15; BLS16; HK17; MV10; NV08]. In sieving, initially, we create a long list of randomly selected lattice points. The points in the list are then combined or "reduced" in some way to find points of smaller length. One way to achieve this is by finding a minimimal sublist of "center" points in the initial list such that spheres centered at these points cover all list points. Subtracting the center points yields short lattice points. ListSieve [MV10] uses a smaller initial list to divide the space into two half-spaces, one closer to the center and one closer to the respective point. The list is then used to reduce the length of newly sampled points as much as possible by subtracting each list vectors such that the result is located in the half-space closer to the center respectively. Once two points with a distance less than the target distance are found, they are subtracted and the result is returned.

Algorithm	Cost Estimate
List Sieve [MV10]	$2^{0.3199n+o(n)}$ time, $2^{0.1325+o(n)}$ memory
NV-sieve [ADH+19; NV08]	$2^{0.415n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory
NV-sieve (quantum) [ADH+19; NV08]	$2^{0.311n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory
Gauss sieve [HK17; MV10]	$2^{0.415n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory
BGJ-sieve [BGJ15]	$2^{0.311n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory
3-sieve [BLS16; HK17]	$2^{0.3962n+o(n)}$ time, $2^{0.1887n+o(n)}$ memory
BDGL-sieve [BDGL16]	$2^{0.292n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory
BDGL-sieve (quantum) [BDGL16]	$2^{0.265n+o(n)}$ time, $2^{0.2075n+o(n)}$ memory

Table 3.1: Overview of Popular Sieving Algorithms

Table 3.1 presents a list of currently best sieving algorithms. Note that some runtime and space complexities are only conjectured and not proven yet.

In the Nguyen-Vidick sieve [NV08], we iteratively reduce a pair of list points whose combined length is smaller than the longest list vector. The longest vector is then replaced by the result. The list length is fixed. In the Gauss sieve [MV10], we start with an empty list and a stack. In each step, a new point is either sampled or taken from the stack. We then attempt to reduce the new point with all points in the list. If a reduction is successful, the longer vector of the pair is replaced. If the longer vector was the list point, the replacement is inserted in the stack. If no reduction is possible, the stack points are moved back to the list. If the stack is empty, all list points are reduced pairwisely. In practice, the Gauss sieve outperforms Nguyen-Vidick sieve. The Becker-Gama-Joux sieve [BGJ15] exploits coding theory to find vectors that are likely to be nearest neighbors. Similar vectors are stored in the same bucket to speed up the search for reduction candidates. The 3-sieve [BLS16; HK17] reduces the required list size by using triples instead of pairs of points for combination. Finally, the Becker-Ducas-Gama-Laarhoven sieve [BDGL16] applies locality sensitive hashing to create buckets of points in near neighborhood similar to the Becker-Gama-Joux sieve.

Name	Reference	Cost model
Sieving		
Q-Sieve (paranoid lower bound)	[ADPS16]	$2^{0.2075k}$
Q-Sieve	[ADPS16; AGPS20; Laa16]	$2^{0.265k}$
Q-Sieve + O(1)	[SAL+17]	$2^{0.265k+16}$
Q-Sieve (min space)	[SHRS17]	$2^{0.2975k}$
Sieve	[ADPS16; AGPS20; BDGL16]	$2^{0.292k}$
Sieve + O(1)	[SAL+17]	$2^{0.292k+16}$
Sieve (min space)	[SHRS17]	$2^{0.368k}$
Enumeration		
Lotus	[ACD+18; PHAM17]	$2^{0.125k \log k - 0.755k + 2.254}$
Enum + O(1)	[ACD+18; Che13; SHRS17]	$2^{0.187k \log k - 1.019k + 16.1}$
Q-Enum + O(1)	[ACD+18; Che13; SHRS17]	$2^{0.0936k \log k - 0.51k + 8.05}$
BCLV-Enum (quadratic fit) + O(1)	[BCLV17]	$2^{0.000784k^2 + 0.366k + 0.875}$
BKZ2.0-Enum	[ABF+20; Che13; CN11]	$2^{0.184k \log k - 0.995k + 16.25}$
ABF-Enum	[ABF+20]	$2^{0.125k \log k}$
ABF-Enum + O(1)	[ABF+20]	$2^{0.125k \log k - 0.547k + 10.4}$
Q-ABF-Enum	[ABF+20]	$2^{0.0625k \log k}$
ABLR-Enum + O(1)	[ABLR21]	$2^{0.125k \log k - 0.654k + 25.84}$

Table 3.2: SVP Cost Models Overview (based on Table 4 in [ACD+18])

3.2 Algorithms for Solving LWE

3.2.1 Overview

Distinguishing attacks (MR09, RS10): distinguish (with noticeable advantage) LWE instance from uniformly random => break semantic security of LWE-based cryptosystem with same advantage (typically), find short nonzero integral vector $\mathbf{v}$ s.t. $\mathbf{A}^\intercal \mathbf{v} = 0 \mod q$ => short vector in (scaled) dual of LWE lattice $\Lambda(\mathbf{A})$ then test whether $\langle \mathbf{v}, z \rangle$ is close to zero mod q. If uniform test accepts with prob 1/2, if LWE with parameter s, $\langle \mathbf{v}, \mathbf{z} \rangle = \langle \mathbf{v}, \mathbf{e} \rangle \mod q$, Gaussian mod q with parameter $\|\mathbf{v}\| \cdot s$. If that's not much larger than q, advantage for distinguishing very close to $\exp(-\pi(\|\mathbf{v}\| s/q)^2)$. high confidence needs $\|\mathbf{v}\| \leq q/(2s)$ advantage an computational effort need to be balanced (often inverse distinguising advantage is in total cost of attack)

Dual Attacks

reduce LWE to SIS recover secret vector by finding a short vector in the dual lattice $\Lambda(\mathbf{A}^\intercal)^\perp = \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{y} = \mathbf{0} \mod q\}$ generated by the rows of $\mathbf{A}$ and scaled by q.

Primal Attacks

lattice reduction algorithms solve SIS and BDD

Direct

- algebraic approach Arora and Ge with subexponential complexity when $\sigma \leq \sqrt{n}$, else fully exponential, mainly of asymptotic interest (higher complexity than others)

- combinatorial algorithms: BKW as basis [BKW03], resembles generalized birthday approach by Wagner, originally for solving LPN, can be analyzed => explicit complexity for different LWE instances, theoretical analysis and actual performance close, very memory expensive (often same order as time complexity)

3.2.2 BKW [BKW03]

The Blum, Kalai and Wasserman (BKW) algorithm was originially designed to solve the Learning Parity with Noise problem (LPN) [BKW03]. In Section 2.3.1 we pointed out that LPN is a subproblem of LWE and Albrecht *et al.* adapted BKW to LWE in [ACF+15]. The runtime and memory complexity of BKW is in $2^{O(n)}$ for an LWE instance with secret dimension n prime modulus $q \in \text{poly}(n)$. The number of samples m must be sufficiently large (in $O(n \log n)$).

BKW falls into the regime of dual attacks, that is, it solves LWE by finding a short vector $\mathbf{s}$ in the scaled dual lattice $\Lambda(\mathbf{A}^\intercal)^\perp$.

three stages [ACF+15]: sample reduction, hypothesis testing and back substitution

Sample Reduction. In the following, we present an outline of the main BKW algorithm. The steps in Algorithm 3 are inspired by the textual description in [GJS15] with minor adjustments in notation.

For the algorithm, we use the matrix notation of LWE as in Equation (2.9), i.e. $\mathbf{z} = \mathbf{A}^\intercal \mathbf{s} + \mathbf{e}$. BKW consists of a series of BKW steps that iteratively reduce the dimension of input matrix $\mathbf{A}$ by finding collisions of its column vectors in the currently examined block of b entries. We start from the last b entries of $\mathbf{A}^{(1)} = \mathbf{A}$. In every step i, we maintain a collision table $\mathbf{T}^{(i)}$ and loop over the columns $\mathbf{a}_k^{(i)}$ of $\mathbf{A}^{(i)}$ and distinguish between the following cases: (1) If $\mathbf{a}_k^{(i)}$ only has zero entries in the examined block, pass $\mathbf{a}_k^{(i)}$ and $z_k^{(i)}$ to the next step, (2) if no match of $\mathbf{a}_k^{(i)}$ or the negation of $\mathbf{a}_k^{(i)}$ can be found in the collision table, add $\mathbf{a}_k^{(i)}$ to the colission table, and (3) if a match $\mathbf{a}_l^{(i)}$ is found, compute $\mathbf{a}_l^{(i)} + \mathbf{a}_k^{(i)}$ or in the case of a negation match $\mathbf{a}_l^{(i)} - \mathbf{a}_k^{(i)}$ (in $\mathbb{Z}_q$) such that the last b nonzero entries cancel out. By exploiting the symmetry of $\mathbb{Z}_q$ in this way, in every step we obtain at most $(q^b - 1)/2$ columns with distinct coefficients in the current b entries. We also make note of "observed symbols" $z_j^{(i)}$ that represent the combination of two samples given their respective matching columns (see lines 20, 24 for more details).

In each BKW step, the number of columns (and samples) decreases by at least $(q^b - 1)/2$ (size of the colusion set) and the variance of the error distribution σ^2 increases by a factor of two. The algorithm terminates after $t = \lceil b/(n-d) \rceil$ steps returns a a set of observed symbols $\mathbf{z}^{(t)}$ and a corresponding reduced matrix $\mathbf{A}^{(t)}$ in which only the first d rows have nonzero entries. Parameter d should be set to 1, as in the original BKW algorithm, or 2 for the best performance [ACF+15].

Algorithm 3: BKW (Sample Reduction)

```
**function** BKW($\mathbf{A} \in \mathbb{Z}^{n\times m}, \mathbf{z} \in \mathbb{Z}^m, b \in \mathbb{Z}, d \in \mathbb{Z}$)
    $i = 1$
    $\mathbf{A}^{(i)} = \mathbf{A}$
    $\mathbf{z}^{(i)} = \mathbf{z}$
    **while** *the last $n - d$ coefficients of the columns of* $\mathbf{A}^{(i)}$ *are nonzero* **do**
        // BKW step
        $j = 1$
        $\mathbf{T}^{(i)} = []$                                        // Collision table
        **for** $k = 1, \dots, m^{(i)}$ **do**
            // $m^{(i)}$ is number of columns in $\mathbf{A}^{(i)}$
            **if** *last $(i \cdot b)$ coefficients of* $\mathbf{a}_k^{(i)}$ *are zero* **then**
                $\mathbf{a}_j^{(i+1)} = \mathbf{a}_k^{(i)}$
                $z_j^{(i+1)} = z_k$
                $j = j + 1$
            **else if** *no match for* $\mathbf{a}_k^{(i)}$ *in* **T then**
                $\mathbf{T} = \mathbf{T} + \left[\mathbf{a}_k^{(i)}\right]$                         // append to collision set
            **else if** *match* $\mathbf{a}_l^{(i)}$ *for* $\mathbf{a}_k^{(i)}$ *is found* **then**
                **if** $\mathbf{a}_l^{(i)}$ *matches* $\mathbf{a}_k^{(i)}$ *in the last $(i \cdot b)$ components* **then**
                    $\mathbf{a}_j^{(i+1)} = \mathbf{a}_k^{(i)} - \mathbf{a}_l^{(i)}$;          // last $i \cdot b$ coefficients of $\mathbf{a}_j^{(i+1)}$ are now zero
                    $z_j^{(i+1)} = z_k^{(i)} - z_l^{(i)} = y_j^{(i)} + e_j^{(i)}$, where $y_j^{(i)} = \left\langle \mathbf{s}, \mathbf{a}_j^{(i)} \right\rangle$ and $e_j^{(i)} = e_k^{(i)} - e_l^{(i)}$
                    $j = j + 1$
                **else if** *the negation of* $\mathbf{a}_l^{(i)}$ *in* $\mathbb{Z}_q^n$ *matches* $\mathbf{a}_k^{(i)}$ *in the last $(i \cdot b)$ components* **then**
                    $\mathbf{a}_j^{(i+1)} = \mathbf{a}_k^{(i)} + \mathbf{a}_l^{(i)}$
                    $z_j^{(i+1)} = z_k^{(i)} + z_l^{(i)} = y_j^{(i)} + e_j^{(i)}$, where $y_j^{(i)} = \left\langle \mathbf{s}, \mathbf{a}_j^{(i)} \right\rangle$ and $e_j^{(i)} = e_k^{(i)} + e_l^{(i)}$
                    $j = j + 1$
        $i = i + 1$
        // Calculate input for next BKW step
        $\mathbf{A}^{(i)} = (\mathbf{a}_1^{(i)} \cdots \mathbf{a}_{j-1}^{(i)})$
        $\mathbf{z}^{(i)} = (z_1^{(i)}, \dots, z_{j-1}^{(i)})$
    Return $(\mathbf{A}^{(i)}, \mathbf{z}^{(i)})$
```

The remaining part $\mathbf{s}'$ of the secret vector $\mathbf{s}$ is then guessed by means of hypothesis testing. After t steps the error term $\left(\mathbf{z}_j^{(t)} - \left\langle \mathbf{s}', \mathbf{a}_j^{(t)} \right\rangle\right)$ with $j \in [m']$ of the m' remaining observed symbols follows a Gaussian distribution χ with noise $\sigma'^2 = 2^t \cdot \sigma^2$ (see Lemma 1 in [ACF+15]). We can test the noise of the error term for all $\mathbf{s}'' \in \mathbb{Z}_q^d$ against the hypothesized noise σ'^2 by means of the log-likelihood ratio (for details we again refer to [ACF+15]) and are thus able to determine $\mathbf{s}'$ given sufficiently many samples m'.

Finally, we can apply back substitution to recover all elements of $\mathbf{s}$. We again apply a similar procedure as in Algorithm 3 to reduce a number of columns from the colission tables computed in the Sample Reduction step and obtain m' columns with $d + d'$ nonzero entries and their corresponding “observed symbols”. We then substitute the part of s that was recovered in the previous steps and recover the next part of $\mathbf{s}$ by hypothesis testing and repeat the process until we have found $\mathbf{s}$.

Theorem 1 (BKZ Complexity [ACF+15], Corollary 2)
Let $(\mathbf{a}_i, z_i)$ be samples following $\mathcal{A}_{\mathbf{s},\chi}$, set $a = \lfloor \log_2(1/(2\alpha)^2) \rceil$, $b = n/a$ and q a prime. Let d be a small constant $0 < d < \log_2(n)$. Assume α is such that $q^b = q^{n/a} = q^{n/\lfloor \log_2(1/(2\alpha)^2) \rceil}$ is superpolynomial in n. Then, given these parameters the cost of the BKW algorithm to solve Search-LWE is

$$\left(\frac{q^b-1}{2}\right) \cdot \left(\frac{a(a-1)}{2} \cdot (n+1)\right) + \left\lceil (\frac{q^b}{2} \right\rceil \cdot \left(\left\lceil (\frac{n}{d} \right\rceil + 1\right) \cdot d \cdot a + poly(n) \approx (a^2 n) \cdot \frac{q^b}{2} \tag{3.4}$$

operations in $\mathbb{Z}_q$. Furthermore,

$$a \cdot \left\lceil (\frac{q^b}{2} \right\rceil + poly(n) \text{ calls to } \mathcal{A}_{\mathbf{s},\chi} \text{ and storage of } \left(a \cdot \left\lceil (\frac{q^b}{2} \right\rceil \cdot n\right) \text{ elements in } \mathbb{Z}_q \text{ are needed.} \tag{3.5}$$

The first summand of Equation (3.4) roughly corresponds to the cost of creating the colission tables and the second summand is the cost of backsubstitution. For a more detailed cost analysis, see Theorem 2 in [ACF+15].

Coded-BKW [GJS15]

- change BKW step -> more column entries are removed, but additional noise - index set I, $\mathbf{x}_I$ is part of $\mathbf{x}$ with entries indexed by I - step i: I set of b positions to be removed, fix some q-ary linear $[N_i, b]$ code C_i with q^b codewords, find the closest codeword $\mathbf{c}_I \in C$ for every input vector $\mathbf{a}_I$ such that $\mathbf{a}_I = \mathbf{c}_I + \mathbf{e}_I$, where the error part $\mathbf{e}_I \in \mathbb{Z}_q^{N_i}$ is minimized by a decoding procedure.

Finally, we subtract two vectors and their correpsonding samples and pass the result to the next BKW step. Consider the inner product$\langle \mathbf{s}_I, \mathbf{a}_I \rangle = \langle \mathbf{s}_I, \mathbf{c}_I \rangle + \langle \mathbf{s}_I, \mathbf{e}_I \rangle$. In the subtraction, only the error part $\langle \mathbf{s}_I, \mathbf{e}_I \rangle$ remains.

3.2.3 Dual Attack [MR09]

"Gama and Nguyen [GN08b]: (in)feasibility of obtaining various Hermite factors natural distinguishing attack on LWE by finding one relatively short vector in associated lattice"

3.2.4 Decoding Attack [LP11]

combines lattice basis reduction followed by an enumeration algorithm (bounded-distance decoding with preprocessing?) => time/success tradeoff specifically for LWE, exploits structural properties of LWE on search version of LWE problem, approach preferable to distinguishing attack on decision LWE in [MR09; RS10], same or better advantage than distinguishing attack using lattice vectors of lower quality => runtime is smaller post-reduction: simple extension of Babai's "nearest-plane" algorithm [Bab85] => trade basis quality against decoding time related to Klein's (de)randomized algorithm [Kle00] for bounded-distance decoding

use entire reduced basis, post-reduction part is fully parallelizable

LLL reduction to input Lattice, integer combination of basis vectors close to target (like inner loop in reduction step of LLL), seek vector in lattice close to target, finds output that is in fundamental parallelipiped $\mathcal{P}(\mathbf{B})$ Section 2.2.1 => if error vector not in $\mathcal{P}(\mathbf{B})$, secret is not restored => basis quality has to be sufficiently good

Algorithm 4: Babai's Nearest Plane Algorithm [Bab85]

```
function NearestPlane($\mathbf{B} \in \mathbb{R}^{m \times n}, \mathbf{t} \in \mathbb{R}^m$)
    run $\delta$-LLL on basis $\mathbf{B}$ with $\delta = \frac{3}{4}$
    $\mathbf{b} = \mathbf{t}$
    for $i = n, \dots, 1$ do
        $c_i$ = round($\langle \mathbf{b}, \tilde{\mathbf{b}}_i \rangle / \langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle$) $\mathbf{b} = \mathbf{b} - c_i \mathbf{b}_i$
    output $\mathbf{t} - \mathbf{b}$
```

Output is a lattice vector $\mathbf{v} \in \Lambda(\mathbf{B})$ such that $||\mathbf{v} - \mathbf{t}|| \leq 2^{n/2} \mathrm{dist}(\mathbf{t}, \Lambda(\mathbf{B}))$

Goal: recover lattice vector relatively close to target vector Intuition: - project $\mathbf{t}$ to span($\mathbf{B}$) - from $i = n, \dots, 1$ find closest hyperplane $c_i \tilde{\mathbf{b}}_i + \mathrm{span}(\mathbf{b}_1, \dots, \mathbf{b}_i)$ to the projection, subtract $c_i \mathbf{b}_i$ from the projection and continue - output vector is $\sum_{i=1}^{n} c_i \mathbf{b}_i$ for every basis vector $\mathbf{b}_i$ find c_i such that distance between target and hyperplane spanned by $\mathbf{b}_1, ..., \mathbf{b}_{i-1}$ and shifted by $c_i \tilde{\mathbf{b}}_i$ is minimal , subtract $c_i \mathbf{b}_i$ from target vector and continue for $i = n, \dots, 1$. After the last iteration $\sum_{i=1}^{n} c_i \mathbf{b}_i$ is returned.

Application to LWE: $\mathbf{t} = \mathbf{A}^\intercal \mathbf{s} + \mathbf{e}$ => we get $\mathbf{v}$ where $\mathbf{t} - \mathbf{v} = \mathbf{e}$ is in fundamental parallelipiped of Gram-Schmidt basis

Generalized version by [LP11]: Problem: in reduced basis last Gram-Schmidt vectors of B short, first long => long and skinny parallelipiped, Gaussian e unlikely to be in it => incorrect answer from NearestPlane

=> generalized version admitting time/success tradeoff recurse on some $d_i \geq 1$ distinct planes in ith

Instead of choosing only the nearest plane in each iteration step, Algorithm 5 selects a variable amount d_k of distinct planes in each step. As a consequence, the fundamental parallelipiped of the Gram-Schmidt basis is stretched in the direction of $\tilde{\mathbf{b}}_k$. The values of $\mathbf{d}$ should be chosen such that the covered area is approximately the same in each direction (i.e. by maximizing $\min_i(d_i \cdot ||\tilde{\mathbf{b}}_i||)$).

Algorithm 5: Generalized Nearest Plane Algorithm [LP11]

```
function GeneralizedNearestPlane($\mathbf{B} \in \mathbb{R}^{m\times k}, \mathbf{t} \in \mathbb{R}^m, \mathbf{d} \in (\mathbb{Z}^+)^k$)
    if $k = 0$ then
        Return $\mathbf{0}$
    else
        Compute projection $\mathbf{v}$ of $\mathbf{t}$ onto span($\mathbf{B}$)
        Compute the $d_k$ distinct integers $c_1, \ldots, c_{d_k}$ closest to $\langle \mathbf{v}, \tilde{\mathbf{b}}_k\rangle / \langle \tilde{\mathbf{b}}_k, \tilde{\mathbf{b}}_k\rangle)$
        Return $\bigcup_{i\in\{1,\ldots,d_k\}}(c_i \cdot \mathbf{b}_k +$
          GeneralizedNearestPlane($\{\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}\}, (d_1, \ldots, d_{k-1}), \mathbf{v} - c_i \cdot \mathbf{b}_k$))
```

In particular this implies that the d_k are larger for larger k as the Gram-Schmidt vectors have a smaller length. Compared to Algorithm 4 the runtime increases by a factor $\prod_{i\in\{1,\ldots,d_k\}} d_i$, however, the recursion step can be fully parallelized.

It should be evident that a lower quality of the reduced input basis can be compensated for by increasing the values of $\mathbf{d}$. Hence we can adjust the input parameters for the lattice reduction and Algorithm 5 to minimize the runtime given a fixed required success probability.

3.2.5 Primal-uSVP [ADPS16; BG14]

BKZ: reduce lattice basis using SVP oracle in smaller dimension b, known that number of calls to oracle polynomial - enumeration algorithm as oracle: in super-exponential time - sieve algorithms as oracle: exponential time but so far slower in practice for accesible dimensions $b \approx 130$

primal attack: construct unique-SVP instance from LWE instance LWE instance $(\mathbf{A}, \mathbf{z} = \mathbf{A}^\intercal\mathbf{s} + \mathbf{e})$ construct lattice

$$(3.6)\quad \Lambda = \left\{\mathbf{x} \in \mathbb{Z}^{m+n+1} \mid (\mathbf{A}^\intercal | -\mathbf{I}_m| - \mathbf{b})\mathbf{x} = \mathbf{0} \mod q\right\}$$

lattice has dimension $d = m + n + 1$, volume q^m and unique-SVP solution $\mathbf{v} = (\mathbf{s}, \mathbf{e}, 1)$

success condition: - geometric series assumption known to be optimistic from attacker's point of view => finds basis with Gram-Schmidt norms $\|mat\tilde{h}bfb_i\| = \delta^{d-2i-1} \cdot \mathrm{Vol}(\Lambda)^{1/d}$ and $\delta = ((\pi b)^{1/b} \cdot b/2\pi e)^{1/2(b-1)}$ unique short vector $\mathbf{v}$ is detected if projection of $\mathbf{v}$ onto span of last b Gram-Schmidt vectors is shorter than $mat\tilde{h}bfb_{d-b}$, norm of projection is expected to be $\gamma\sqrt{b}$ => attack successful iff $\gamma\sqrt{b} \leq \delta^{d-2i-1} \cdot q^{m/d}$

LWE as inhomogeneous-SIS (ISIS)

As in Section 3.2.4, we view the $\mathrm{LWE}_{n,q,m,\chi}$ instance $(\mathbf{A}, \mathbf{z})$ as a BDD instance in the q-ary lattice $\Lambda(\mathbf{A}^\intercal) = \{\mathbf{y} \mid \exists \mathbf{x} \in \mathbb{Z}_q^n : \mathbf{y} = \mathbf{A}^\intercal\mathbf{x} \mod q\}$ Section 2.3.1 generated by rows of LWE instance. The target vector is $\mathbf{z}$.

Recall the γ-uSVP problem. Given a lattice Λ where $\lambda_2(\Lambda) > \gamma\lambda_1(\Lambda)$, we are asked to find shortest nonzero vector in Λ. In the primal attack, instead of directly solving BDD, we reduce BDD to uSVP, i.e., we reduce a BDD instance to a γ-uSVP instance. By solving γ-uSVP, we obtain a solution to BDD. To do this we apply Kannan's embedding technique [Kan87]. Intuitively, Kannan's embedding creates a lattice with uSVP structure. We know that $\mathbf{A}^\top\mathbf{s} \mod q$ is the closest vector to the target $\mathbf{z} = \mathbf{A}^\top\mathbf{s}+\mathbf{e}^\top \mod q$ in $\Lambda(\mathbf{A}^\top)$. We now add a linearly indpenendent basis vector $(\mathbf{z}, \mu)$ and append a zero coefficient to each basis vector of the original lattice (i.e. the rows of $\mathbf{A}$). Thereby, we ensure that the new lattice contains the vector $[-\mathbf{e}, -\mu]^\top$ as $[\mathbf{A} \mid \mathbf{0}]^\top\mathbf{s} - 1 \cdot [\mathbf{z}^\top, \mu] = [-\mathbf{e}, -\mu]^\top$.

More formally, let $\mathbf{B}$ be a basis of $\Lambda(\mathbf{A}^\top)$ and an embedding factor $\mu = \mathrm{dist}(\mathbf{z}, \Lambda(\mathbf{A}^\top)) = \|\mathbf{z} - \mathbf{s}\|$ where $\mathbf{s}$ is the secret vector of the LWE instance. A relatively close approximation of μ can be guessed in polynomial time (see [LM09] for more details). We now embed $\Lambda(\mathbf{A}^\top)$ into $\Lambda(\mathbf{B}')$ with γ-uSVP structure as follows:

$$\text{(3.7)}\quad \mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{z} \\ \mathbf{0}^\top & \mu \end{pmatrix}$$

If $\gamma \geq 1$ and $\mu < \frac{\lambda_1(\Lambda(\mathbf{B})}{2\gamma}$ (or equivalently, $(\Lambda(\mathbf{A}^\top), \mathbf{z})$ a $\mathrm{BDD}_{1/(2\gamma)}$-instance), then $\Lambda(\mathbf{B}')$ contains a γ-unique shortest vector $\mathbf{z}' = [(\mathbf{A}^\top\mathbf{s} - \mathbf{z})^\top, -\mu]^\top = [-\mathbf{e}^\top, -\mu]^\top$. The statement can be proven by showing by contradiction that all vectors $\mathbf{v} \in \Lambda\mathbf{B}'$ that are independent of $\mathbf{z}'$ satisfy $\|\mathbf{v}\| \geq \lambda_1\Lambda(\mathbf{B})/\sqrt{2} > \sqrt{2}\gamma\mu = \gamma\|\mathbf{z}'\|$ (see Section 4 of [LM09] for more details). Note that the reduction can be done in polynomial time (Theorem 4.1 in [LM09]).Hence, from $\mathbf{z}'$ we can recover the error vector $\mathbf{e}$ and thereby the secret vector $\mathbf{s} = \mathbf{z} - \mathbf{e} \mod q$.

A solution to γ-uSVP can be found by reducing it to κ-HSVP where $\gamma = \kappa^2$ [APS15]. Various algorithms, in particular, lattice reduction algorithms, exist to solve κ-HSVP. If we are able to solve a linear number of κ-HSVP instances that correspond to a κ^2-approximate SVP instance, we can construct a solution the latter (see Definition 2.2.1, see Section 1.2.21 in [Lov87] for more details). Consider any lattice with uSVP structure. In exactly one direction, that is, in the direction of its unique shortest vector, the lattice has vectors that are significantly smaller than in other directions. A lattice reduction algorithm that yields a sufficiently good output basis quality, therefore, must return some small vector in the desired direction. Let $\mathbf{v}$ be a solution to SVP^2_κ, i.e. $\|\mathbf{v}\| \leq \kappa^2\lambda_1(\Lambda)$. All other vectors $\mathbf{w} \in \Lambda$ that are not multiples of a shortest vector have length $\|\mathbf{w}\| \geq \lambda_2(\Lambda) > \kappa^2\lambda_1(\Lambda)$. Thus, we obtain a solution to γ-uSVP and, as shown above, we can reconstruct the secret vector to solve LWE.

3.2.6 Meet-in-the-Middle [APS15]

3.2.7 Arora-Ge [AG11]

3.3 Algorithms for Solving SIS

Recall that the $\mathrm{SIS}_{n,q,m,\beta}$ problem asks to find a short vector $\mathbf{s} \in \mathbb{Z}_q^m$ of norm $\|\mathbf{s}\| \leq \beta$ such that such that $\mathbf{A} \cdot \mathbf{s} = 0 \bmod q$ for some uniformly distributed matrix $\mathbf{A}^{n \times m}$. Solving SIS is equivalent to finding a short vector in the dual lattice $\Lambda(\mathbf{A}^\top)^\perp = \{\mathbf{y} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{y} = \mathbf{0} \mod q\}$.

3.3.1 Lattice Reduction [MR09; RS10]

MR variant [MR09]

Our first approach to solving SIS follows quite naturally. Given $\mathbf{A}$, we can efficiently compute the basis $\mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}$ of the dual lattice $\Lambda(\mathbf{A}^\top)^\perp$ in polynomial time using Gauss-Jordan elimination or other more modern algorithms.

We can then apply a lattice reduction algorithm and obtain a basis with root Hermite factor δ. The first basis $\mathbf{b}_1$ vector of the reduced basis has length $\|\mathbf{b}_1\| = \delta^m \det(\Lambda(\mathbf{A}^\top)^\perp)^{1/m}$. We can see that δ depends on the subdimension m which we want to be ideal in order to minimize the cost of the lattice reduction by relaxing δ.

We further assume that $\det(\Lambda(\mathbf{A}^\top)^\perp) = \mathrm{Vol}(\Lambda(\mathbf{A}^\top)^\perp) = q^n$ (see [MR09] for more details). For q prime and m much larger than n we have that the rank of $\mathbf{A}$ is n as the rows of $\mathbf{A}$ are with high probability linearly independent. The nullity or the dimension of the kernel of $\mathbf{A}$ is $m - n$ and as a result the dual lattice has q^{m-n} points in $\mathbb{Z}_q^m$. Consider the fundamental domain $D = \mathcal{P}(\Lambda(\mathbf{A}^\top)^\perp)$ and the fact that $\Lambda(\mathbf{A}^\top)^\perp + (D \mod q) = \mathbb{R}^m/q\mathbb{R}^m$ is a partition. The volume of $\mathbb{R}^m/q\mathbb{R}^m$ is given by $q^m = |\Lambda(\mathbf{A}^\top)^\perp||D \mod q|$ and thus

$$\det(\Lambda(\mathbf{A}^\top)^\perp) = |D \mod q| = \frac{q^m}{|\Lambda(\mathbf{A}^\top)^\perp|} = \frac{q^m}{q^{m-n}} = q^n. \tag{3.8}$$

We now have our first equation

$$\|\mathbf{b}_1\| = \delta^m q^{\frac{n}{m}}, \tag{3.9}$$

which becomes minimal for $m = \sqrt{n \log q / \log \delta}$.

Theorem 2 (Optimal subdimension *m* [MR09])
Given a q-ary scaled dual lattice $\Lambda(\mathbf{A}^\top)^\perp$ defined by a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$ with m sufficiently larger than n and a prime q. Then a lattice reduction algorithm yields an optimal output if performed in subdimension

$$m' = \sqrt{\frac{n \log q}{\log \delta}}. \tag{3.10}$$

Higher dimension increase the complexity of the reduction algorithms and lower dimensions may cause a lack of sufficiently short lattice vectors [MR09]. In contexts in which Equation (3.9) does not hold, we may still choose m as in Equation (3.10) heuristically. Removing columns from $\mathbf{A}$ does not greatly impact our results since we can just set the corresponding components of the secret vector s to zero. We reformulate Equation (3.9) a bit:

$$\|\mathbf{b}_1\| = \delta^m q^{\frac{n}{m}} \iff \log\beta = m \log \delta + \frac{n \log q}{m} \tag{3.11}$$

$$\iff \log \delta = \frac{\log \beta}{m} - \frac{n \log q}{m^2} \tag{3.12}$$

We continue by plugging Equation (3.10) into Equation (3.12):

(3.13)
$$\log\delta = \frac{\log\beta}{\sqrt{\frac{n\log q}{\log\delta}}} - \frac{n\log q}{\left(\sqrt{\frac{n\log q}{\log\delta}}\right)^2} \iff \log\delta = \frac{\log\beta}{\sqrt{\frac{n\log q}{\log\delta}}} - \log\delta$$

(3.14)
$$\iff 2\log\delta = \frac{\log\beta}{\sqrt{\frac{n\log q}{\log\delta}}}$$

(3.15)
$$\iff \log\delta = \frac{\log^2\beta}{4n\log q}$$

Theorem 3 (Optimal subdimension *m* [MR09])
Given a q-ary scaled dual lattice $\Lambda(\mathbf{A}^\intercal)^\perp$ defined by a matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$ with m sufficiently larger than n and a prime q. Then a lattice reduction algorithm performed in its optimal subdimension achieves a log root Hermite factor of

(3.16) $$\log\delta = \frac{\log^2\beta}{4n\log q}.$$

To estimate the cost of the lattice reduction for SIS, we call a function from the *Estimator* to find the required block size k such that BKZ achieves root Hermite factor δ and apply a cost model with the optimal subdimension m' and block size k.

Note that for LWE we have α, q as input parameters instead of a bound. We can convert α to a required bound $\beta = \frac{1}{\alpha}\sqrt{\ln(\frac{1}{\epsilon})/\pi}$ such that the success probability of solving an LWE instance is given by ϵ (Corollary 2 in [APS15]). The *Estimator* uses a rinse and repeat strategy to find the best tradeoff between runtime and success probability.

RS variant [RS10]

A similar approach is described in [RS10]. The optimal subdimension and required root Hermite factor are given by a slightly different expression. Apart from that the attack works as described in Section 3.3.1.

Theorem 4 (Optimal subdimension *m* ([RS10], Conjecture 2))
For every $n \geq 128$, constant $c \geq 2$, $q \geq n^c$, $m = \Omega(n\log_2(q))$ and $\beta < q$, the best known approach to solve SIS with parameters (n, m, q, β) involves solving δ-HSVP in dimension $m' = \min(x : q^{2n/x} \leq \beta)$ with $\delta = \sqrt[m]{d}\beta/q^{n/m'}$.

We reformulate the expression for m'

(3.17)
$$q^{2n/m'} \leq \beta$$

(3.18)
$$\frac{2n}{m' \log(q)} \leq \beta$$

(3.19)
$$m' \geq \frac{2n \log(q)}{\log(\beta)}$$

and obtain $m' = \left\lceil \frac{2n \log(q)}{\log(\beta)} \right\rceil$. If $m' > m$, we take $m' = m$.

The root Hermite factor δ must be larger than 1 for the reduction to be tractable. From $\delta = \sqrt{d}\beta / q^{n/m'} \geq 1$ it follows that we need $m' \geq n \log_2(q)/\log_2(\beta)$.

3.3.2 Combinatorial Attack [MR09]

Micciancio and Regev also describe a combinatorial method for solving SIS [MR09].

Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and the dual lattice $\Lambda(\mathbf{A}^\intercal)^\perp$, we want to find a lattice vector $\mathbf{v} \in \Lambda(\mathbf{A}^\intercal)^\perp$ with coefficients bounded by b. Expressed differently, we want to find $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = 0 \mod q$ and $\|\mathbf{v}_i\| \leq b$ for all $i \in [m]$.

We begin by dividing the columns of $\mathbf{A}$ into 2^k for some k. Each set contains $m/2^k$ column vectors. We now compute all linear combinations $c_1\mathbf{b}_1 + \cdots + c_{m/2^k}\mathbf{b}_{m/2^k}$, where $\mathbf{b}_i$ denote the indexed column vectors in each set, such that $|c_i| \leq b$ and obtain 2^k new sets $\mathbf{A}_j^{(k)}$ of $L = (2b+1)^{m/2^k}$ vectors, $j \in [2^k]$. Remember that each c_i represents a coeffient of the lattice vector. By means of these new sets we satisfy the shorteness criteria of the output vector.

We then continue iteratively for $i = k$ to $i = 0$ as follows. In each step we merge pairs of two sets $\mathbf{A}_j^{(i)}, \mathbf{A}_{j+1}^{(i)}$. If a vector $\mathbf{x}$ in first set of the pair can be combined with each vector $\mathbf{y}$ in second set such that the first $\log_q L$ components in $\mathbf{x} \pm \mathbf{y}$ are zero, we put the result in combined set in $\mathbf{A}_j^{(i-1)}$. The size of the combined sets is at most L as we consider a part of the vectors with $\log_q L$ components that can take at most $q^{\log_q L} = L$ different values. We start from $j = 0$ and increment j by 2 after each merge for $j < 2^i$. After the merge now 2^{i-1} sets.

We choose k such that

(3.20)
$$n \approx (k+1)\log_q L = (k+1)\log_q (2b+1)^{m/2^k} \iff \frac{2^k}{k+1} \approx \frac{m \log(2\beta + 1)}{n \log(q)}$$

After k steps the first $k \log_q L$ entries of the columns in the result set are cancelled out. We expect that of the remaining $\approx \log_q L$ entries in result set that we should find at least one zero vector as at there are most L different vectors (see above). The zero vector represents the linear combination with entries bounded by b and we can easily reconstruct the short lattice vector $\mathbf{v}$.

To find an optimal k, we iterate over k starting from $k = 1$ and calculate diff as follows:

(3.21)

$$\frac{2^k}{k+1} \approx \frac{m \log(2\beta + 1)}{n \log(q)}$$

(3.22)

$$\text{diff} = \text{abs}\left(\frac{2^k}{k+1} - \frac{m \log(2\beta + 1)}{n \log(q)}\right).$$

When diff does not decrease for 10 iteration steps, we stop and take the current k.

We make a conservative estimate of the cost by estimating the number of operations needed to create the initial sets as the overall cost is dominated by this parameter. Each of the 2^k lists contains L vectors. The cost for any operation on a list element is at least $\log_2(q) \cdot n$. Hence, the total cost is $2^k \cdot L \cdot \log_2(q) \cdot n$.

4 Lattice Parameter Estimation Tool

The main goal of the thesis is the creation a tool that combines the algorithm estimates for LWE and SIS that we introduced above and can be configured without much knowledge of the workings of the underlying algorithms. To our knowledge there has not been a tool that includes the function of generically searching for secure parameters for both LWE and SIS instances as well as for ring and module variants. Some schemes (e.g. commitment schemes, see Section 5.1) depend on the statistical security of either LWE or SIS. We hence include classes respectively to find parameters satisfying this criteria. Furthermore, we provide a set of utility classes and methods for most commonly used distributions and norms. A configuration class allows for a substantial customization of the estimation process. The tool can either estimate the bit security level of fixed parameter sets or generically search for parameter sets that satisfy a certain bit security level.

4.1 Supported Distributions

4.1.1 Gaussian Distribution

In some applications we receive a Gaussian distribution as input but require a bound in some norm in order to estimate the hardness of an SIS instance. Hence, we need to transform a Gaussian with parameter $s = \sqrt{2\pi}\sigma$ into a bound β given some security parameter sec. Note that a n-dimensional Gaussian $D_{\mathbf{Z}^n,s}$ can be sampled by combining samples from n independent one-dimensional Gaussians $D_{\mathbf{Z},s}$ [GJS15].

For a Gaussian distribution and a random variable X with $X \sim D_{\mathbf{Z},s}$, the following holds:

$$\Pr\left[|X| \geq \beta\right] \leq 2e^{-\pi \|\mathbf{v}\|^2/s^2} \tag{4.1}$$

We demand $2e^{-\pi\beta^2/s^2} \approx 2^{-sec}$ with $\beta = \|\mathbf{v}\|$ and obtain

$$\begin{aligned} 2e^{-\pi\beta^2/s^2} &\approx 2^{-sec} \\ -\pi\frac{\beta^2}{s^2} &\approx (-sec - 1)\ln(2) \\ \beta &\approx s\sqrt{\frac{(sec+1)\ln(2)}{\pi}}. \end{aligned}$$

Theorem 5 (Gaussian to Bound)
Given a Gaussian distribution $D_{\mathbf{Z}^n,s}$ with width parameter $s = \sqrt{2\pi}\sigma$ and a security parameter sec, we can compute a bound β such that a sample $\mathbf{v}$ drawn from $D_{\mathbf{Z}^n,s}$ satisfies Pr $[\|\mathbf{v}\| \geq \beta] \leq 2^{-sec}$ as follows:

$$\beta \approx s\sqrt{\frac{(sec+1)\ln(2)}{\pi}}. \tag{4.2}$$

If the Gaussian is specified componentwise we consider β as an instance of `norm.Lp` with `p` $= \infty$, else as an instance of `norm.Lp` with `p` $= 2$.

4.1.2 Uniform Distribution

4.2 Norms and Bounds

Let $\mathcal{R}_q$ be a ring as defined in [BDL+18] and $f \in \mathcal{R}_q$ with $f = \sum_i f_i X^i$. We define the following norms [BDL+18]:

$$\ell_1 : \|f\|\|_1 = \sum_i |f_i| \tag{4.3}$$

$$\ell_2 : \|f\|\|_2 = \left(\sum_i |f_i|^2\right)^{\frac{1}{2}} \tag{4.4}$$

$$\ell_\infty : \|f\|\|_\infty = \max_i |f_i| \tag{4.5}$$

Then the following inequations hold [BDL+18]:

$$\|f\|_1 \leq \sqrt{n}\|f\|_2 \tag{4.6}$$

$$\|f\|_1 \leq n\|f\|_\infty \tag{4.7}$$

$$\|f\|_2 \leq \sqrt{n}\|f\|_\infty \quad (\text{since } \sqrt{n}\|f\|_2 \leq n\|f\|_\infty) \tag{4.8}$$

$$\|f\|_\infty \leq \|f\|_1 \tag{4.9}$$

Let O_K be the ring of integers of a number field $K = \mathbb{Q}(\theta)$, where θ is an algebraic number and σ denote the canonical embedding as defined in [DPSZ12]. Then, for $x, y \in O_K$ it holds the following inequations hold (we assume that C_m in [DPSZ12] is 1) [DPSZ12].

(4.10)

$$\|f\|_\infty \leq \|\sigma(f)\|_\infty$$

(4.11)

$$\|\sigma(f)\|_\infty \leq \|f\|_1$$

From the above inequations, we obtain the following norm transformations to ℓ_p-norms:

- From Equation (4.6), it follows that $\|f\|_1 \leq \sqrt{n}\|f\|_2$ and from Equation (4.7), $\|f\|_1 \leq n\|f\|_\infty$.

- From Equation (4.8) and Equation (4.9), it follows that $\|f\|_2 \leq \sqrt{n}\|f\|_1$ and from Equation (4.8), $\|f\|_2 \leq \sqrt{n}\|f\|_\infty$.
- From Equation (4.9), it follows that $\|f\|_\infty \leq \|f\|_1$ and from Equation (4.6) and Equation (4.9), $\|f\|_\infty \leq \sqrt{n}\|f\|_2$.
- From Equation (4.11), it follows that $\|\sigma(f)\|_\infty \leq \|f\|_1$, from Equation (4.6) and Equation (4.11), $\|\sigma(f)\|_\infty \leq \sqrt{n}\|f\|_2$, and from Equation (4.7) and Equation (4.11), $\|\sigma(f)\|_\infty \leq n\|f\|_\infty$.

Likewise, we get the following transformations to the C_∞-norm:

- From Equation (4.7) and Equation (4.10), it follows that $\|f\|_1 \leq n\|\sigma(f)\|_\infty$.
- From Equation (4.8) and Equation (4.10), it follows that $\|f\|_2 \leq \sqrt{n}\|\sigma(f)\|_\infty$.
- From Equation (4.10), it follows that $\|f\|_\infty \leq \|\sigma(f)\|_\infty$.

Let f be defined as above and let $g \in \mathcal{R}_q$ where $g = \sum_i \overline{g}_i X^i$ where $g_i \in [-(q-1)/2, (q-1)/2]$ and $\overline{g}_i = g_i \mod q$ as in [BDL+18]. Then, we can define the following inequations for multiplication according to [BDL+18]:

- If $\|f\|_\infty \leq \beta, \|g\|_1 \leq \gamma$ then $\|f \cdot g\|_\infty \leq \beta \cdot \gamma$.
- If $\|f\|_2 \leq \beta, \|g\|_2 \leq \gamma$ then $\|f \cdot g\|_\infty \leq \beta \cdot \gamma$.

Let $x, y \in \mathcal{O}_K$. Again, we assume that $C_m = 1$. Then, the following inequation holds according to [DPSZ12]:

(4.12)
$$\|x \cdot y\|_\infty \leq C_m \cdot n^2 \cdot \|x\|_\infty \cdot \|y\|_\infty$$

(4.13)
$$\|\sigma(x \cdot y)\|_\infty \leq \|\sigma(x)\|_\infty \cdot \|\sigma(y)\|_\infty.$$

4.3 Problems

We now present the problem classes in `lattice_parameter_estimation/problem` (see Figure 4.1).

`LWE` and `SIS` inherit from the base class `BaseProblem` respectively. All instances provide a method `get_estimate_algorithms()` that returns a list of algorithm instances that can be executed by the function `estimate()`. Furthermore, any instance of `BaseProblem` can be compared to a bit security level (for more details, we refer the reader to the documentation). The `LWE` class is initialized by the secret dimension n, a modulus q, the number of samples m, a `secret_distribution` and a `error_distribution`. Both `secret_distribution` and `error_distribution` must be instances of the class `distributions.Distribution`. Instead of `secret_distribution` and `error_distribution`, a bound of type `norm.BaseNorm` must be set for `SIS`. Note that both distributions.Uniform and distributions.Gaussian are instances of `norm.BaseNorm` and can thus be used as a bound. We compute the bound for a given distribution instance as described in Section 4.1.

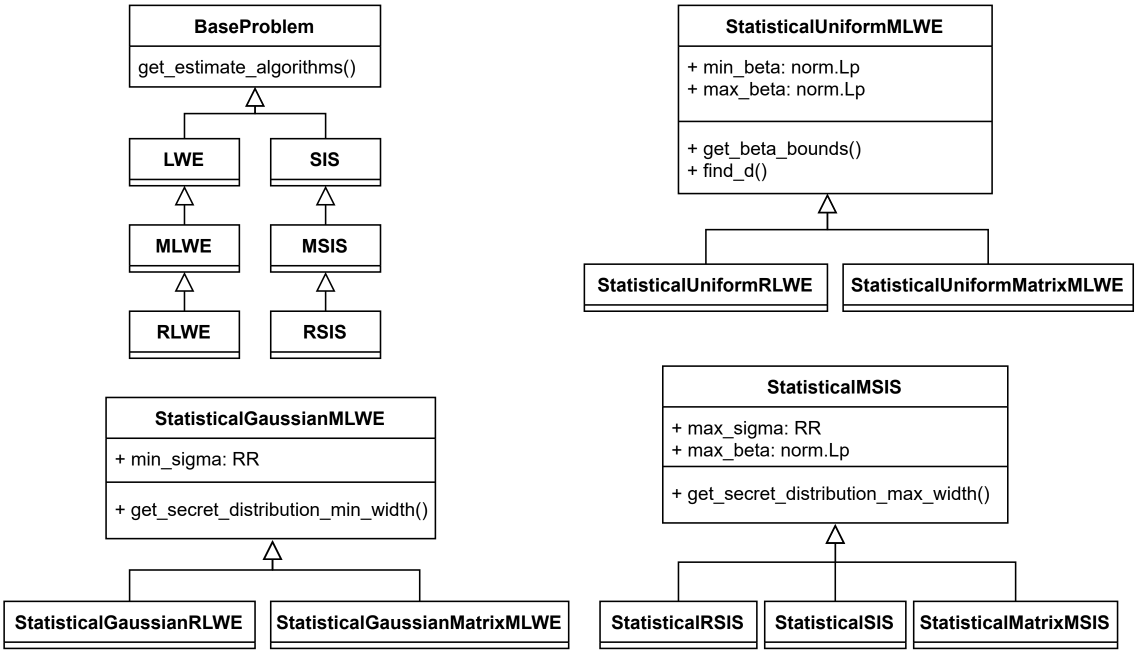

Figure 4.1: Problem Classes

For ring and module variants `RLWE`, `RSIS` and `MLWE`, `MSIS` respectively n denotes the degree of the polynomial of the underlying ring $\mathcal{R}_q$. The module variants `MLWE` and `MSIS` take an addition parameter d for the rank of the module.

While there exist special cases where the ring structure of problem instances can be exploited in an attack on LWE or SIS, in general, the hardness of ring and Module variants is estimated by interpreting the coefficients of elements of $\mathcal{R}_q$ as vectors in $\mathbb{Z}_q^n$ [ACD+18]. We thus reduce ring and Module instances as follows when calling `get_estimate_algorithms()` on the ring and module variant of `LWE` and `SIS`:

- $\mathrm{RLWE}_{n,q,m,\chi} \longrightarrow \mathrm{LWE}_{n,q,m\cdot n,\chi}$
- $\mathrm{MLWE}_{n,d,q,m,\chi} \longrightarrow \mathrm{LWE}_{n\cdot d,q,m\cdot n,\chi}$
- $\mathrm{RSIS}_{n,q,m,\beta} \longrightarrow \mathrm{SIS}_{n,q,m\cdot n,\beta}$
- $\mathrm{MSIS}_{n,d,q,m,\beta} \longrightarrow \mathrm{SIS}_{n\cdot d,q,m\cdot n,\beta}$

`StatisticalGaussianMLWE`. For LWE, we define a statistically secure variant over a Gaussian distribution and over a uniform distribution. `StatisticalGaussianMLWE` follows Corollary 7.5 and Theorem 7.4 in [LPR13]. To avoid confusion we first specify the mapping for differing usage of parameters in [LPR13] as compared to this work Table 4.1 and obtain the following theorem:

Theorem 6
Statistically Secure MLWE Over a Gaussian Distribution [LPR13] Let $\mathcal{R}$ be the ring of integers in the m'th cyclotomic number field K of degree n, and $q \geq 2$ an integer. For positive integers $m \leq m + d \leq poly(n)$, let $\mathbf{A} = [\mathbf{I}_{[m]} \mid \bar{\mathbf{A}}] \in (\mathcal{R}_q)^{[m]\times[m+d]}$, where $\mathbf{I}_{[m]} \in (\mathcal{R}_q)^{[m]\times[m]}$ is the identity matrix and $\bar{\mathbf{A}} \in (\mathcal{R}_q)^{[m]\times[d]}$ is uniformly random. Then with probability $1 - 2^{-\Omega(n)}$ *over*

Parameters in [LPR13]	Use Here	Represents
l	$m + d$	width of matrix **A**
k	m	height of matrix **A**

Table 4.1: Parameter Mapping from [LPR13]

Parameters in [BDL+18]	Use Here	Represents
k	$m + d$	width of matrix $[\mathbf{I}_n\ \mathbf{A}']$
n	m	height of matrix $[\mathbf{I}_n\ \mathbf{A}']$
d	d_2	variable
N	n	degree of Ring polynomial

Table 4.2: Parameter Mapping from [BDL+18]

the choice of $\bar{\mathbf{A}}$*, the distribution of* $\mathbf{A}\mathbf{x} \in (\mathcal{R}_q)^{[m]}$ *where each coordinate of* $\mathbf{x} \in (\mathcal{R}_q)^{[m+d]}$ *is chosen from a discrete Gaussian distribution of parameter* $s > 2n \cdot q^{m/(m+d)+2/(n(m+d))}$ *over* $\mathcal{R}$*, satisfies that the probability of each of the* q^{nm} *possible outcomes is in the interval* $(1 \pm 2^{-\Omega(n)})q^{-n}$ *(and in particular is within statistical distance* $2^{-\Omega(n)}$ *of the uniform distribution over* $(\mathcal{R}_q)^{[m]}$*).*

If a security parameter is passed and $sec > n$, we raise an exepction. The resulting minimal standard deviation is stored in the instance variable `min_sigma` and the corresponding distribution can be obtained by calling `get_secret_distribution_min_width()` on the class instance.

`StatisticalUniformMLWE`. The authors of [BDL+18] describe statistically secure MLWE instances over a Uniform distribution with invertible elements. The samples $(\mathbf{A}', h_{\mathbf{A}'}(y))$ of the resulting MLWE instance are within statistical distance 2^{-sec} of $(\mathbf{A}', \mathbf{u})$ for uniformly distributed $\mathbf{u}$.

In Table 4.2 we specify the mapping of parameters used in [BDL+18] and obtain the following theorem:

Theorem 7 (Statistically Secure MLWE Over a Uniform Distribution [BDL+18])
Let $1 < d_2 < n$ *be a power of 2. If* q *is a prime congruent to* $2d_2 + 1 \pmod{4d_2}$ *and*

$$q^{m/(m+d)} \cdot 2^{2sec/((m+d)\cdot n)} \leq 2\beta < \frac{1}{\sqrt{d_2}} \cdot q^{1/d_2} \tag{4.14}$$

then any (all-powerful) algorithm $\mathcal{A}$ *has advantage at most* 2^{-sec} *in solving* $DKS^{\infty}_{m,m+d,\beta}$*, where* DKS^{∞} *is the decisional knapsack problem in* ℓ_∞*-norm.*

Hence, we have:

$$\beta_{min} = \frac{q^{m/(m+d)} \cdot 2^{2sec/((m+d)\cdot n)}}{2} \tag{4.15}$$

$$\beta_{max} = \frac{1}{2\sqrt{d_2}} \cdot q^{1/d_2} - 1 \tag{4.16}$$

Parameters in [DOTT21]	Use Here	Represents
m'	$m + d$	width of matrix $\hat{\mathbf{A}}_1$
m	m	height of matrix $\hat{\mathbf{A}}_1$
B	B	norm-bound of secret
s	s	Gaussian width (not stddev)
N	n	degree of polynomial

Table 4.3: Parameter Mapping from [DOTT21]

The variable d_2 can be passed as an argument. If it is not passed, we try to find d_2 by iterating over all powers of 2 that are smaller than n. The resulting bounds are converted to ℓ_∞ and stored in the instance variables `min_beta` and `max_beta`. We also provide an instance method `get_beta_bounds()` to obtain a tuple of both.

For both statistically secure MLWE variants, we include the corresponding ring versions `StatisticalGaussianRLWE` and `StatisticalUniformRLWE` for $d = 1$ and matrix versions `StatisticalGaussianMatrixMLWE` and `StatisticalUniformMatrixMLWE` for which the width and height of the matrix $\mathbf{A}$ in [LPR13] can be passed instead of m and d.

`StatisticalMSIS`. We can find parameters for a statistically secure MSIS instance by following Section 4.1 of [DOTT21]. More specifically, we ask to find a MLWE instance where the probability that non zero elements $\mathbf{r}$ in the Euclidean ball $B_m(0, 2B)$ satisfy $\hat{\mathbf{A}}_1 \cdot \mathbf{r} = \mathbf{0}$ is smaller than 2^{-sec}.

We give a mapping of the parameters in [DOTT21] in Table 4.3

The number of elements in $B_{m+d}(0, 2B)$ can be estimated from above as $|B_{m+d}(0, 2B)| \ll (2\pi e/((m+d)n))^{(m+d)n/2} \cdot (2B)^{(m+d)n}$. The scheme is statistically binding if the probability that non zero elements in $B_{m+d}(0, 2B)$ of radius $2B$ in $\mathcal{R}_q^{m+d}$ map to $\mathbf{0}$ in $\mathcal{R}_q^m$ is negligible. Hence, it must hold that $|B_{m+d}(0, 2B)|/q^{mn} \leq 2^{-sec}$ and we get:

$$\left(\sqrt{\frac{2\pi e}{(m+d)\cdot n}} \cdot 2B\right)^{(m+d)\cdot n} \leq 2^{-sec} \cdot q^{m \cdot n} \tag{4.17}$$

$$B \leq 2^{\frac{-sec}{(m+d)\cdot n} - 1} \cdot q^{\frac{m}{m+d}} \cdot \sqrt{\frac{(m+d)\cdot n}{2\pi e}} \tag{4.18}$$

We convert the bound B to a Gaussian over ℓ_2-norm by following the procedure described in **??**:

$$s \approx x\sqrt{\frac{\pi}{(sec+1)\ln(2)}} \tag{4.19}$$

The resulting parameters B and s can be accessed by the instance variables `max_sigma` and `max_beta` or by calling `get_secret_distribution_max_width()` on the class instance.

As for statistically secure MLWE we again include a matrix version `StatisticalMatrixMSIS` and a ring `StatisticalRSIS` by setting $d = 1$. In addition, the proof also applies to the base SIS variant and hence we include `StatisticalSIS`. Here the height of the matrix n becomes the rank of the modulus in the MSIS instance, i.e. $d = n$, and the degree of the polynomial is 1.

4.4 Configuration

We now describe estimate configuration options in our tool that can be set in a helper class `algorithms.Configuration`.

Cost Models. Attacks that use BKZ require a cost model to estimate the number of CPU cycles in the SVP subroutine. Default cost models are shown in Table 3.2. We distinguish between classical, quantum, sieving and enumeration SVP estimates and each of these categories can be chosen by setting the respective parameter to `True` or `True`. Note that at least one of classical and quantum and of sieving and enumeration has to be in the selection. We included an option of taking the most conservative estimate for each category combination for a more efficient parameter search or estimation. Furthermore, we assigned an ordinal priority value to each cost model which enables us to first run cost models that yield a lower cost and thus terminate the estimation process earlier for an insecure parameter set. Priority values and default cost models are derived from Figure 4.2 Custom cost models can be specified and the default cost models deactivated. The number of BKZ rounds can be configured by passing a `lambda` with parameters `beta, d` where `beta` is the block size and `d` the lattice dimension. By default we chose the conservative "Core"-SVP model [ADPS16] in which the polynomial factor of the runtime complexity of BKZ is ignored. In addition, we provide a model `BKZ_SVP_REPEAT_8d` for $8 \cdot n$ BKZ rounds (see Line 17).

4.4.1 Algorithms

- include plots - write about pro/cons - maybe add section of cost comparison for the algorithms of the previous section

4.4.2 Generic Search

The main functionality of our tool is encapsulated in the function `param_search.generic_search()`. The high-level idea of the search is presented in Algorithm 6. We begin with an initial parameter set. We then create a list of problem instances generated by a `parameter_problem` function and estimate the cost of all instances in the list. If the list contains multiple SIS instances or multiple LWE instances, we attempt to reduce the instances to the easiest problem instance respectively. Once the `estimate` function finds an instance that is insecure (i.e. the estimated attack cost in clock cycles is smaller than 2^{sec}), it terminates the estimation procedure and returns an insecure result. We then use the `next_parameters` function to generate a list L of (multiple) new parameter sets from our current parameter set and sort each of the new parameter sets into an ordered list (duplicates are

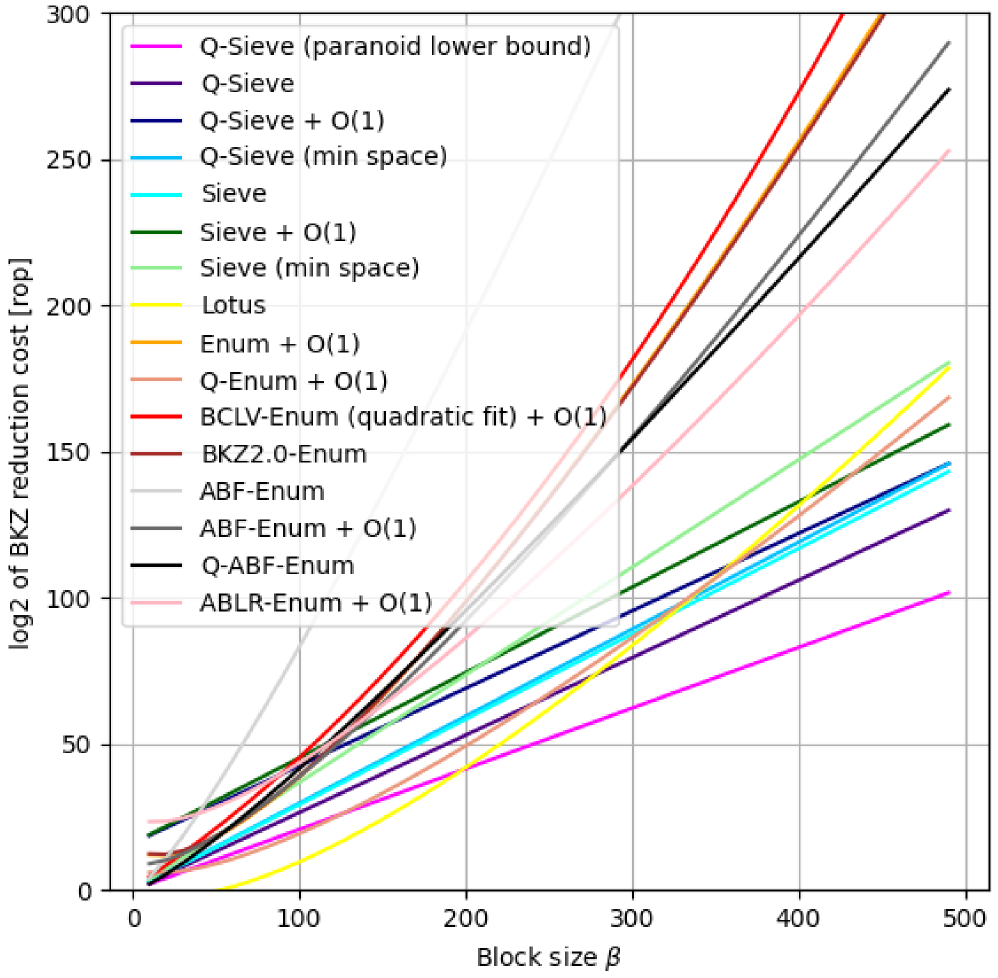

Figure 4.2: Cost Models

not accepted). The order is defined by a `parameter_cost` function. In the next step we retrieve the parameter set with the lowest cost from L and repeat the procedure until the cost estimation step returns a secure result. The result includes the estimates for all cost models and algorithms.

The `estimate` function can be configured to run in parallel in the configuration which may speed up the search, in particular if many cost models need to be tested (e.g. with configuration setting `conservative=False` and long running algorithms like `ARORA_GB`, `CODED_BKW` and `PRIMAL_DECODE` are used. We recommend to include `PRIMAL_USVP` for LWE instances and `LATTICE_REDUCTION` for SIS instances to make full use of early termination since the estimate algorithms for these attacks have a short runtime and yield relatively low cost estimates. The runtime and performance of the algorithms used in the tool are shown in Figure 4.3, Figure 4.4 and Figure 4.5.

Algorithm 6: Generic Search

```
function generic_search(sec, initial_params, next_parameters, parameter_cost,
  problem_instance
  )
L = OrderedList(initial_params)
while $L \neq \emptyset$ do
    current_params = L.pop()
    instances = parameter_problem(current_params)
    result = estimate(instances, sec)
    if result is secure then
        Return result
    else
        next_param_sets = next_parameters(current_params)
        forall param_set in next_param_sets do
            sort param_set into L according to parameter_cost function
```

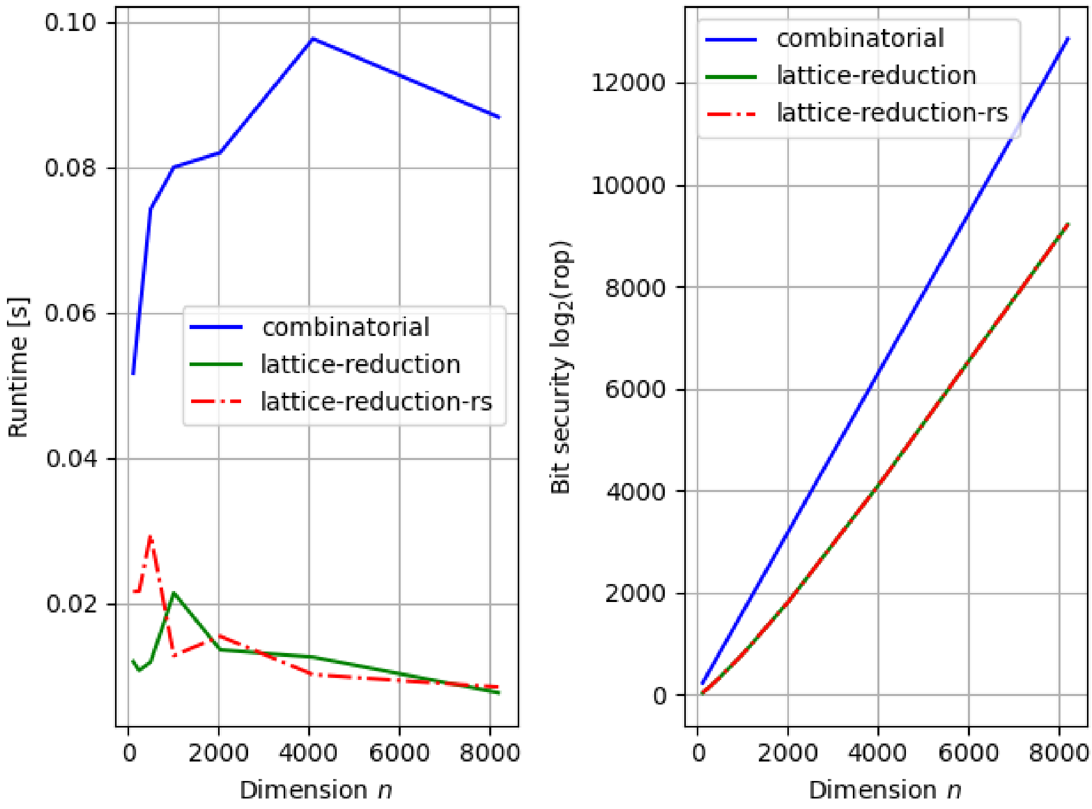

Figure 4.3: SIS instance with $\sigma = 2.828, \ m = n^2, \ 2^{2n} < q < 2^{2n+1}$

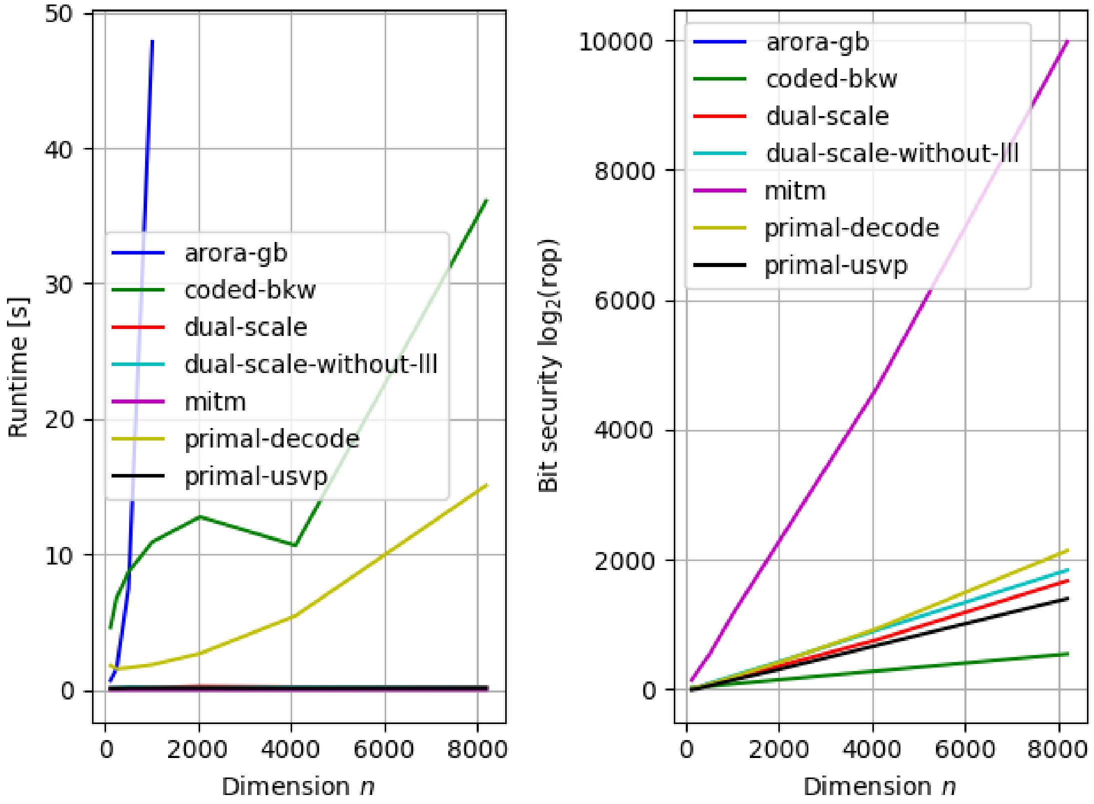

Figure 4.4: LWE instance with $\sigma = 0.125,\ m = \infty,\ 2^n < q < 2^{n+1}$

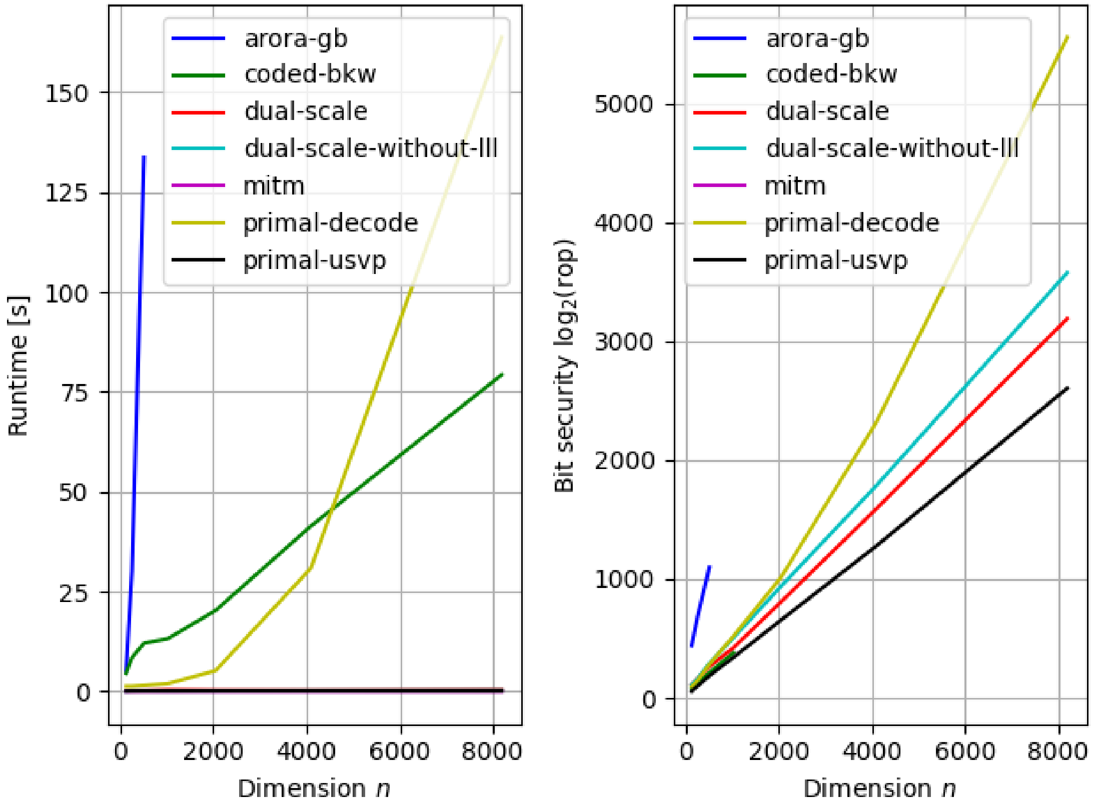

Figure 4.5: LWE instance with $\sigma = 2.828, \ m = \infty, \ 2^n < q < 2^{n+1}$

5 Usage Examples

5.1 Two Problem Search

two problems lwe sis, what they are, how to solve it by the tool

5.2 TODO: find other schemes to apply

6 Conclusion

Bibliography

[ABB10] S. Agrawal, D. Boneh, X. Boyen. “Efficient Lattice (H)IBE in the Standard Model”. In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*. Ed. by H. Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 553–572. DOI: 10.1007/978-3-642-13190-5_28. URL: https://doi.org/10.1007/978-3-642-13190-5%5C_28 (cit. on pp. 17, 21).

[ABF+20] M. R. Albrecht, S. Bai, P.-A. Fouque, P. Kirchner, D. Stehlé, W. Wen. “Faster Enumeration-Based Lattice Reduction: Root Hermite Factor $k^{1/(2k)}$ Time $k^{k/8+o(k)}$”. In: *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*. Ed. by D. Micciancio, T. Ristenpart. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 186–212. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-030-56880-1_7%7D (cit. on pp. 29, 31).

[ABLR21] M. R. Albrecht, S. Bai, J. Li, J. Rowell. “Lattice Reduction with Approximate Enumeration Oracles - Practical Algorithms and Concrete Performance”. In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*. Ed. by T. Malkin, C. Peikert. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 732–759. DOI: 10.1007/978-3-030-84245-1_25. URL: https://doi.org/10.1007/978-3-030-84245-1%5C_25 (cit. on pp. 26, 28, 31).

[ACD+18] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, T. Wunderer. “Estimate All the {LWE, NTRU} Schemes!” In: *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*. Ed. by D. Catalano, R. D. Prisco. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 351–367. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-319-98113-0_19%7D (cit. on pp. 31, 46).

[ACF+15] M. R. Albrecht, C. Cid, J.-C. Faugère, R. Fitzpatrick, L. Perret. “On the complexity of the BKW algorithm on LWE”. In: *Des. Codes Cryptogr.* 74.2 (2015), pp. 325–354. URL: https://doi.org/10.1007/s10623-013-9864-x (cit. on pp. 32, 34).

[ACPS09] B. Applebaum, D. Cash, C. Peikert, A. Sahai. “Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems”. In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. Ed. by S. Halevi. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35. URL: https://doi.org/10.1007/978-3-642-03356-8%5C_35 (cit. on p. 21).

[AD97] M. Ajtai, C. Dwork. “A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence”. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*. STOC ’97. El Paso, Texas, USA: Association for Computing Machinery, 1997, pp. 284–293. ISBN: 0897918886. URL: https://doi.org/10.1145/258533.258604 (cit. on pp. 17, 21).

[ADH+19] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, M. Stevens. “The General Sieve Kernel and New Records in Lattice Reduction”. In: *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*. Ed. by Y. Ishai, V. Rijmen. Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 717–746. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-030-17656-3_25%7D (cit. on pp. 28, 30).

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe. “Post-quantum Key Exchange - A New Hope”. In: *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*. Ed. by T. Holz, S. Savage. USENIX Association, 2016, pp. 327–343. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim (cit. on pp. 29, 31, 36, 49).

[AFG13] M. R. Albrecht, R. Fitzpatrick, F. Göpfert. “On the Efficacy of Solving LWE by Reduction to Unique-SVP”. In: *Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers*. Ed. by H.-S. Lee, D.-G. Han. Vol. 8565. Lecture Notes in Computer Science. Springer, 2013, pp. 293–310. DOI: 10.1007/978-3-319-12160-4_18. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-319-12160-4_18%7D (cit. on p. 19).

[AG11] S. Arora, R. Ge. “New Algorithms for Learning in Presence of Errors”. In: *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*. Ed. by L. Aceto, M. Henzinger, J. Sgall. Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 403–415. DOI: 10.1007/978-3-642-22006-7_34. URL: https://doi.org/10.1007/978-3-642-22006-7%5C_34 (cit. on p. 37).

[AGPS20] M. R. Albrecht, V. Gheorghiu, E. W. Postlethwaite, J. M. Schanck. “Estimating Quantum Speedups for Lattice Sieves”. In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*. Ed. by S. Moriai, H. Wang. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 583–613. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-030-64834-3_20%7D (cit. on p. 31).

[AGV09] A. Akavia, S. Goldwasser, V. Vaikuntanathan. “Simultaneous Hardcore Bits and Cryptography against Memory Attacks”. In: *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*. Ed. by O. Reingold. Vol. 5444. Lecture Notes in Computer Science. Springer, 2009, pp. 474–495. DOI: 10.1007/978-3-642-00457-5_28. URL: https://doi.org/10.1007/978-3-642-00457-5%5C_28 (cit. on p. 21).

[Ajt96] M. Ajtai. “Generating Hard Instances of Lattice Problems (Extended Abstract)”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. Ed. by G. L. Miller. ACM, 1996, pp. 99–108. URL: https://doi.org/10.1145/237814.237838 (cit. on pp. 17, 23).

[Ajt98] M. Ajtai. “The Shortest Vector Problem in L_2 is *NP*-hard for Randomized Reductions (Extended Abstract)”. In: *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*. Ed. by J. S. Vitter. ACM, 1998, pp. 10–19. DOI: 10.1145/276698.276705. URL: https://doi.org/10.1145/276698.276705 (cit. on p. 28).

[ANS18] Y. Aono, P. Q. Nguyen, Y. Shen. “Quantum Lattice Enumeration and Tweaking Discrete Pruning”. In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*. Ed. by T. Peyrin, S. D. Galbraith. Vol. 11272. Lecture Notes in Computer Science. Springer, 2018, pp. 405–434. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-030-03326-2_14%7D (cit. on p. 29).

[APS15] M. R. Albrecht, R. Player, S. Scott. “On the concrete hardness of Learning with Errors”. In: *J. Math. Cryptol.* 9.3 (2015), pp. 169–203. URL: http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml (cit. on pp. 15, 28, 37, 39).

[AWHT16] Y. Aono, Y. Wang, T. Hayashi, T. Takagi. “Improved Progressive BKZ Algorithms and Their Precise Cost Estimation by Sharp Simulator”. In: *Proceedings, Part I, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9665*. Berlin, Heidelberg: Springer-Verlag, 2016, pp. 789–819. ISBN: 9783662498897. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-662-49890-3_30%7D (cit. on p. 27).

[Bab85] L. Babai. “On Lovász’ Lattice Reduction and the Nearest Lattice Point Problem (Shortened Version)”. In: *STACS 85, 2nd Symposium of Theoretical Aspects of Computer Science, Saarbrücken, Germany, January 3-5, 1985, Proceedings*. Ed. by K. Mehlhorn. Vol. 182. Lecture Notes in Computer Science. Springer, 1985, pp. 13–20. DOI: 10.1007/BFb0023990. URL: https://doi.org/10.1007/BFb0023990 (cit. on p. 35).

[BCLV17] D. J. Bernstein, C. Chuengsatiansup, T. Lange, C. van Vredendaal. “NTRU Prime: Reducing Attack Surface at Low Cost”. In: *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*. Ed. by C. Adams, J. Camenisch. Vol. 10719. Lecture Notes in Computer Science. Springer, 2017, pp. 235–260. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-319-72565-9_12%7D (cit. on p. 31).

[BDGL16] A. Becker, L. Ducas, N. Gama, T. Laarhoven. “New directions in nearest neighbor searching with applications to lattice sieving”. In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. Ed. by R. Krauthgamer. SIAM, 2016, pp. 10–24. URL: https://doi.org/10.1137/1.9781611974331.ch2 (cit. on pp. 30, 31).

[BDL+18] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, C. Peikert. "More Efficient Commitments from Structured Lattice Assumptions". In: *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*. Ed. by D. Catalano, R. D. Prisco. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 368–385. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-319-98113-0_20%7D (cit. on pp. 44, 45, 47).

[BG14] S. Bai, S. D. Galbraith. "Lattice Decoding Attacks on Binary LWE". In: *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings*. Ed. by W. Susilo, Y. Mu. Vol. 8544. Lecture Notes in Computer Science. Springer, 2014, pp. 322–337. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-319-08344-5_21%7D (cit. on p. 36).

[BGJ15] A. Becker, N. Gama, A. Joux. "Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search". In: *IACR Cryptol. ePrint Arch.* 2015 (2015), p. 522. URL: http://eprint.iacr.org/2015/522 (cit. on p. 30).

[BKW03] A. Blum, A. Kalai, H. Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *J. ACM* 50.4 (2003), pp. 506–519. URL: https://doi.org/10.1145/792538.792543 (cit. on pp. 22, 32).

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. "Classical hardness of learning with errors". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. Ed. by D. Boneh, T. Roughgarden, J. Feigenbaum. ACM, 2013, pp. 575–584. URL: https://doi.org/10.1145/2488608.2488680 (cit. on p. 15).

[BLS16] S. Bai, T. Laarhoven, D. Stehlé. "Tuple lattice sieving". In: *IACR Cryptol. ePrint Arch.* (2016), p. 713. URL: http://eprint.iacr.org/2016/713 (cit. on p. 30).

[Bra12] Z. Brakerski. "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP". In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Ed. by R. Safavi-Naini, R. Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 868–886. DOI: 10.1007/978-3-642-32009-5_50. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-32009-5_50%7D (cit. on p. 21).

[BV11] Z. Brakerski, V. Vaikuntanathan. "Efficient Fully Homomorphic Encryption from (Standard) LWE". In: *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*. Ed. by R. Ostrovsky. IEEE Computer Society, 2011, pp. 97–106. URL: https://doi.org/10.1109/FOCS.2011.12 (cit. on p. 21).

[Che13] Y. Chen. "Réduction de réseau et sécurité concrete du chiffrement completement homomorphe". PhD thesis. Paris 7, 2013 (cit. on pp. 28, 29, 31).

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert. "Bonsai Trees, or How to Delegate a Lattice Basis". In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*. Ed. by H. Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 523–552. DOI: 10.1007/978-3-642-13190-5_27. URL: https://doi.org/10.1007/978-3-642-13190-5%5C_27 (cit. on pp. 17, 21, 23).

[CN11] Y. Chen, P. Q. Nguyen. “BKZ 2.0: Better Lattice Security Estimates”. In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. Ed. by D. H. Lee, X. Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 1–20. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-25385-0_1%7D (cit. on pp. 27, 31).

[DGK+10] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, V. Vaikuntanathan. “Public-Key Encryption Schemes with Auxiliary Inputs”. In: *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*. Ed. by D. Micciancio. Vol. 5978. Lecture Notes in Computer Science. Springer, 2010, pp. 361–381. DOI: 10.1007/978-3-642-11799-2_22. URL: https://doi.org/10.1007/978-3-642-11799-2%5C_22 (cit. on p. 21).

[DOTT21] I. Damgård, C. Orlandi, A. Takahashi, M. Tibouchi. “Two-Round n-out-of-n and Multi-signatures and Trapdoor Commitment from Lattices”. In: *Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I*. Ed. by J. A. Garay. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, pp. 99–130. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-030-75245-3_5%7D (cit. on p. 48).

[DPSZ12] I. Damgård, V. Pastro, N. P. Smart, S. Zakarias. “Multiparty Computation from Somewhat Homomorphic Encryption”. In: *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*. Ed. by R. Safavi-Naini, R. Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 643–662. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-32009-5_38%7D (cit. on pp. 44, 45).

[DSW21] L. Ducas, M. Stevens, W. P. J. van Woerden. “Advanced Lattice Sieving on GPUs, with Tensor Cores”. In: *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*. Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 249–279. DOI: 10.1007/978-3-030-77886-6_9. URL: https://doi.org/10.1007/978-3-030-77886-6%5C_9 (cit. on p. 28).

[Gen09] C. Gentry. “A Fully Homomorphic Encryption Scheme”. AAI3382729. PhD thesis. Stanford, CA, USA, 2009. ISBN: 9781109444506 (cit. on pp. 17, 21).

[GGH96] O. Goldreich, S. Goldwasser, S. Halevi. “Collision-Free Hashing from Lattice Problems”. In: *Electron. Colloquium Comput. Complex.* 3.42 (1996). URL: https://eccc.weizmann.ac.il/eccc-reports/1996/TR96-042/index.html (cit. on p. 23).

[GGH97] O. Goldreich, S. Goldwasser, S. Halevi. “Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem”. In: *Advances in Cryptology - CRYPTO ’97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*. Ed. by B. S. K. Jr. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 105–111. DOI: 10.1007/BFb0052230. URL: https://doi.org/10.1007/BFb0052230 (cit. on p. 21).

[GJS15] Q. Guo, T. Johansson, P. Stankovski. "Coded-BKW: Solving LWE Using Lattice Codes". In: *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*. Ed. by R. Gennaro, M. Robshaw. Vol. 9215. Lecture Notes in Computer Science. Springer, 2015, pp. 23–42. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-662-47989-6_2%7D (cit. on pp. 22, 32, 34, 43).

[GKPV10] S. Goldwasser, Y. T. Kalai, C. Peikert, V. Vaikuntanathan. "Robustness of the Learning with Errors Assumption". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. Ed. by A. C.-C. Yao. Tsinghua University Press, 2010, pp. 230–240. URL: http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/19.html (cit. on p. 21).

[GN08a] N. Gama, P. Q. Nguyen. "Finding short lattice vectors within mordell's inequality". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. Ed. by C. Dwork. ACM, 2008, pp. 207–216. URL: https://doi.org/10.1145/1374376.1374408 (cit. on p. 25).

[GN08b] N. Gama, P. Q. Nguyen. "Predicting Lattice Reduction". In: *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*. Ed. by N. P. Smart. Vol. 4965. Lecture Notes in Computer Science. Springer, 2008, pp. 31–51. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-540-78967-3_3%7D (cit. on pp. 28, 34).

[GNR10] N. Gama, P. Q. Nguyen, O. Regev. "Lattice Enumeration Using Extreme Pruning". In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*. Ed. by H. Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 257–278. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-13190-5_13%7D (cit. on p. 27).

[GPV08] C. Gentry, C. Peikert, V. Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. Ed. by C. Dwork. ACM, 2008, pp. 197–206. URL: https://doi.org/10.1145/1374376.1374407 (cit. on pp. 17, 21, 23).

[GSW13] C. Gentry, A. Sahai, B. Waters. "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based". In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. Ed. by R. Canetti, J. A. Garay. Vol. 8042. Lecture Notes in Computer Science. Springer, 2013, pp. 75–92. DOI: 10.1007/978-3-642-40041-4_5. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-40041-4_5%7D (cit. on p. 21).

[HK17] G. Herold, E. Kirshanova. "Improved Algorithms for the Approximate k-List Problem in Euclidean Norm". In: *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*. Ed. by S. Fehr. Vol. 10174.

Lecture Notes in Computer Science. Springer, 2017, pp. 16–40. DOI: 10.1007/978-3-662-54365-8_2. URL: https://doi.org/10.1007/978-3-662-54365-8%5C_2 (cit. on p. 30).

[HPS11] G. Hanrot, X. Pujol, D. Stehlé. "Analyzing Blockwise Lattice Algorithms Using Dynamical Systems". In: *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*. Ed. by P. Rogaway. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 447–464. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-22792-9_25%7D (cit. on pp. 27, 28).

[HPS98] J. Hoffstein, J. Pipher, J. H. Silverman. "NTRU: A ring-based public key cryptosystem". In: *Algorithmic Number Theory*. Ed. by J. P. Buhler. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 267–288. ISBN: 978-3-540-69113-6 (cit. on p. 17).

[HS07] G. Hanrot, D. Stehlé. "Improved Analysis of Kannan's Shortest Lattice Vector Algorithm". In: *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*. Ed. by A. Menezes. Vol. 4622. Lecture Notes in Computer Science. Springer, 2007, pp. 170–186. DOI: 10.1007/978-3-540-74143-5_10. URL: https://doi.org/10.1007/978-3-540-74143-5%5C_10 (cit. on p. 29).

[Kan83] R. Kannan. "Improved Algorithms for Integer Programming and Related Lattice Problems". In: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*. Ed. by D. S. Johnson, R. Fagin, M. L. Fredman, D. Harel, R. M. Karp, N. A. Lynch, C. H. Papadimitriou, R. L. Rivest, W. L. Ruzzo, J. I. Seiferas. ACM, 1983, pp. 193–206. DOI: 10.1145/800061.808749. URL: https://doi.org/10.1145/800061.808749 (cit. on p. 29).

[Kan87] R. Kannan. "Minkowski's Convex Body Theorem and Integer Programming". In: *Math. Oper. Res.* 12.3 (1987), pp. 415–440. DOI: 10.1287/moor.12.3.415. URL: https://doi.org/10.1287/moor.12.3.415 (cit. on p. 36).

[Kho05] S. Khot. "Hardness of approximating the shortest vector problem in lattices". In: *J. ACM* 52.5 (2005), pp. 789–808. DOI: 10.1145/1089023.1089027. URL: https://doi.org/10.1145/1089023.1089027 (cit. on p. 28).

[Kle00] P. N. Klein. "Finding the closest lattice vector when it's unusually close". In: *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA*. Ed. by D. B. Shmoys. ACM/SIAM, 2000, pp. 937–941. URL: http://dl.acm.org/citation.cfm?id=338219.338661 (cit. on p. 35).

[KTX07] A. Kawachi, K. Tanaka, K. Xagawa. "Multi-bit Cryptosystems Based on Lattice Problems". In: *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings*. Ed. by T. Okamoto, X. Wang. Vol. 4450. Lecture Notes in Computer Science. Springer, 2007, pp. 315–329. DOI: 10.1007/978-3-540-71677-8_21. URL: https://doi.org/10.1007/978-3-540-71677-8%5C_21 (cit. on pp. 21, 23).

[Laa16] T. Laarhoven. "Search problems in cryptography: from fingerprinting to lattice sieving". English. Proefschrift. PhD thesis. Mathematics and Computer Science, Feb. 2016. ISBN: 978-90-386-4021-1 (cit. on pp. 29, 31).

[LLL82] A. Lenstra, H. Lenstra, L. Lovász. "Factoring Polynomials with Rational Coefficients". In: *Mathematische Annalen* 261 (Dec. 1982). URL: %5Curl%7Bhttps://doi.org/10.1007/BF01457454%7D (cit. on pp. 26–28).

[LM09] V. Lyubashevsky, D. Micciancio. "On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem". In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. Ed. by S. Halevi. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 577–594. DOI: 10.1007/978-3-642-03356-8_34. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-03356-8%5C_34%7D (cit. on pp. 20, 21, 37).

[LN20] J. Li, P. Q. Nguyen. "A Complete Analysis of the BKZ Lattice Reduction Algorithm". In: *IACR Cryptol. ePrint Arch.* (2020), p. 1237. URL: https://eprint.iacr.org/2020/1237 (cit. on p. 29).

[Lov87] L. Lovasz. *An Algorithmic Theory of Numbers, Graphs and Convexity*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1987. ISBN: 9780898712032. URL: https://books.google.com/books?id=sJ3mBHTU55QC (cit. on p. 37).

[LP11] R. Lindner, C. Peikert. "Better Key Sizes (and Attacks) for LWE-Based Encryption". In: *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*. Ed. by A. Kiayias. Vol. 6558. Lecture Notes in Computer Science. Springer, 2011, pp. 319–339. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-19074-2_21%7D (cit. on pp. 26, 35, 36).

[LPR13] V. Lyubashevsky, C. Peikert, O. Regev. "A Toolkit for Ring-LWE Cryptography". In: *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*. Ed. by T. Johansson, P. Q. Nguyen. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 35–54. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-38348-9_3%7D (cit. on pp. 46–48).

[LS15] A. Langlois, D. Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 75.3 (2015), pp. 565–599. URL: https://doi.org/10.1007/s10623-014-9938-4 (cit. on p. 23).

[Lyu08] V. Lyubashevsky. "Lattice-Based Identification Schemes Secure Under Active Attacks". In: *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*. Ed. by R. Cramer. Vol. 4939. Lecture Notes in Computer Science. Springer, 2008, pp. 162–179. DOI: 10.1007/978-3-540-78440-1_10. URL: https://doi.org/10.1007/978-3-540-78440-1%5C_10 (cit. on p. 23).

[MP13] D. Micciancio, C. Peikert. "Hardness of SIS and LWE with Small Parameters". In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*. Ed. by R. Canetti, J. A. Garay. Vol. 8042. Lecture Notes in Computer Science. Springer, 2013, pp. 21–39. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-642-40041-4_2%7D (cit. on p. 15).

[MR04] D. Micciancio, O. Regev. “Worst-Case to Average-Case Reductions Based on Gaussian Measures”. In: *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*. IEEE Computer Society, 2004, pp. 372–381. URL: https://doi.org/10.1109/FOCS.2004.72 (cit. on p. 23).

[MR09] D. Micciancio, O. Regev. “Lattice-based Cryptography”. In: *Post-Quantum Cryptography*. Ed. by D. J. Bernstein, J. Buchmann, E. Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 147–191. ISBN: 978-3-540-88702-7. URL: https://doi.org/10.1007/978-3-540-88702-7_5 (cit. on pp. 34, 35, 38–40).

[MV03] D. Micciancio, S. P. Vadhan. “Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More”. In: *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*. Ed. by D. Boneh. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 282–298. DOI: 10.1007/978-3-540-45146-4_17. URL: https://doi.org/10.1007/978-3-540-45146-4%5C_17 (cit. on p. 23).

[MV10] D. Micciancio, P. Voulgaris. “Faster Exponential Time Algorithms for the Shortest Vector Problem”. In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. Ed. by M. Charikar. SIAM, 2010, pp. 1468–1480. URL: https://doi.org/10.1137/1.9781611973075.119 (cit. on p. 30).

[MW16] D. Micciancio, M. Walter. “Practical, Predictable Lattice Basis Reduction”. In: *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*. Ed. by M. Fischlin, J.-S. Coron. Vol. 9665. Lecture Notes in Computer Science. Springer, 2016, pp. 820–849. URL: %5Curl%7Bhttps://doi.org/10.1007/978-3-662-49890-3_31%7D (cit. on p. 29).

[NS05] P. Q. Nguên, D. Stehlé. “Floating-Point LLL Revisited”. In: *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*. EUROCRYPT’05. Aarhus, Denmark: Springer-Verlag, 2005, pp. 215–233. ISBN: 3540259104. URL: %5Curl%7Bhttps://doi.org/10.1007/11426639_13%7D (cit. on p. 26).

[NV08] P. Q. Nguyen, T. Vidick. “Sieve algorithms for the shortest vector problem are practical”. In: *J. Math. Cryptol.* 2.2 (2008), pp. 181–207. DOI: 10.1515/JMC.2008.009. URL: https://doi.org/10.1515/JMC.2008.009 (cit. on p. 30).

[Pei09] C. Peikert. “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. Ed. by M. Mitzenmacher. ACM, 2009, pp. 333–342. URL: https://doi.org/10.1145/1536414.1536461 (cit. on pp. 17, 21, 22).

[PHAM17] L. T. Phong, T. Hayashi, Y. Aono, S. Moriai. “Lotus”. In: *Technical report, National Institute of Standards and Technology*. 2017. URL: %5Curl%7Bhttps://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions%7D (cit. on p. 31).

[PVW08] C. Peikert, V. Vaikuntanathan, B. Waters. “A Framework for Efficient and Composable Oblivious Transfer”. In: *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*. Ed. by D. A. Wagner. Vol. 5157. Lecture Notes in Computer Science. Springer, 2008, pp. 554–571. DOI: 10.1007/978-3-540-85174-5_31. URL: https://doi.org/10.1007/978-3-540-85174-5%5C_31 (cit. on p. 21).

[PW08] C. Peikert, B. Waters. “Lossy trapdoor functions and their applications”. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. Ed. by C. Dwork. ACM, 2008, pp. 187–196. DOI: 10.1145/1374376.1374406. URL: https://doi.org/10.1145/1374376.1374406 (cit. on pp. 17, 21).

[Reg03] O. Regev. “New lattice based cryptographic constructions”. In: *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*. Ed. by L. L. Larmore, M. X. Goemans. ACM, 2003, pp. 407–416. DOI: 10.1145/780542.780603. URL: https://doi.org/10.1145/780542.780603 (cit. on pp. 17, 21).

[Reg04] O. Regev. *Lecture notes in Lattices in Computer Science*. Fall 2004. URL: %5Curl%7Bhttps://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/lll.pdf%7D (cit. on p. 26).

[Reg05] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*. Ed. by H. N. Gabow, R. Fagin. ACM, 2005, pp. 84–93. URL: https://doi.org/10.1145/1060590.1060603 (cit. on pp. 15, 17, 21, 22).

[Reg09] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *J. ACM* 56.6 (2009), 34:1–34:40. URL: http://doi.acm.org/10.1145/1568318.1568324 (cit. on p. 21).

[Reg10] O. Regev. “The Learning with Errors Problem (Invited Survey)”. In: *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*. IEEE Computer Society, 2010, pp. 191–204. URL: https://doi.org/10.1109/CCC.2010.26 (cit. on p. 21).

[RS10] M. Rückert, M. Schneider. “Estimating the Security of Lattice-based Cryptosystems”. In: *IACR Cryptol. ePrint Arch.* 2010 (2010), p. 137. URL: http://eprint.iacr.org/2010/137 (cit. on pp. 35, 38, 39).

[SAL+17] N. P. Smart, M. R. Albrecht, Y. Lindell, E. Orsini, V. Osheter, K. Paterson, G. Peer. “Lima”. In: *Technical report, National Institute of Standards and Technology*. 2017. URL: %5Curl%7Bhttps://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions%7D (cit. on p. 31).

[Sch03] C.-P. Schnorr. “Lattice Reduction by Random Sampling and Birthday Methods”. In: *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*. Ed. by H. Alt, M. Habib. Vol. 2607. Lecture Notes in Computer Science. Springer, 2003, pp. 145–156. DOI: 10.1007/3-540-36494-3_14. URL: https://doi.org/10.1007/3-540-36494-3%5C_14 (cit. on p. 25).

[SE91] C.-P. Schnorr, M. Euchner. “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems”. In: *Fundamentals of Computation Theory, 8th International Symposium, FCT ’91, Gosen, Germany, September 9-13, 1991, Proceedings*. Ed. by L. Budach. Vol. 529. Lecture Notes in Computer Science. Springer, 1991, pp. 68–85. URL: %5Curl%7Bhttps://doi.org/10.1007/3-540-54458-5_51%7D (cit. on pp. 26, 28).

[SFS08] N. Sommer, M. Feder, O. Shalvi. “Low-Density Lattice Codes”. In: *IEEE Trans. Inf. Theory* 54.4 (2008), pp. 1561–1585. DOI: 10.1109/TIT.2008.917684. URL: https://doi.org/10.1109/TIT.2008.917684 (cit. on p. 19).

[SHRS17] J. M. Schanck, A. Hulsing, J. Rijneveld, P. Schwabe. “NTRU-HRSS-KEM”. In: *Technical report, National Institute of Standards and Technology*. 2017. URL: %5Curl%7Bhttps://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions%7D (cit. on p. 31).

[Van12] J. H. Van Lint. *Introduction to coding theory*. Vol. 86. Springer Science & Business Media, 2012 (cit. on p. 19).

All links were last followed on October 1, 2021.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature