

Master's programme in Mathematics and Operations Research

Vanishing Short Integer Solution: Reductions, Trapdoors, and Applications

Kalle Jyrkinen

Copyright © 2024 Kalle Jyrkinen

Author Kalle Jyrkinen

Title Vanishing Short Integer Solution: Reductions, Trapdoors, and Applications

Degree programme Mathematics and Operations Research

Major Applied Mathematics

Supervisor Prof. Chris Brzuska

Advisor Prof. Russell W. F. Lai

Date February 15, 2024 **Number of pages** 91+2 **Language** English

Abstract

The development of quantum computers has given rise to the field of post-quantum cryptography; that is, the study of cryptographic schemes that provide security even against quantum adversaries. One of the most promising directions is lattice-based cryptography, which studies schemes with security based on the hardness of computational problems over lattices.

Short Integer Solution (SIS) is one of the well-established lattice-based problems. Given a set of vectors as input, it asks to find a non-zero linear function with short coefficients which sends each of the input vectors to zero. Vanishing Short Integer Solution (vSIS) was recently proposed as a structured variant of SIS [Cini et al., CRYPTO'23]. It asks to find a non-zero polynomial with short coefficients which vanishes at each of the input points.

Despite recent progress of using vSIS for more efficient cryptographic constructions, notably lattice-based succinct arguments with quasi-linear time provers, not much research effort focused on the hardness of vSIS and the assumption has only been based on heuristic evidence. This thesis aims towards bridging this gap. We show that vSIS (in a certain parameter regime) is as hard as the problem of finding short elements in an ideal lattice (ideal-SVP). To complete the picture, we also explore the connections between different variants of vSIS.

Finally, we aim to create tools for vSIS-based cryptography. Towards this, we showcase how to build trapdoors for the vSIS problem. This is a powerful result, since vSIS trapdoors can serve as drop-in replacements for the SIS trapdoors used in many advanced lattice-based constructions. Furthermore, we observe that vSIS polynomials have interesting homomorphic properties. We demonstrate this by constructing a new homomorphic signature scheme for low-degree polynomials.

Keywords Lattice-based cryptography, post-quantum cryptography, Vanishing Short Integer Solution, trapdoor, ideal lattice, NTRU

Tekijä Kalle Jyrkinen

Työn nimi Häviävä lyhyt kokonaislukuratkaisu: reduktiot, takaportit ja sovellukset

Koulutusohjelma Mathematics and Operations Research

Pääaine Applied Mathematics

Työn valvoja Prof. Chris Brzuska

Työn ohjaaja Prof. Russell W. F. Lai

Päivämäärä 15.2.2024

Sivumäärä 91+2

Kieli englanti

Tiivistelmä

Kvanttitietokoneiden kehityksen myötä on alettu tutkia niin sanottuja kvanttiturvallisista kryptosysteemeistä eli menetelmiä, jotka tarjoavat turvaa myös kvanttietokoneella varustettua hyökkääjää vastaan. Yksi lupaavimmista lähestymistavoista on hilaperusteinen kryptografia. Se tutkii menetelmiä, joiden turvallisuus perustuu oletukseen eräiden laskennallisten hilaongelmien vaikeudesta.

Lyhyt kokonaislukuratkaisu (SIS) on yksi vakiintuneista ongelmista hilaperusteisen kryptografian alalla. Siinä tavoitteena on löytää ei-triviaali lineaarinen funktio jonka kertoimet ovat pieniä ja joka kuvaa jokaisen annetuista vektoreista nolaksi. Häviävä lyhyt kokonaislukuratkaisu (vSIS) on vastikään esitelty SIS-ongelman rakenteinen variantti [Cini et al., CRYPTO'23]. Siinä tavoitteena on löytää ei-triviaali pienikertoiminen polynomi jolla on annetut pisteet nolakohtina.

Tuoreessa tutkimuksessa vSIS-ongelman pohjalta onnistuttiin rakentamaan tehokkaita kryptografisia menetelmiä — kenties huomionarvoisimpana hilapohjainen kompakti argumentti, jonka todistajan tarvitsema aika on kvasilineaarinen. Tästä huolimatta vSIS-ongelman laskennallista vaikeutta ei ole juuri tutkittu ja menetelmien oletukset ovat perustuneet heuristiselle näytölle. Tämä diplomityö täydentää ymmärrystä tältä osin. Työssä näytetään, että (tietyillä parametreilla) vSIS-ongelman ratkaiseminen on vähintään yhtä vaikeaa kuin lyhyen vektorin löytäminen (SVP-ongelma) ideaalihilassa. Kokonaiskuvan täydentämiseksi tutkitaan myös yhteyksiä vSIS-ongelman eri variaatioiden välillä.

Diplomityön tavoitteena on aiemmin mainitun lisäksi myös luoda uusia työkaluja vSIS-pohjaisen kryptografian tarpeisiin. Tähän liittyen työssä näytetään, kuinka vSIS-ongelmaan voidaan sisällyttää takaportti. Tulos on merkittävä, sillä vSIS-takaportilla voidaan suoraan korvata SIS-takaportti lukuisissa olemassaolevissa hilapohjaisissa kryptografisissa sovelluksissa. vSIS-polynomeilla on lisäksi mielenkiintoisia homomorfisia ominaisuuksia. Tämän demonstroimiseksi työssä rakennetaan uusi homomorfinen digitaalinen allekirjoitusalgoritmi matalan asteluvun polynomeille.

Avainsanat Hilaperusteinen kryptografia, kvanttiturvallinen kryptografia, häviävä lyhyt kokonaislukuratkaisu, takaportti, ideaalihila, NTRU

Preface

First and foremost, I would like to thank my advisor, Professor Russell W. F. Lai; this thesis was made possible by his innovative research ideas and guidance along the way. I also want to thank Professor Chris Brzuska and the rest of the Aalto Cryptography group for all of the discussions and support.

Special thanks to Professor Damien Stehlé for correspondence regarding the details of the algorithm computing the small two-element representation of ideals. Similarly, thanks to Professor Martin R. Albrecht for ideas on the connections between NTRU and v SIS. I am grateful to both for taking the time to write insightful responses in the middle of their other responsibilities.

Lastly: thanks to family, friends, and everyone else who has supported me during my studies and made the journey so fun and memorable. I am incredibly fortunate to have all of you in my life.

Espoo, 15 February 2024

Kalle Jyrkinen

Contents

Abstract	3
Abstract (in Finnish)	4
Preface	5
Contents	6
Symbols and abbreviations	8
1 Introduction	10
1.1 Lattice-based cryptography	10
1.2 NTRU and Vanishing-SIS	11
1.3 Contributions	11
1.4 Technical overview	12
1.5 Discussion	14
2 Preliminaries	17
2.1 Notation and general definitions	17
2.1.1 Basic notation	17
2.1.2 Matrices over commutative rings	17
2.1.3 Modules	18
2.1.4 Polynomials	19
2.1.5 Probability distributions	20
2.1.6 Gram-Schmidt orthogonalization	21
2.1.7 Extended Euclidean algorithm	22
2.2 Cryptography	23
2.2.1 Complexity theory	23
2.2.2 Modeling security	24
2.2.3 Signature schemes	24
2.3 Lattices	26
2.3.1 Definition and important results	26
2.3.2 Worst-case computational problems	28
2.3.3 Average-case computational problems	29
2.3.4 SIS trapdoors	30
2.4 Algebraic number theory	30
2.4.1 Field extensions	31
2.4.2 Number fields	33
2.4.3 Rings of integers	34
2.4.4 Ideals	34
2.4.5 Example: Cyclotomic fields	37
2.4.6 Splitting of primes	37
2.4.7 Embeddings and ideal lattices	38
2.4.8 Computational problems over ideal lattices and rings	41

2.4.9	NTRU	41
3	Connections between different vSIS variants	44
3.1	Different degree of the polynomial	44
3.2	Different number of variables	47
3.3	Different number of points	47
3.4	Different moduli	48
3.5	From worst-case vSIS to average-case vSIS	49
4	A reduction from ideal-HSVP to vSIS	51
4.1	The reduction	51
4.2	On the restrictions	55
5	vSIS trapdoors	58
5.1	Trapdoors for univariate, single-point vSIS	58
5.1.1	vSIS modules	58
5.1.2	The trapdoor basis	59
5.1.3	Solving for the last column	60
5.2	Preimage sampling	62
5.3	Bounding the Gram-Schmidt norm	63
5.3.1	Lower bound for the norm	63
5.3.2	Numerical results	65
5.4	Nearest plane algorithm	68
5.5	vSIS trapdoor sampling algorithm	70
5.6	Hiding the NTRU secret	70
5.7	Trapdoors for multivariate vSIS	72
6	Applications	76
6.1	Replacing SIS trapdoors	76
6.2	Single-data homomorphic signatures from vSIS trapdoors	76
6.2.1	Homomorphic signatures	77
6.2.2	vSIS-based construction	79
6.3	From single-data to multi-data	83
6.3.1	A selectively secure multi-data HS scheme	84
6.3.2	A fully secure multi-data HS scheme	85
	References	87
A	Computing a somewhat short generator of an ideal	92

Symbols and abbreviations

Notation

$[n]$	the set $\{1, 2, \dots, n\}$
$\binom{i}{j}$	binomial coefficient
ϕ	Euler's totient function
\log	base 2 logarithm
\mathbb{N}	natural numbers (i.e., $\{1, 2, \dots\}$)
\mathbb{Z}	integers
\mathbb{Q}	rationals
\mathbb{R}	reals
\mathbb{C}	complex numbers
\mathbb{A}	algebraic numbers
\mathbb{B}	algebraic integers
\emptyset	empty set
$ z $	modulus of $z \in \mathbb{C}$
\bar{z}	complex conjugate of z
\mathbf{A}, \mathbf{x}	matrix, vector
$\mathbf{x}^T, \mathbf{x}^\dagger$	transpose, conjugate transpose of vector \mathbf{x}
$\ \mathbf{x}\ _p, \ \mathbf{x}\ , \ \mathbf{x}\ _\infty$	ℓ^p norm, Euclidean norm, infinity norm of vector \mathbf{x}
$\langle \mathbf{x}, \mathbf{y} \rangle$	inner product of vectors \mathbf{x} and \mathbf{y}
$\det(\mathbf{A})$	determinant of matrix \mathbf{A}
$\ \mathbf{A}\ _{\text{GS}}$	Gram-Schmidt norm of matrix \mathbf{A}
K	(number) field
L/K	field extension
$[L : K]$	degree of field extension
$\text{Gal}(L/K)$	Galois group of L over K
$N_{L/K}(x)$	field norm of $x \in L$ over $K \subseteq L$
$N(x), N(I)$	field norm of x over \mathbb{Q} , norm of ideal I
Δ_K	discriminant of a number field K
id	identity automorphism of a field
$\mathcal{R}, \mathcal{R}^\times$	ring, set of units in \mathcal{R}
$\mathcal{R}[X_1, \dots, X_n]$	polynomial ring in X_1, \dots, X_n over \mathcal{R}
\mathcal{R}/I	quotient ring
\mathcal{R}_q	$\mathcal{R}/q\mathcal{R}$ where $q \in \mathbb{N}$
$\langle z_1, \dots, z_n \rangle$	ideal generated by the set $\{z_1, \dots, z_n\}$
\mathfrak{p}	prime ideal
τ	coefficient embedding
σ	canonical embedding
γ_K	expansion factor of the field K
δ_K	$\max_i \{\ \sigma(b_i)\ _\infty\}$ where $\{b_i\}_i$ is a \mathbb{Z} -basis of K
$[x]$	integral part of x

$\{x\}$	fractional part of x
$\text{span}_{\mathcal{R}}(S)$	\mathcal{R} -span of set S
$\text{deg}(p)$	degree of polynomial p
$f \circ g$	composition of functions f and g
\wedge	logical “and”

Abbreviations

PQC	post-quantum cryptography
LBC	lattice-based cryptography
GS, GSO	Gram-Schmidt, Gram-Schmidt orthogonalization
EEA	extended Euclidean algorithm
PPT	probabilistic polynomial time
EUF-CMA	existential unforgeability under chosen message attack
(H)SVP	(Hermite) shortest vector problem
SIVP	shortest independent vectors problem
CVP	closest vector problem
(v)SIS	(Vanishing) Short Integer Solution
LWE	Learning with Errors
GPV	Gentry-Peikert-Vaikuntanathan
BKZ	block Korkine-Zolotarev
LLL	Lenstra–Lenstra–Lovász
(F)HS	(fully) homomorphic signature
ROM	random oracle model

1 Introduction

1.1 Lattice-based cryptography

For decades, most commonly used public-key cryptosystems have relied either on the computational hardness of the factorization of integers or that of the discrete logarithm problem (see, e.g. [1]). In 1994, Peter Shor showed that both of these problems could be solved in polynomial time using a sufficiently large quantum computer [2]. At the time, nobody knew if it was even possible to build such a machine. Nevertheless, the mere chance of this caused a stir in the scientific community. It was vital to begin the search for new efficient cryptosystems that would be secure even against attacks that utilize a quantum computer. This marked the dawn for the field of post-quantum cryptography (PQC).

Today, several parties have developed working quantum computers [3, 4, 5]. The number of qubits in them has also increased steadily, recently surpassing 1000 [6, 7]. Although it has become apparent that building large quantum computers is no easy feat, it does seem attainable. It is widely hypothesized that quantum computers could break classical cryptosystems during the upcoming decades, and the transition to quantum-resistant cryptosystems should be made well before that. In fact, the threat is already imminent when it comes to applications such as encryption; this is because a malicious party could be storing encrypted confidential information. In conclusion, the study of PQC is as important as ever.

One of the first candidates for PQC was lattice-based cryptography (LBC). The security of LBC relies on the assumption that certain computational problems over lattices (that is, discrete subsets of \mathbb{R}^n) are hard. Notable examples of such problems include, for instance, the shortest vector problem (SVP) and the shortest independent vectors problem (SIVP). Both of them can be traced back to the 19th century when they were studied by mathematicians like Dirichlet and Minkowski.

In 1996, Miklós Ajtai introduced these problems to cryptography, initiating the field of LBC. He proposed the Short Integer Solution (SIS) problem [8] and showed that, under certain parameters, it is at least as hard as SIVP. The SIS problem essentially asks one to find a short preimage of the function $\mathbf{x} \mapsto \mathbf{A}\mathbf{x} \bmod q$, given an integer q and a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Another fundamental problem in LBC, Learning with Errors (LWE), was introduced in 2005 by Oded Regev [9]. It comes in two variants: search-LWE asks to learn $\mathbf{s} \in \mathbb{Z}_q^n$ given a sequence of evaluations of a noisy linear function, given by a vector $\mathbf{b}^T = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e}^T$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is a public random matrix and $\mathbf{e} \in \mathbb{Z}_q^m$ is a noise vector with short coefficients. Decision-LWE asks one to distinguish between such \mathbf{b} and a uniform one in \mathbb{Z}_q^m .

Currently, LBC remains as one of the most promising contenders for building PQC. This is evident, for example, when looking at the candidates of the NIST PQC standardization process [10]. The efficiency and capabilities of applications have improved drastically in recent years. A typical approach to improve the efficiency is to take advantage of so-called algebraic lattices [11]. A common approach involves selecting a ring \mathcal{R} and formulating the problems over the ring or, more generally, \mathcal{R} -modules. Such problems include the ring versions of SIS [12] and LWE [13], as

well as the module versions of both [14]. Usually, \mathcal{R} is chosen to be the ring of integers of a number field or a specific polynomial ring.

1.2 NTRU and Vanishing-SIS

Another problem relying on algebraic lattices is the NTRU problem, introduced by Hoffstein et al. in [15] as early as 1996. Like LWE, it comes in two variants: search-NTRU asks, given $h \in \mathcal{R}$, to find short f, g such that $gh + f = 0 \pmod{q}$ and decision-NTRU asks to distinguish the h for which there exist such short f, g from uniform samples in \mathcal{R} . NTRU typically offers great efficiency for applications, and it has found wide usage over more than two decades. Some of the earlier applications include NTRUSign [16] and NTRUEncrypt [15]. The signature scheme FALCON [17] provides a notable recent example.

Despite its early introduction and wide usage, the NTRU assumption has mainly been based on heuristic evidence. It was only fairly recently that Pellet-Mary and Stehlé reduced the hardness of search-NTRU from the hardness of SVP restricted to a subset of lattices called ideal lattices (the ideal-SVP problem) [18]. The result was then improved by Felderhoff et al. via a new ideal-SVP self-reduction [19]. In addition, [20] provided further insight about the hardness of NTRU by presenting a reduction from the problem of finding a unique short element in a rank-2 module (Module Unique-SVP) to worst-case search-NTRU.

In [21], Cini et al. built a variety of new applications, including a recursive folding protocol for linear relations as well as a lattice-based verifiable delay function and a quasi-linear-time prover succinct argument for NP. The security of these was based on a new assumption called Vanishing-SIS (vSIS). Interestingly, vSIS can be viewed as a generalization of search-NTRU.

1.3 Contributions

Although [21] presents heuristic evidence of the hardness of the vSIS problem, there has not been much further research devoted to the hardness of the problem. Therefore, the first objective of this thesis is to develop a better understanding of the hardness of vSIS. In Section 3, we relate the hardness between vSIS problems for a variety of different parameters. Then, in Section 4, we generalize the results of [18] to show that vSIS (for certain parameters) is at least as hard as a subset of instances of the Hermite Shortest Vector Problem in ideal lattices (ideal-HSVP). Our reduction is from worst-case to worst-case, or alternatively, from average-case to average-case.

Although we do not succeed in reducing the hardness of vSIS from worst-case ideal-HSVP, we consider this work to be a first step towards this goal, which hopefully inspires further research. Such a result would be interesting for two reasons. The first reason is practical: it would give more credibility to the vSIS assumption and to the security of the applications. The second is a more theoretical observation; in [18], the authors suggest that NTRU might be strictly harder than ideal-SVP. Since vSIS can be viewed as a generalization of NTRU, it could serve as a means of breaking this hardness gap into smaller pieces.

The second main objective of this thesis is to build new tools for vSIS cryptography. We introduce vSIS trapdoors, a generalization of the widely used NTRU trapdoors [16, 22]. According to our numerical results, vSIS trapdoors seem to offer better efficiency than the corresponding NTRU trapdoors; the tradeoff is the need for a stronger security assumption.

vSIS trapdoors also offer some additional flexibility compared to NTRU trapdoors and, therefore, seem to imply stronger applications. We illustrate this in Section 6 by building a new homomorphic signature (HS) scheme for low-degree polynomials. Our scheme can be shown to be selectively secure (see [23]) for a single data set. Moreover, using transformations from [23], we can modify the scheme to handle multiple data sets while still satisfying selective security in the standard model or full security in the random oracle model (ROM).

1.4 Technical overview

In this section, we attempt to provide a high-level explanation of the ideas behind the results of sections 4, 5 and 6. In Section 3, we explore the connections between different variants of the vSIS problem; for instance, between different degrees of polynomials, different number of variables, or different number of points where the polynomial is required to vanish. We also provide a worst-case to average-case reduction for vSIS. However, as the section consists of a multitude of smaller independent results, we choose not to include it in this overview.

In the following, we will consider $K = \mathbb{Q}[X]/\langle X^n + 1 \rangle$ and $\mathcal{R} = \mathbb{Z}[X]/\langle X^n + 1 \rangle$ where n is a power of 2; a “short” element in \mathcal{R} refers to one in which the absolute values of the coefficients are small. Furthermore, we write $\lfloor \cdot \rfloor$ for the coefficient-wise rounding in K and $\{ \cdot \}$ for taking the (balanced) fractional part of the coefficients. Finally, the parameter q can be assumed to be a prime in \mathbb{N} .

Reducing vSIS from ideal-HSVP In Section 4, we show how solving the vSIS problem is at least as hard as finding short elements in certain ideals of \mathcal{R} . In particular, we consider the ideal-HSVP $_\gamma$ problem; we refer the reader to Section 2.4.8 for a precise definition. Let us illustrate the main idea in the special case of principal integral ideals $I \subseteq \mathcal{R}$ that can be written as $I = \langle z^D \rangle = \{ rz^D \mid r \in \mathcal{R} \}$ for some $z \in \mathcal{R}$. Moreover, let $D \in \mathbb{N}$ be an upper bound for the degree of the vSIS solution polynomial.

Now, suppose that we want to find a short, non-zero element in I by using an oracle to the vSIS problem. We begin by computing

$$v = \left\lfloor \frac{q}{z} \right\rfloor = \frac{q}{z} - \left\{ \frac{q}{z} \right\} \bmod q^D.$$

We input this to the vSIS oracle, assuming to receive $p \in \mathcal{R}[Y]$ that has short coefficients and satisfies $p(v) = 0 \bmod q^D$. We have to show two things; that the vSIS problem is well-posed (i.e., a solution exists) and that any solution to the problem implies a solution to the ideal-HSVP problem. Both are true, given that the choice of parameters is appropriate.

For the first one, we consider the polynomial $p = \alpha(Y + \{q/z\})^D$ where α denotes the shortest non-zero element in I . Since $\alpha \in I$, it can be written as $\alpha = rz^D$ for some $r \in \mathcal{R}$. This implies that $p(v) = \alpha (q/z)^D = 0 \pmod{q^D}$ and that $p \in \mathcal{R}[Y]$. Since Minkowski's bound gives an upper bound for the norm of α , the coefficients of p are somewhat short.

For the second one, we can prove that the leading coefficient of any such p is a short element in $I \setminus \{0\}$; the approach is elementary but the details turn out to be somewhat technical. The idea is to write the equation $p(v) = 0 \pmod{q}$ as

$$\sum_{i=0}^D p_i \left(\frac{q}{z} - \left\{ \frac{q}{z} \right\} \right)^i = r' q^D,$$

where p_0, \dots, p_D are the coefficients of p and r' is some arbitrary element of \mathcal{R} . Expanding the highest-degree term on the left-hand side, we can write

$$p_D \frac{q}{z} + s = r' q^D,$$

where s captures all of the remaining terms of the right-hand side. Now, multiplying both sides by α/q^D (where α is still the shortest element in I) yields

$$\frac{p_D \alpha}{q^D} + \frac{s \alpha}{q^D} = r' \alpha.$$

The key is to bound the norm of the coefficient vector of the middle term; for large enough q we can show that the norm is less than 1. Moreover: since the left- and rightmost terms are in \mathcal{R} , we find that the middle term must be in \mathcal{R} as well. The only way to satisfy both facts is for the middle term to be equal to zero. This in turn implies $p_D \in I$; that is, we have found a short, non-zero element in I .

Trapdoors for vSIS In Section 5, we explain how to generate vSIS instances together with a trapdoor that allows one to solve vSIS with respect to that specific instance. More importantly, the solutions can be sampled randomly (we call this *trapdoor sampling*), which allows one to give out solutions without leaking information about the trapdoor itself. This is useful for constructing various applications.

Our approach is to generalize the techniques that are familiar from the context of NTRU trapdoors [16, 22]. To sample an NTRU instance h together with a trapdoor, one typically samples short f, g in \mathcal{R} and defines $h = f/g \pmod{q}$; by the decision-NTRU assumption, this quotient is indistinguishable from uniformly random in $\mathcal{R}/q\mathcal{R}$. However, if one has access to f and g they can solve NTRU with respect to h since $gh - f = 0 \pmod{q}$. The idea then is to find $F, G \in \mathcal{R}$ such that $Gh - F = 0 \pmod{q}$ and $(f, g), (F, G)$ are \mathcal{R} -linearly independent elements in the module \mathcal{R}^2 ; this gives a complete basis of the *NTRU lattice* $\{(a, b) \in \mathcal{R}^2 \mid ah + b = 0 \pmod{q}\}$. For an appropriate range of parameters (the norms of f, g around $1.17\sqrt{q}$ [22]) the basis works as a trapdoor for the lattice.

To construct trapdoors for vSIS, we choose $h = f/g \pmod{q}$ to be a quotient of short ring elements just as before. We define the corresponding *vSIS module* of

degree D as the set of polynomials of degree at most D satisfying $p(h) = 0 \pmod q$ or, equivalently, as the set

$$\{ (p_D, \dots, p_0) \in \mathcal{R}^{D+1} \mid p_D h^D + \dots + p_1 h + p_0 = 0 \pmod q \}.$$

Notice that $\{(g, -f, 0, \dots, 0), (0, g, -f, 0, \dots, 0), \dots, (0, \dots, 0, g, -f)\}$ is a set of D \mathcal{R} -linearly independent elements in the module. We demonstrate how a basis completion technique, similar to those familiar from the context of NTRU, can be used to efficiently find one more independent element in the module.

To complete the picture, we provide numerical results that support the hypothesis that the construction provides a meaningful trapdoor. In particular, we find that the Gram-Schmidt norm of the completed basis is expected to be slightly over $q^{1/(D+1)}$ for an optimal choice of parameters. This result is consistent with the existing results for the special case of NTRU trapdoors. More importantly, it demonstrates how the use of vSIS allows for more flexibility in the choice of parameters than NTRU.

Homomorphic signatures for polynomials vSIS trapdoors fall under the broader category of SIS trapdoors (discussed in detail in Section 2.3.4). As a consequence, there exists a wide range of applications that can be modified to use vSIS trapdoors instead. However, it seems that this is not the full extent of the capabilities of vSIS trapdoors; we demonstrate this in Section 6 by constructing a vSIS-based HS scheme for low-degree polynomials.

We start from a fairly simple observation. Let $v \in \mathcal{R}$ and $f_1, f_2 \in \mathcal{R}[Y]$ be polynomials satisfying $f_i(v) = 0 \pmod q$ for $i \in \{1, 2\}$. Moreover, define $s = f_1 + f_2$ and $p = f_1 f_2$. We obtain three elementary properties for s and p : (i) we have $s(v) = p(v) = 0 \pmod q$, (ii) the constant coefficients satisfy $s(0) = f_1(0) + f_2(0)$ and $p(0) = f_1(0)f_2(0)$, and (iii) if f_1 and f_2 have short coefficients, the coefficients of s and p are also somewhat short. These findings generalize from just the sum and product to arbitrary polynomial functions (with fixed depth and short coefficients) of f_1 and f_2 . Similarly, they generalize from just two vanishing polynomials to any number, as long as there is some constant upper bound.

Now, suppose that we have access to a vSIS trapdoor of degree D . Then, we can sample polynomials of degree $D + 1$ where we choose the constant coefficient and the rest of the coefficient are short yet random. This, together with the property (ii), allows us to encode short messages in the constant coefficient. Combining this reasoning with the blueprint borrowed from the SIS-based leveled HS scheme proposed by Gorbunov et al. [23], we obtain a vSIS-based somewhat homomorphic signature scheme.

1.5 Discussion

Different variants of vSIS In Section 3, we provide a wide variety of reductions between different variants of the vSIS problem. It is worth noting that, apart from the trivial reductions, the parameters often exhibit exponential dependencies. This can limit the practical applicability of the results. Moreover, in some of the reductions, we have to consider the problem of obtaining several linearly independent vSIS solutions.

This can make the problem considerably harder and even make the problem have no solution to begin with.

Nevertheless, we consider pointing out even trivial connections between the different variants of vSIS to have value, given the limited amount of similar research currently available. It is also possible that many of the parameters could be improved in the future via the use of more sophisticated techniques, which we propose as a potential open direction for further research.

We note that, in addition to different parameterizations, one can go further in the search for different variants of vSIS. One natural source of inspiration is NTRU. For instance, “decision vSIS” could ask, given $v \in \mathcal{R}$, to distinguish whether there exists a short bounded-degree polynomial vanishing at v or not. Similarly, one could attempt to generalize the “NTRU_{mod}” problem (see [18]). On the other hand, LWE with respect to the matrix $\mathbf{A} = [v^d \ \cdots \ v \ 1]$ gives a natural LWE analogue of vSIS, and it could be interesting to explore the possibilities of this “Vanishing-LWE” problem. We leave a more comprehensive inspection of such problems as a topic for future research.

vSIS and ideal-HSVP We emphasize the fact that the reduction from ideal-HSVP to vSIS in Section 4 is unsatisfactory in the sense that it is not from worst-case ideal-HSVP. Rather, the instances of the ideal-HSVP problem have to satisfy several specific requirements, as discussed in detail in Section 4.2. While the reduction could be stronger in this perspective, our results demonstrate how the vSIS problem is connected to finding short elements in ideal lattices. Removing the limitations of the reduction remains an open problem; such a result would be of theoretical interest as it would yield a hierarchy of problems from ideal-HSVP to search-NTRU, with the vSIS problems lying in between.

Another direction could involve exploring the connections between vSIS and SVP problems in modules, similar to what was done in [20] in the case of NTRU. We note that due to the fact that vSIS lattices are even more structured than NTRU lattices, it seems difficult to directly generalize the techniques of [20].

Trapdoors The construction of NTRU trapdoors discussed in Section 5 is perhaps the most practically significant result of the thesis. We manage to extend many well-established results from the context of NTRU trapdoors to the context of vSIS. We also provide heuristic evidence that vSIS trapdoors have the practical advantage of being able to sample shorter preimages than NTRU trapdoors, potentially improving the efficiency of cryptographic schemes based on SIS trapdoors.

However, we only construct trapdoors for univariate, single-point vSIS; further generalizations are left as an open question. In Section 5.7 we discuss the difficulties of multivariate trapdoors, and trapdoors for the multiple point vSIS problem do seem to be even more problematic. Both of these directions could have practical implications: whereas allowing multivariate trapdoors would provide increased flexibility, considering multiple points would mean that breaking the schemes requires solving a more difficult problem. The latter could result in more secure schemes.

On the applications The HS scheme of Section 6 is presented mainly to demonstrate the homomorphic properties of vanishing polynomials. In particular, we have not found practical advantages of using the scheme over existing ones such as the one of [23]. A major limitation of our scheme is that the scheme only allows evaluation of polynomial functions up to a fixed degree. It remains an open question whether this limitation can be lifted; for example, by using a “bootstrapping” technique similar to that of [24]. Apart from this, another potential direction for future research would be to construct other homomorphic schemes using vSIS trapdoors or, more generally, vanishing polynomials. One interesting question is if the aforementioned Vanishing-LWE problem could be used to construct a homomorphic encryption scheme, similar to the LWE-based fully homomorphic encryption scheme of [25].

2 Preliminaries

In this section, we cover the preliminaries necessary to present our results. We assume that the reader is familiar with some elementary notions in abstract algebra (especially groups, rings and their ideals, fields). We also assume some familiarity with linear algebra and elementary number theory. Good references in these subjects include, e.g. [26], for abstract and linear algebra, and [27], for number theory.

Several sections are largely inspired by the (unpublished) lecture notes in lattice-based cryptography, written for the spring 2023 Aalto University course “Advanced Topics in Cryptography” by Russell W. F. Lai [28].

2.1 Notation and general definitions

2.1.1 Basic notation

For a ring \mathcal{R} , we denote its set of units as \mathcal{R}^\times . Moreover, for $q \in \mathbb{N}$ we use the shorthand \mathcal{R}_q to refer to the quotient ring $\mathcal{R}/q\mathcal{R}$. When discussing a ring, we always refer to a commutative ring with unity.

Vectors are written in bold lowercase letters, and matrices are written in bold uppercase letters. The i th element of a vector \mathbf{x} is denoted by x_i . For $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{C}^n$ and $p > 1$ we define the ℓ^p norm of \mathbf{x} as

$$\|\mathbf{x}\|_p = (|x_1|^p + \dots + |x_n|^p)^{\frac{1}{p}}.$$

For $p \rightarrow \infty$ we obtain the infinity norm, defined as $\|\mathbf{x}\|_\infty = \max_i |x_i|$. We take $p = 2$ when omitted.

Furthermore, for $\mathbf{a}, \mathbf{b} \in \mathbb{C}^n$, we denote the standard complex inner product as $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{b}^\dagger \mathbf{a} = \sum_{i=1}^n \overline{b_i} a_i$, where \mathbf{b}^\dagger denotes the conjugate transpose of \mathbf{b} and $\overline{b_i}$ the complex conjugate of b_i .

For $n \in \mathbb{N}$, we denote $[n] = \{1, \dots, n\}$.

2.1.2 Matrices over commutative rings

Let \mathcal{R} be a commutative ring and suppose that \mathbf{M} is a square matrix over \mathcal{R} , i.e. $\mathbf{M} \in \mathcal{R}^{n \times n}$ for some $n \in \mathbb{N}$. We say that \mathbf{M} is invertible over \mathcal{R} if there exists $\mathbf{M}^{-1} \in \mathcal{R}^{n \times n}$ such that $\mathbf{M}\mathbf{M}^{-1}$ is the identity matrix. Such matrices are said to be *unimodular* over \mathcal{R} , and they can easily be characterized using their determinant.

Definition 2.1 (Unimodularity). Let \mathcal{R} be a ring, $n \in \mathbb{N}$ and $\mathbf{M} \in \mathcal{R}^{n \times n}$. We say that matrix \mathbf{M} is unimodular over \mathcal{R} if $\det(\mathbf{M}) \in \mathcal{R}^\times$.

The *Laplace expansion* gives a convenient recursive formula for the determinant of a matrix.

Definition 2.2 (Laplace expansion). Let \mathcal{R} be a commutative ring and $\mathbf{A} \in \mathcal{R}^{n \times n}$. Also, for $i, j \in [n]$ let $M_{i,j}$ denote the (i, j) minor of \mathbf{A} , that is, the determinant of the

$(n - 1) \times (n - 1)$ submatrix of \mathbf{A} obtained by deleting the i th row and the j th column of \mathbf{A} . Then, for all $i \in [n]$,

$$\det(\mathbf{A}) = \sum_{j=1}^n (-1)^{i+j} a_{i,j} M_{i,j}.$$

This is called the Laplace expansion along the i th row of \mathbf{A} . Similarly, for all $j \in [n]$,

$$\det(\mathbf{A}) = \sum_{i=1}^n (-1)^{i+j} a_{i,j} M_{i,j},$$

which is called the Laplace expansion along the j th column of \mathbf{A} .

2.1.3 Modules

For a commutative ring \mathcal{R} , we will frequently encounter the concept of \mathcal{R} -modules (or just modules when the ring \mathcal{R} is clear from the context).

Definition 2.3 (Module). Let \mathcal{R} be a commutative ring. An abelian group \mathcal{M} , equipped with an operation $\cdot : \mathcal{R} \times \mathcal{M} \rightarrow \mathcal{M}$, is an \mathcal{R} -module if for all $r, s \in \mathcal{R}, m, n \in \mathcal{M}$,

- (i) $(r + s)m = rm + sm$,
- (ii) $r(m + n) = rm + rn$,
- (iii) $r(sm) = (rs)m$ and
- (iv) $1m = m$,

where 1 denotes the multiplicative identity of \mathcal{R} .

When \mathcal{R} is a field, an \mathcal{R} -module is in fact a vector space. Thus, modules provide a generalization of the concept of vector spaces. Familiar notions such as linear independence and span extend naturally to modules.

Definition 2.4. Let \mathcal{R}, \mathcal{M} be as in Definition 2.3 and denote the additive identity of \mathcal{M} by 0. Then:

- A finite subset $\{m_1, \dots, m_n\} \subseteq \mathcal{M}$ is said to be \mathcal{R} -linearly independent if

$$\sum_{i \in [n]} r_i m_i = 0 \quad \text{and} \quad r_i \in \mathcal{R} \quad \forall i \in [n]$$

implies $r_1 = \dots = r_n = 0$.

- The *span* (or more explicitly, \mathcal{R} -span) of a finite subset $\{m_1, \dots, m_n\} \subseteq \mathcal{M}$ is denoted as $\text{span}_{\mathcal{R}}(\{m_1, \dots, m_n\})$ and defined as the set

$$\left\{ \sum_{i \in [n]} r_i m_i \mid r_i \in \mathcal{R} \quad \forall i \in [n] \right\}.$$

- The *rank* of \mathcal{M} is defined as the maximal size of an \mathcal{R} -linearly independent subset of \mathcal{M} .

Since rings provide less algebraic structure than fields, some properties are lost when considering modules instead of vector spaces. For one, a module does not always have a basis. If a module can be given a basis, we say that it is *free*; in this thesis, we mainly work with free modules.

Definition 2.5 (Free module). Let \mathcal{R} be a commutative ring and \mathcal{M} be an \mathcal{R} -module of rank $k \in \mathbb{N}$. \mathcal{M} is said to be free if it has a basis B ; that is, an \mathcal{R} -linearly independent subset $\{b_1, \dots, b_k\} \subseteq \mathcal{M}$ such that every $m \in \mathcal{M}$ can be written as a finite sum

$$m = \sum_{i=1}^k r_i b_i$$

for some $r_1, \dots, r_k \in \mathcal{R}$. In this case, we say that B *generates* \mathcal{M} .

A basis does not need to be unique. The following lemma characterizes when two finite bases generate the same module.

Lemma 2.6. Let \mathcal{R} be a commutative ring and $n, k \in \mathbb{N}$ such that $n \geq k$. Furthermore, let $B = \{b_1, \dots, b_k\}$ and $B' = \{b'_1, \dots, b'_k\}$ be \mathcal{R} -linearly independent subsets of \mathcal{R}^n . Then, B and B' generate the same module if and only if there exists $\mathbf{U} \in \mathcal{R}^{k \times k}$ such that

$$[b_1 \ \cdots \ b_k] \mathbf{U} = [b'_1 \ \cdots \ b'_k]$$

and \mathbf{U} is unimodular over \mathcal{R} .

2.1.4 Polynomials

To avoid confusion especially when dealing with multivariate polynomials, let us introduce the related notation.

Definition 2.7 (Polynomials). Let $m, n \in \mathbb{N}$ and $\mathbf{X} = (X_1, \dots, X_n)$ be a tuple of indeterminates; moreover, let us denote the exponent vectors as $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$. A set of coefficients $p_\alpha \in \mathcal{R}$ define a polynomial

$$p = \sum_{\alpha} p_{\alpha} \mathbf{X}^{\alpha} = \sum_{\alpha} p_{\alpha} X_1^{\alpha_1} \cdots X_n^{\alpha_n} \in \mathcal{R}[X_1, \dots, X_n].$$

For univariate polynomials, we will use the simpler notation

$$p = \sum_{i=0}^m p_i X^i \in \mathcal{R}[X].$$

A polynomial $p \in \mathcal{R}[X_1, \dots, X_n]$ is said to be *irreducible* if it cannot be expressed as a product of two non-constant polynomials in $\mathcal{R}[X_1, \dots, X_n]$. Furthermore, p is

said to be *monic* if its leading coefficient (with respect to a monomial order) is the identity in the ring \mathcal{R} .

We will utilize the following results about the number of multivariate polynomials of a certain degree. The first of them is a consequence of elementary combinatorics (it is equivalent to counting the number of ways D indistinguishable elements can be put into w bins) and the second follows by summation over different degrees. Thus, we will omit the proofs.

Claim 2.8. *Let $D \in \mathbb{N}$. There are $\binom{D+w-1}{D}$ distinct w -variate monomials of degree D .*

Corollary 2.9. *There are $\binom{w+D}{w} = \frac{(w+1)\dots(w+D)}{D!}$ distinct w -variate monomials of degree less than or equal to D .*

To obtain bounds for some of our results, we need to bound the magnitude of coefficients resulting from exponentiation of polynomials. To enable a clearer presentation of these results, we introduce the following notation.

Definition 2.10. Let $D \in \mathbb{N}$, $\mathbf{X} = (X_1, \dots, X_w)$ be a vector of indeterminates and $p \in \mathbb{Z}[\mathbf{X}]$ be a w -variate degree- D polynomial having all coefficients equal to 1, i.e.

$$p = \sum_{\alpha : \sum_i \alpha_i \leq D} \mathbf{X}^\alpha.$$

For $n \in \mathbb{N}$, we define $v_{w,D,n}$ as the largest coefficient of p^n .

Remark 2.11. Obtaining an expression or even a tight bound for $v_{w,D,n}$ is beyond the scope of this thesis, and the author is unaware of existing related results. However, it seems to be related to some known problems in combinatorics. For example, $v_{1,D,n}$ is the maximum of the number of $\{0, \dots, D\}$ -restricted compositions of an integer k into n parts over $k \in \{0, \dots, Dn\}$. For $w > 1$ the problem is somewhat more involved, but seems to be related e.g. to the properties of “extended Pascal triangles” [29].

To facilitate understanding, note that a trivial loose bound can be derived using Corollary 2.9: $v_{w,D,n} \leq \binom{w+D}{w}^n$. For $w = 1$ this reads $v_{1,D,n} \leq D^n$.

2.1.5 Probability distributions

Let \mathcal{X} be a distribution over a set S ; we denote $x \leftarrow \$ \mathcal{X}$ to imply that x is a random element sampled from \mathcal{X} . If S is finite, assigning a uniformly random element from S to x is denoted as $x \leftarrow \$ S$. $\mathcal{X}(x)$ denotes the probability density function of \mathcal{X} evaluated at x .

We define *statistical distance* that measures the closeness of two distributions. In order to determine when this distance is small, we also need the concept of *negligible functions*.

Definition 2.12 (Negligible function). A function $f : \mathbb{N} \rightarrow \mathbb{R}$ is said to be negligible if for every $c \in \mathbb{N}$ there exists N_c such that

$$|f(x)| < \frac{1}{x^c} \quad \forall x \geq N_c.$$

Any function that is not negligible is said to be *non-negligible*.

Definition 2.13 (Statistical distance). Let \mathcal{X}, \mathcal{Y} be distributions over a countable set S . Then, the *statistical distance* between \mathcal{X} and \mathcal{Y} is defined as

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

Let $\{\mathcal{X}_n\}_{n \in \mathbb{N}}$ and $\{\mathcal{Y}_n\}_{n \in \mathbb{N}}$ be two distribution ensembles. If $\Delta(\mathcal{X}_n, \mathcal{Y}_n)$ is negligible, $\{\mathcal{X}_n\}_{n \in \mathbb{N}}$ and $\{\mathcal{Y}_n\}_{n \in \mathbb{N}}$ are said to be *statistically close*.

Gaussian distributions form an important family of distributions; we are especially interested in *discrete Gaussian distributions*.

Definition 2.14 (Gaussian distributions). We define the Gaussian density function (over \mathbb{R}^n) with parameter $s > 0$ and center $\mathbf{c} \in \mathbb{R}^n$ as

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

for all $\mathbf{x} \in \mathbb{R}^n$. Moreover, let S be a discrete subset of \mathbb{R}^n . Then, the discrete Gaussian distribution over S with parameter $s > 0$ and center $\mathbf{c} \in \mathbb{R}^n$ is defined as

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

for all $\mathbf{x} \in S$.

2.1.6 Gram-Schmidt orthogonalization

In this thesis, we will frequently come across the *Gram-Schmidt orthogonalization* (GSO). Thus, to avoid any ambiguity, let us give a formal definition. It requires the notion of an orthogonal projection.

Definition 2.15 (Orthogonal projection). Let K be a field and V be an inner product space over K . For a subspace $W \subseteq V$, we denote the orthogonal projection of $\mathbf{v} \in V$ onto W as $\text{proj}_W(\mathbf{v})$. If I is an index set such that $\{\mathbf{w}_i\}_{i \in I}$ is a basis of W , then

$$\text{proj}_W(\mathbf{v}) = \sum_{i \in I} \frac{\langle \mathbf{v}, \mathbf{w}_i \rangle}{\langle \mathbf{w}_i, \mathbf{w}_i \rangle} \mathbf{w}_i.$$

An orthogonal projection onto a certain subspace is always unique. Therefore, GSO is well-defined.

Definition 2.16 (Gram-Schmidt orthogonalization). Let K, V be as in Definition 2.15, $\mathbf{b}_1, \dots, \mathbf{b}_m$ be linearly independent vectors in V and $\mathbf{B} = [\mathbf{b}_1 \ \dots \ \mathbf{b}_m]$. We call $\tilde{\mathbf{B}} = [\tilde{\mathbf{b}}_1 \ \dots \ \tilde{\mathbf{b}}_m]$ the Gram-Schmidt orthogonalization of \mathbf{B} , and it is defined as

$$\begin{cases} \tilde{\mathbf{b}}_1 = \mathbf{b}_1 \\ \tilde{\mathbf{b}}_i = \mathbf{b}_i - \text{proj}_{\text{span}_K(\{\mathbf{b}_1, \dots, \mathbf{b}_{i-1}\})}(\mathbf{b}_i), \quad i \geq 2. \end{cases}$$

The *Gram-Schmidt norm* (GS-norm) of the matrix \mathbf{B} is denoted $\|\mathbf{B}\|_{\text{GS}}$ and defined as the maximum of the ℓ_2 norm over the columns of $\tilde{\mathbf{B}}$, i.e.

$$\|\mathbf{B}\|_{\text{GS}} = \max_i \|\tilde{\mathbf{b}}_i\|.$$

Remark 2.17. Many authors consider the process where the columns of $\tilde{\mathbf{B}}$ are normalized, known as Gram-Schmidt orthonormalization. However, we deliberately choose to avoid this since the norms $\|\tilde{\mathbf{b}}_i\|$ often play a central role when working with lattices.

2.1.7 Extended Euclidean algorithm

In Section 5 we encounter linear Diophantine equations and, as a tool, we utilize the theory of Bézout coefficients for more than two elements. Recall that given $a_1, a_2 \in \mathbb{Z}$, there exists $x_1, x_2 \in \mathbb{Z}$ such that $x_1 a_1 + x_2 a_2 = \gcd(a_1, a_2)$. These can be efficiently computed using the extended Euclidean algorithm (EEA). This implies the following, more general claim.

Claim 2.18. *Let $n \geq 2$. For $a_1, \dots, a_n \in \mathbb{Z}$, there exists x_1, \dots, x_n such that*

$$\sum_{i=1}^n x_i a_i = \gcd(a_1, \dots, a_n). \quad (2.1)$$

Moreover, such x_1, \dots, x_n can be efficiently computed.

Proof. Use induction; for the base case, the claim holds for $n = 2$. Now, for the induction hypothesis, assume that the claim holds for some $n \geq 2$. Let $a_1, \dots, a_{n+1} \in \mathbb{Z}$. By the induction hypothesis, we may assume that we know $x'_1, \dots, x'_n \in \mathbb{Z}$ such that $\sum_{i=1}^n x'_i a_i = \gcd(a_1, \dots, a_n)$. Using EEA one can compute y_1, y_2 such that $y_1 \gcd(a_1, \dots, a_n) + y_2 a_{n+1} = \gcd(\gcd(a_1, \dots, a_n), a_{n+1}) = \gcd(a_1, \dots, a_{n+1})$; then

$$\begin{cases} x_i = y_1 x'_i, & i \in \{1, \dots, n\} \\ x_{n+1} = y_2 \end{cases}$$

satisfy $\sum_{i=1}^{n+1} x_i a_i = \gcd(a_1, \dots, a_{n+1})$. □

The proof is constructive; i.e., it provides a recursive algorithm for computing the coefficients x_i . Since it is a straight-forward generalization of EEA, we will simply refer to it as EEA as well in the rest of the thesis.

We will also need the following lemma about greatest common divisors in Section 5.

Lemma 2.19. *Let $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$. Then,*

$$\gcd(a^n, a^{n-1}b, \dots, ab^{n-1}, b^n) = \gcd(a, b)^n.$$

Proof. Denote $g = \gcd(a, b)$. Write a and b using their unique prime power decompositions:

$$a = \prod_{i=1}^m p_i^{e_i}, \quad b = \prod_{i=1}^m p_i^{f_i}$$

for some m , set of primes $\{p_i\}_{i=1}^m$ and corresponding powers $e_i, f_i \geq 0$. Then, $g = \prod_{i=1}^m p_i^{\min\{e_i, f_i\}}$. This implies

$$\gcd(a^n, b^n) = \prod_{i=1}^m p_i^{\min\{ne_i, nf_i\}} = \prod_{i=1}^m p_i^{n \min\{e_i, f_i\}} = g^n.$$

Now, since $g \mid a$ and $g \mid b$, g^n divides all of $a^n, a^{n-1}b, \dots, ab^{n-1}, b^n$. This means that g^n must be a divisor of $\gcd(a^n, a^{n-1}b, \dots, ab^{n-1}, b^n)$. However, we also have $\gcd(a^n, a^{n-1}b, \dots, ab^{n-1}, b^n) \leq \gcd(a^n, b^n) = g^n$. The only situation when both of these hold is when $g^n = \gcd(a^n, a^{n-1}b, \dots, ab^{n-1}, b^n)$. \square

2.2 Cryptography

In this section, our goal is to present the necessary prerequisites related to cryptography and complexity theory. As we can merely scratch the surface from both, we direct the readers looking for a more thorough coverage to [30] or [31] for cryptography, and e.g. [32] for complexity theory. In addition to these, this section is largely based on [33] and [28].

To give a general definition of cryptography, let us quote Oded Goldreich.

Cryptography is concerned with the construction of schemes that should be able to withstand any abuse. Such schemes are constructed so as to maintain a desired functionality, even under malicious attempts aimed at making them deviate from their prescribed functionality. [31]

In somewhat more concrete terms, most cryptography aims to maintain properties such as confidentiality, authenticity or integrity of communication over an insecure channel. That is, these properties should be satisfied even if there is a malicious adversary that may have the power, for example, to intercept messages or send messages of their own. Perhaps the most classic problem in cryptography is achieving secret communication between two parties; a solution to this problem is provided by encryption schemes. Other examples of cryptographic schemes include signature schemes and fault-tolerant protocols.

2.2.1 Complexity theory

To rigorously define the notion of an algorithm or talk about their running time, we need to consider some model of computation. For concreteness, we model algorithms using (quantum) Turing machines. Note that this choice is not significant in our context. This is because we are only interested whether the running time of an algorithm is polynomial or not; from this perspective, there exist many equivalent choices. In the context of classical algorithms, this is discussed e.g. in Chapter 1.2.3 of [32]. For quantum computation, quantum circuits provide an example of an equivalent model.

Definition 2.20. An *algorithm* is a sequence of operations that can be represented by a Turing machine or a quantum Turing machine. The algorithms that can be represented using solely the former are said to be *classical*; the rest are called *quantum* algorithms.

For a given input, the *running time* of the algorithm is the number of steps the machine takes to terminate. If the steps are allowed to involve randomness, we say that the algorithm is *probabilistic*; otherwise, the algorithm is said to be *deterministic*.

Suppose that \mathbf{x} is a function of the input of an algorithm; for example, it could represent the number of bits of the input. If, for any input, the running time of an

algorithm is $O(p(\mathbf{x}))$ for some polynomial p , we say that the algorithm is polynomial-time with respect to \mathbf{x} . This is equivalent to saying that the running time is $\text{poly}(\mathbf{x})$.

In the context of cryptography, we introduce a special parameter $\lambda \in \mathbb{N}$, called the *security parameter*. It often denotes the length of the cryptographic key (in bits); however, it can be used more generally to parameterize how difficult a certain scheme is to break or a certain problem is to solve. If the running time of an algorithm is $\text{poly}(\lambda)$, it is said to be *efficient*. We call the class of efficient probabilistic algorithms *probabilistic polynomial-time (PPT)* algorithms.

Remark 2.21. Following a usual convention, the security parameter is passed to a function as a sequence of λ bits, each with the value 1; this is denoted as 1^λ . This is often only done explicitly in the key generation phase, and we assume that λ is implicitly known for the rest of the algorithms.

To discuss the hardness of different computational problems, we consider *adversaries* (denoted by \mathcal{A}) that are represented by probabilistic algorithms. Since we study post-quantum cryptography, we allow the adversaries to be quantum algorithms. We are particularly interested in statements of type “no adversary \mathcal{A} that runs in time T can solve problem P with a probability greater than ϵ ”. Specifically, in the context of this thesis, we say that a problem P (parameterized by λ) is hard if no PPT \mathcal{A} can solve it with a probability greater than $\text{negl}(\lambda)$.

The concept of *reductions* allows us to relate the hardness of different problems. Suppose that we have problems P and Q ; assume that P is hard and that there exists a PPT adversary \mathcal{A} that solves Q with a non-negligible probability. Moreover, assume that we can construct \mathcal{B} , a PPT adversary that solves Q with a non-negligible probability by using \mathcal{A} as a subroutine. This contradicts the hardness of P , showing that Q must be at least as hard as P . In this case, we write $P \leq Q$ and \mathcal{B} is called a reduction from P to Q . Observe that \mathcal{A} can be modeled as a black box; we call it an *oracle* for the problem Q .

2.2.2 Modeling security

To state that a certain cryptographic scheme is secure, we formulate breaking (a certain security requirement of) the scheme as a computational problem. These are represented by interactive algorithms called *security games*, and their goal is to capture some notion of security that we require from the scheme. The games can interact with an arbitrary adversary, and the adversary wins the game if it manages to break the scheme under the relevant notion of security. The winning probability of a certain adversary is called its *advantage*. We say that the scheme satisfies a certain security notion if there does not exist an efficient adversary with a non-negligible advantage. When the relevant security notion is implicit from the context, we may simply say that a scheme is secure.

2.2.3 Signature schemes

To provide an example of a cryptographic primitive and a related security game, let us consider a signature scheme. This will also turn out to be useful regarding Section 6.

We use the same definitions as in [28]. Notice that we simply provide the syntax of the algorithms, along with the correctness requirement and a notion of security; we do not consider implementation of the algorithms.

Definition 2.22 (Signature scheme). A signature scheme for message space \mathcal{X} consists of the following PPT algorithms:

- $(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$: The key generation algorithm inputs the security parameter, generates the public and secret keys.
- $s \leftarrow \text{Sign}_{sk}(x)$: The signing algorithm inputs a message $x \in \mathcal{X}$, outputs a signature s .
- $b \leftarrow \text{Verify}_{pk}(x, s)$: The verification algorithms inputs a message x and a signature s , and outputs a bit $b \in \{0, 1\}$ that determines whether to accept the signature or not.

Definition 2.23. (Correctness) Let Σ be a signature scheme, defined as in Definition 2.22. Σ is said to be correct if, for any $\lambda \in \mathbb{N}$, $x \in \mathcal{X}$, $(pk, sk) \in \text{KeyGen}(1^\lambda)$ and $s \in \text{Sign}_{sk}(x)$, we have

$$\text{Verify}_{pk}(x, s) = 1.$$

For security, we define the commonly used notion of existential unforgeability against chosen message attack (EUF-CMA). In the security game, an adversary \mathcal{A} attempts to forge a signature; that is, without having access to sk , provide (x^*, s^*) that are accepted by the verification algorithm. As the name suggests, the adversary gets to choose the messages for which they try to forge a signature. Moreover, the adversary gets access to a signing oracle SignO that they may use to sign any (polynomially bounded) number of messages. Obviously, we require that the adversary has not used SignO to sign x^* .

Definition 2.24 (EUF-CMA security). Let Σ be a signature scheme, defined as in 2.22. Σ is said to be existentially unforgeable against chosen message attack if the advantage of any PPT adversary \mathcal{A} is negligible in λ in the following security game:

EUF-CMA $_{\Sigma, \mathcal{A}}(1^\lambda)$	SignO(x)
$S := \emptyset$	if $S[x] = \perp$ then
$(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$	$s \leftarrow \text{Sign}_{sk}(x)$
$(x^*, s^*) \leftarrow \mathcal{A}^{\text{SignO}}(pk)$	$S[x] := s$
if $(\text{Verify}_{pk}(x^*, s^*) = 1) \wedge (S[x^*] = \perp)$ then	return $S[x]$
return 1 // \mathcal{A} wins	
return 0	

In the above, S represents a table consisting of (key, value) pairs. $S[x] = \perp$ means that S does not yet have a value associated with the key x .

2.3 Lattices

Before talking about lattice-based cryptography it is necessary to formally define the concept of lattices. We will introduce some classic results and some common computational problems; the latter will serve as the foundation of lattice-based cryptography.

2.3.1 Definition and important results

In this section we will go over some of the basic theory of lattices. We will omit the proofs of our statements since they are considered folklore; we refer the interested reader to [34], [35] or [36]. Our presentation will mostly follow the first one.

Let us begin by giving a formal definition of a lattice.

Definition 2.25 (Lattice). Let $m, n \in \mathbb{N}$ such that $m \leq n$. Furthermore, let $\{\mathbf{b}_1, \dots, \mathbf{b}_m\}$ be a linearly independent set of vectors in \mathbb{R}^n and define

$$\mathbf{B} = [\mathbf{b}_1 \ \dots \ \mathbf{b}_m] \in \mathbb{R}^{n \times m}.$$

Then, the additive subgroup of $(\mathbb{R}^n, +)$ defined as

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{ \mathbf{B}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^m \}$$

is called a *lattice*.

We say that \mathbf{B} is the basis of \mathcal{L} , and \mathcal{L} is said to be generated by the matrix \mathbf{B} . n is called the *dimension* of the lattice and m is called its *rank*. If $m = n$, the lattice is said to be *full rank*.

Note that the basis \mathbf{B} is never unique. In fact, it is easy to verify that lattices are free \mathbb{Z} -modules; as such, Lemma 2.6 implies that right-multiplying \mathbf{B} by any integer matrix with determinant in $\{1, -1\}$ gives another basis of the same lattice.

A commonly used invariant of a lattice is its determinant. It measures the volume of the *fundamental parallelepiped* of the lattice, that is, the set $\{ \mathbf{B}\mathbf{x} \mid \mathbf{x} \in [0, 1)^m \}$.

Definition 2.26 (Determinant). Use the same definitions as in Definition 2.25. The determinant of \mathcal{L} is defined as

$$\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}^T \mathbf{B})}.$$

For full-rank lattices, this is equivalent to $\det(\mathcal{L}) = |\det(\mathbf{B})|$.

Shifting each element of a lattice by some fixed vector, we obtain a *lattice coset*.

Definition 2.27 (Lattice coset). Let \mathcal{L} be a lattice of dimension n and $\mathbf{c} \in \mathbb{R}^n$. We define the corresponding lattice coset

$$\mathcal{L} + \mathbf{c} = \{ \mathbf{x} + \mathbf{c} \mid \mathbf{x} \in \mathcal{L} \}.$$

Let us next introduce the concept of *successive minima*; it builds on top of the definition of a (closed) ball.

Definition 2.28 (Ball). For $\mathbf{c} \in \mathbb{R}^n$ and $r, p \in \mathbb{R}$ such that $r > 0$ and $p \geq 1$, we define the open ball of radius r and center \mathbf{c} (with respect to a norm ℓ^p) as

$$\mathcal{B}_r^{(p)}(\mathbf{c}) = \{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{c}\|_p < r \}.$$

Correspondingly, for $r \geq 0$ and p, \mathbf{c} as before we define the closed ball

$$\overline{\mathcal{B}}_r^{(p)}(\mathbf{c}) = \{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{c}\|_p \leq r \}.$$

We take $p = 2$ when omitted.

Definition 2.29 (Successive minima). Use the same definitions as in Definition 2.25, and in addition let $p \geq 1$. For $k \in \{1, \dots, m\}$ we define the k th successive minimum λ_k of the lattice \mathcal{L} (with respect to a norm ℓ^p) to be the minimal r such that $\overline{\mathcal{B}}_r^{(p)}(0)$ contains at least k linearly independent (non-zero) lattice vectors. More formally,

$$\lambda_k^{(p)}(\mathcal{L}) = \min \left\{ r \in \mathbb{R} \mid \text{rank}(\text{span}(\overline{\mathcal{B}}_r^{(p)}(0) \cap \mathcal{L})) \geq k \right\}.$$

Remark 2.30. Some authors choose to replace the minimum by the infimum and the closed ball by open ball in the above definition. However, both definitions are equivalent since \mathcal{L} is always a discrete subset of \mathbb{R}^n , or equivalently $\mathcal{L} \cap \overline{\mathcal{B}}_r^{(p)}(0)$ is finite (see Theorem 6.1 of [36] for instance). We find that the one used here emphasizes the fact that the successive minima are actually achieved, i.e. for all k there exists $\mathbf{x} \in \mathcal{L}$ such that $\|\mathbf{x}\|_p = \lambda_k^{(p)}(\mathcal{L})$, as stated in Claim 7 of [37].

We are especially interested in two of the successive minima: $\lambda_1(\mathcal{L})$, which is the length of the shortest non-zero vector in \mathcal{L} and $\lambda_k(\mathcal{L})$, which is the maximum of the lengths of the shortest independent set of vectors.

The following theorem is closely related to the successive minima of a lattice. It was introduced by Minkowski in 1896.

Theorem 2.31 (Minkowski's Theorem). *Let \mathcal{L} be a full-rank lattice of dimension n and $X \subseteq \mathbb{R}^n$ be a convex subset that is symmetric around the origin. Denote the volume of X as $\text{vol}(X)$; if*

$$\text{vol}(X) > 2^n \det(\mathcal{L}),$$

X contains at least one non-zero point of \mathcal{L} .

The following corollary gives an upper bound on the shortest vector of the lattice.

Corollary 2.32. *Let \mathcal{L} be a full-rank lattice of dimension n . Then,*

$$\lambda_1^\infty(\mathcal{L}) \leq \det(\mathcal{L})^{\frac{1}{n}}.$$

and by equivalence of norms,

$$\lambda_1^2(\mathcal{L}) \leq \sqrt{n} \cdot \det(\mathcal{L})^{\frac{1}{n}}.$$

2.3.2 Worst-case computational problems

There exist a number of presumed hard computational problems over lattices. For our purposes, these are important in the sense that the security of lattice-based cryptography is based on their hardness. We introduce a few common problems that are relevant to this work, but note that there are several others as well.

The problems come in two variants: the exact version that asks to find the best possible solution, and the approximate version that asks to find a solution that is nearly as good as the best possible one. The slack given for the approximation is characterized by the *approximation factor* $\mu \geq 1$. For conciseness, we only write down the approximate versions; the exact versions are obtained by setting $\mu = 1$.

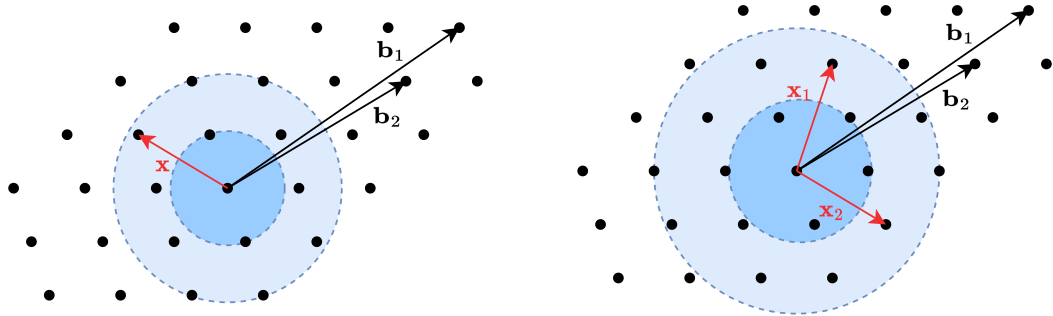
Definition 2.33 ((Approximate) Shortest vector problem (SVP_μ)). Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$ as input, SVP_μ asks to find $\mathbf{x} \in \mathcal{L} \setminus \{0\}$ satisfying

$$\|\mathbf{x}\| \leq \mu \cdot \lambda_1(\mathcal{L}).$$

Definition 2.34 ((Approximate) Shortest independent vectors problem (SIVP_μ)). Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$ as input, SIVP_μ asks to find $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{L} \setminus \{0\}$ satisfying

$$\max_i \{\|\mathbf{x}_i\|\} \leq \mu \cdot \lambda_m(\mathcal{L}).$$

Both SVP and SIVP are illustrated in Figure 1. In addition to them, we also consider the *closest vector problem*.



(a) Approximate SVP: find a lattice vector inside the blue region. \mathbf{x} represents one possible solution.

(b) Approximate SIVP: find two linearly independent vectors inside the blue region. $\{\mathbf{x}_1, \mathbf{x}_2\}$ represents one possible solution.

Figure 1: Approximate SVP and SIVP over the same two-dimensional lattice, given by the basis $\{\mathbf{b}_1, \mathbf{b}_2\}$. The deep blue region represents the exact version ($\gamma = 1$). In the approximate version we allow some slack, i.e., the light blue region.

Definition 2.35 ((Approximate) Closest vector problem (CVP_μ)). Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$ and a target vector $\mathbf{t} \in \mathbb{R}^n$ as input, CVP_μ asks to find $\mathbf{x} \in \mathcal{L} \setminus \{0\}$ satisfying

$$\|\mathbf{x} - \mathbf{t}\| \leq \mu \cdot \min_{\mathbf{y} \in \mathcal{L}} \{\|\mathbf{y} - \mathbf{t}\|\}.$$

Solving any of the three problems (in the worst case) with $\mu = \text{poly}(m)$ is a long-standing open problem. Since this is the case for both classical and quantum algorithms, it is conjectured that the problems are hard for both classical and quantum adversaries. This assumption forms the basis for the security of lattice-based cryptography.

Let us introduce one more problem that will turn out to be useful in Section 4, namely Hermite SVP. It is a variant of SVP that uses the determinant of the lattice to bound the length of the vector.

Definition 2.36 ((Approximate) Hermite shortest vector problem (HSVP $_\mu$)). Given a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$ as input, HSVP $_\mu$ asks to find $\mathbf{x} \in \mathcal{L} \setminus \{0\}$ satisfying

$$\|\mathbf{x}\| \leq \mu \cdot \sqrt{m} \cdot \det(\mathcal{L})^{\frac{1}{m}}.$$

The problem is well-posed for any $\mu \geq 1$ by Theorem 2.31. In Section 2.4.8 we state the problem for ideal lattices.

2.3.3 Average-case computational problems

Worst-case computational problems are not useful for building cryptographic primitives: in contrast to having one hard problem instance, we would like the guarantee of having a distribution of exponentially many hard instances. Such problems are called average-case problems. The field of lattice-based cryptography emerged in 1996 when Miklós Ajtai introduced the first such problem; namely, Short Integer Solution (SIS) [8]. He also showed that the average-case of SIS is at least as hard as the worst case of SIVP.

In the following, we assume that the parameters of the problems are functions of the security parameter λ .

Definition 2.37 (SIS). Let $n, m, q, \beta \in \mathbb{N}$ such that $m \geq n$ and $q \geq 2$. Given random $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and some target vector $\mathbf{v} \in \mathbb{Z}_q^n$, the SIS $_{n,m,q,\beta,\mathbf{v}}$ problem asks to find $\mathbf{u} \in \mathbb{Z}^m$ satisfying

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \pmod{q} \quad \text{and} \quad 0 < \|\mathbf{u}\| \leq \beta.$$

Observe that the first condition is easy to satisfy; one can simply use well-established tools from linear algebra to solve the linear system of equations. However, requiring the norm to be bounded from above is what makes the problem difficult.

Another fundamental problem in lattice-based cryptography is the Learning with Errors (LWE) problem, introduced by Oded Regev in 2005 [9]. It comes in two variants: the search version, which asks to learn the coefficients of a linear function given some noisy evaluations, and the decision version, which asks to distinguish if some samples are said noisy evaluations or uniform samples.

Definition 2.38 (LWE, Search). Let $n, m, q \in \mathbb{N}$ such that $m \geq n$ and $q \geq 2$, and χ be a distribution over \mathbb{Z} . Let $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{e} \leftarrow \chi^m$ and $\mathbf{s} \in \mathbb{Z}_q^n$. Furthermore, define

$$\mathbf{b}^T = \mathbf{s}^T \mathbf{A} + \mathbf{e}^T \pmod{q}.$$

Given (\mathbf{A}, \mathbf{b}) the sLWE $_{n,m,q,\chi}$ problem asks to recover \mathbf{s} .

Definition 2.39 (LWE, Decision). Let \mathbf{b} be either generated as in Definition 2.38 or sampled uniformly in \mathbb{Z}_q^m . Given \mathbf{b} , the $\text{dLWE}_{n,m,q,\chi}$ asks to distinguish how it was generated.

2.3.4 SIS trapdoors

Let \mathbf{A}, q be as in Definition 2.37. We define the corresponding SIS function as

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q.$$

By the SIS assumption, $f_{\mathbf{A}}$ is hard to invert for uniformly sampled $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. In other words: given $\mathbf{v} \in \mathbb{Z}_q^n$, it is hard to find \mathbf{u} , a (non-zero) preimage of \mathbf{v} under $f_{\mathbf{A}}$. SIS trapdoors leverage this assumption; the idea behind them is to generate a specially crafted matrix \mathbf{A} that is seemingly uniform (that is, either statistically close or computationally indistinguishable) but has a trapdoor $\text{td}_{\mathbf{A}}$ that allow one to invert $f_{\mathbf{A}}$ efficiently.

There are several different constructions of SIS trapdoors, but one of the most widely used is the one proposed by Gentry, Peikert, and Vaikuntanathan in [38]. Their approach is to choose $\text{td}_{\mathbf{A}}$ to be a matrix \mathbf{T} that is a good basis of the *kernel lattice* of $f_{\mathbf{A}}$, i.e.

$$\{\mathbf{x} \in \mathbb{Z}^m \mid f_{\mathbf{A}}(\mathbf{x}) = 0\}.$$

“Good” basis here refers to $\|\mathbf{T}\|_{\text{GS}}$ being small; the reason for this is the following theorem.

Theorem 2.40 (Theorem 4.1 of [38]). *There is a PPT algorithm that, given a basis \mathbf{B} of an n -dimensional full-rank lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, a parameter $s \geq \|\mathbf{B}\|_{\text{GS}} \cdot \omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^n$, outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\mathcal{L},s,\mathbf{c}}$.*

As a consequence, given \mathbf{T} , one can efficiently sample preimages of the function $f_{\mathbf{A}}$ such that the size of the preimages is governed by $\|\mathbf{T}\|_{\text{GS}}$. To see why, suppose that we are given a target \mathbf{v} ; using linear algebra, one can find arbitrary (not necessarily short) \mathbf{u}' such that $f_{\mathbf{A}}(\mathbf{u}') = \mathbf{v} \bmod q$. The norm of \mathbf{u}' can then be reduced using Theorem 2.40. Let $s \in \mathbb{R}$ be as in the theorem and sample \mathbf{w} from $\mathcal{D}_{\mathcal{L}(\mathbf{T}),s,\mathbf{u}'}$. Then, let $\mathbf{u} = \mathbf{u}' - \mathbf{w}$. It is a preimage of \mathbf{v} since $\mathbf{A}\mathbf{u} = \mathbf{A}\mathbf{u}' - \mathbf{A}\mathbf{w} = \mathbf{v} \bmod q$. Moreover, \mathbf{u} is sampled from a distribution statistically close to $\mathcal{D}_{\mathcal{L}(\mathbf{T})+\mathbf{u}',s,0}$ and hence the expected $\|\mathbf{u}\|$ is determined by $s \geq \|\mathbf{B}\|_{\text{GS}} \cdot \omega(\sqrt{\log n})$.

We call the preimage sampling algorithm described above *GPV sampling*.

2.4 Algebraic number theory

A multitude of cryptographic primitives can already be built from SIS and LWE. However, such applications do have some limitations. First, simply writing down the matrix \mathbf{A} requires $nm \log q$ bits. This means that the running time of the applications is at least quadratic in n , which is often undesirable in practice. Secondly, the only units in \mathbb{Z} are 1 and -1 which is not convenient for constructing e.g. proof systems. A

common solution to both of these problems is to replace \mathbb{Z} by a more complicated ring, and natural choices for the ring arise from algebraic number theory.

In this section, we aim to cover the necessary prerequisites on number fields and their rings of integers. We also discuss the ideals of these rings and demonstrate how they can be viewed as lattices, introducing the concept of ideal lattices. Finally, we introduce some standard worst-case and average-case problems over these lattices.

When presenting results about number fields, we follow [36] for the most part. All of the missing proofs can also be found there.

2.4.1 Field extensions

Let us begin with a brief recap of field extensions. We will provide some related terminology and notation. Note that, for the sake of conciseness, the definitions and results may not be presented in full generality. In particular, we pay special attention to finite Galois extensions since that will be sufficient for our purposes. For a more general approach, we refer the reader e.g. to [26].

Definition 2.41 (Field extension). Let K, L be two fields such that K is a subfield of L . Then:

- L is said to be an extension field of K . We say that L and K form a field extension and denote it as L/K .
- L can be viewed as a vector space over K ; the dimension of this vector space is denoted $[L : K]$ and called the *degree of L (over K)*. A K -basis (or simply a basis when the ground field is clear from the context) of L refers to a basis of this vector space.
- If $[L : K]$ is finite, we say that L is a *finite extension* (of K).
- For $x_1, \dots, x_n \in L$, we write $K(x_1, \dots, x_n)$ to denote the smallest subfield of L containing $K \cup \{x_1, \dots, x_n\}$.
- We say that an element $x \in L$ is algebraic over K if there exists non-zero polynomial $p \in K[X]$ such that $p(x) = 0$. If all elements of L are algebraic over K , L is said to be an *algebraic extension* over K .

Fact 2.42. Let L/K be a field extension. If L is a finite extension, then it is also algebraic.

The *minimal polynomial* is a central tool in the study of field extensions.

Definition 2.43 (Minimal polynomial). Let L/K be a field extension and let $x \in L$. Suppose p is the lowest degree monic polynomial in $K[X]$ satisfying $p(x) = 0$. Then, we say that p is the minimal polynomial of x with respect to K and denote $p = \text{Irr}(x, K, X)$.

If the extension is not algebraic, x might not be a root of any polynomial in $K[X]$. However, if such a polynomial exists, the minimal polynomial is unique. This implies that the minimal polynomial is well-defined.

We pay special attention to certain types of extensions: normal, separable and Galois.

Definition 2.44 (Normal, separable and Galois extension). Let L/K be a finite (and thus algebraic) extension. Then,

- If every irreducible polynomial of $K[X]$ that has a root in L splits into linear factors in L , we say that the extension is *normal*.
- For $x \in L$, we say that x is *separable* if $\text{Irr}(x, K, X)$ does not have repeated roots. If this holds for all $x \in L$, we say that the extension is separable.
- If an extension is normal and separable, we say that it is *Galois*.

For Galois extensions, the *Galois group* is defined as the set of automorphisms of the extension field that fix every element of the subfield.

Definition 2.45. Let L/K be a Galois extension. Then, the set of field automorphisms of L that fix K pointwise form a group under composition. The group is called the Galois group of L over K and is denoted as $\text{Gal}(L/K)$.

The order of the Galois group is equal to the degree of the field extension.

Lemma 2.46. Let L/K be a Galois extension; then, $|\text{Gal}(L/K)| = [L : K]$.

We frequently utilize *field norm* that maps an element of a finite extension onto a subfield. For Galois extensions, the definition is particularly straight-forward.

Definition 2.47 (Field norm for Galois extensions). Let L/K be a finite Galois extension. We define the field norm as

$$\begin{aligned} \mathcal{N}_{L/K} : L &\rightarrow K \\ x &\mapsto \prod_{\sigma_i \in \text{Gal}(L/K)} \sigma_i(x). \end{aligned}$$

Suppose L is an extension of K and M is an extension of L . This kind of sequence of field extension is called a *tower of fields*. We say that the field norm is *transitive* in towers, i.e., the norm $\mathcal{N}_{M/K}$ can be written as the composition of $\mathcal{N}_{L/K}$ and $\mathcal{N}_{M/L}$.

Lemma 2.48 (Transitivity of the field norm). Let $K \subseteq L \subseteq M$ be a tower of fields where each extension is finite and Galois. Then,

$$\mathcal{N}_{M/K} = \mathcal{N}_{L/K} \circ \mathcal{N}_{M/L}.$$

2.4.2 Number fields

Next, let us restrict ourselves to studying extensions of the rational field \mathbb{Q} . First, we introduce the set of *algebraic numbers*.

Definition 2.49 (Algebraic numbers). The elements of \mathbb{C} that are algebraic over \mathbb{Q} are called algebraic numbers and denoted as \mathbb{A} .

\mathbb{A} forms a subfield of \mathbb{C} and it is a useful tool in theory. However, it is often not that interesting in practice since it is an infinite extension of \mathbb{Q} . On the contrary, finite extensions of \mathbb{Q} are often of greater interest.

Definition 2.50 (Number field). If $K \subseteq \mathbb{C}$ is a finite extension of \mathbb{Q} , it is called a *number field*. The degree $[K : \mathbb{Q}]$ is simply referred to as the *degree* of the number field.

For a number field K , the monomorphisms $\sigma_i : K \rightarrow \mathbb{C}$ play a significant role in the theory.

Theorem 2.51. *Let K be a number field of degree d . Then, there exist exactly d distinct monomorphisms $\sigma_i : K \rightarrow \mathbb{C}$.*

If the extension K/\mathbb{Q} is Galois, the monomorphisms σ_i coincide with the Galois group $\text{Gal}(K/\mathbb{Q})$. Even when this is not the case, the field norm $\mathcal{N}_{K/\mathbb{Q}}$ is given by the product of the monomorphisms.

Definition 2.52. Use the same definitions as in Theorem 2.51. For $x \in K$, we define the field norm of x (over \mathbb{Q}) as

$$\mathcal{N}_{K/\mathbb{Q}}(x) = \prod_{i \in [d]} \sigma_i(x).$$

We often simply denote the above as $\mathcal{N}(x)$.

On a more technical side, we need to provide a measure for the number of bits needed to represent elements of a number field. Thus, we define the *size* of elements (with respect to a certain basis). Our definitions follow those of [18].

Definition 2.53. Let $x = x_1/x_2 \in \mathbb{Q}$, where $x_1, x_2 \in \mathbb{Z}$ and $\text{gcd}(x_1, x_2) = 1$. We define

$$\text{size}(x) = 1 + \log|x_1| + \log|x_2|.$$

More generally, let K be a number field and $\{b_i\}_i$ be a \mathbb{Q} -basis of K . For $x = \sum_i x_i b_i \in K$ (where $x_i \in \mathbb{Q}$), we define

$$\text{size}(x) = \sum_i \text{size}(x_i).$$

2.4.3 Rings of integers

The field of algebraic numbers has a special subring, called the ring of *algebraic integers*.

Definition 2.54 (Algebraic integers). Let $x \in \mathbb{C}$; we say that x is an algebraic integer if there exists a non-zero monic polynomial $p \in \mathbb{Z}[X]$ such that $p(x) = 0$. We denote the set of all algebraic integers as \mathbb{B} .

For any number field K , $K \cap \mathbb{B}$ is a subring of K ; we call it the *ring of integers* of K . This definition also motivates the concept *integral bases*.

Definition 2.55 (Integral basis). Let K be a number field of degree d and \mathcal{R} be its ring of integers. We say that $B = \{b_1, \dots, b_d\} \subseteq K$ is an integral basis (or a \mathbb{Z} -basis for short) of K if

- (i) $b_1, \dots, b_d \in \mathcal{R}$, and
- (ii) all $r \in \mathcal{R}$ can be uniquely expressed as $r = \sum_{i=1}^d a_i b_i$ where $a_1, \dots, a_d \in \mathbb{Z}$.

Although it may not be obvious, every number field possesses a \mathbb{Z} -basis. All integral bases of K are equivalent up to a unimodular transformation (see Lemma 2.6). Therefore, the *discriminant* of a number field is well-defined.

Definition 2.56. Let K be a number field of degree d , $\{b_1, \dots, b_d\}$ be a \mathbb{Z} -basis of K and $\sigma_i : K \rightarrow \mathbb{C}$ for $i \in [d]$ be the monomorphisms. The discriminant of K is defined as

$$\Delta_K = \left(\det \begin{bmatrix} \sigma_1(b_1) & \cdots & \sigma_1(b_d) \\ \vdots & \ddots & \vdots \\ \sigma_d(b_1) & \cdots & \sigma_d(b_d) \end{bmatrix} \right)^2.$$

2.4.4 Ideals

Let us next define the concept of fractional ideals of a number field, along with some related language.

Definition 2.57 (Fractional ideal). Let K be a number field and \mathcal{R} be its ring of integers. Then:

- A subset of K is called a *fractional ideal* if it is of form cI where $c \in K$ and I is an ideal of \mathcal{R} . Ideals of \mathcal{R} are called *integral ideals*.
- An integral ideal \mathfrak{p} is said to be *prime* if $IJ \subseteq \mathfrak{p}$ implies either $I \subseteq \mathfrak{p}$ or $J \subseteq \mathfrak{p}$ for all integral ideals I, J .
- For $x \in K$, we denote the principal fractional ideal generated by x as $\langle x \rangle$, i.e., $\langle x \rangle = \{rx \mid r \in \mathcal{R}\}$. More generally, we denote

$$\langle x_1, \dots, x_n \rangle = \left\{ \sum_{i \in [n]} r_i x_i \mid r_i \in \mathcal{R} \forall i \in [n] \right\}.$$

- We define the *norm* of an integral ideal $I \subseteq \mathcal{R}$ as $\mathcal{N}(I) = |\mathcal{R}/I|$. More generally, for fractional ideal $J = cI$ we define $\mathcal{N}(J) = \mathcal{N}(c) \cdot \mathcal{N}(I)$, where $\mathcal{N}(c)$ denotes the field norm of c .

The following theorem summarizes four fundamental properties of fractional ideals.

Theorem 2.58. *Let K, \mathcal{R} be as in Definition 2.57; furthermore, let I be a fractional ideal of K . Then:*

- I has an integral basis (or, a \mathbb{Z} -basis); that is, a \mathbb{Q} -linearly independent set of elements of K that span the ideal as a \mathbb{Z} -module.
- If I is non-zero, it has an inverse; that is, a fractional ideal I^{-1} which satisfies $II^{-1} = \mathcal{R}$.
- I is generated by two elements, i.e., there exist $x_1, x_2 \in K$ such that $I = \langle x_1, x_2 \rangle$.
- I can be uniquely (up to the order of the factors) written as

$$I = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_n^{e_n},$$

where $n \in \mathbb{N}$, $e_i \in \mathbb{Z}$ and \mathfrak{p}_i is a prime ideal for all $i \in [n]$; moreover, the exponents e_i are non-negative if and only if I is integral.

We call the product of the fourth property the *prime decomposition* of a fractional ideal. If two integral ideals do not share any prime factors, we say that they are coprime.

Lemma 2.59. *Use the same definitions as in Theorem 2.58. I can be uniquely written as $I_1 I_2^{-1}$ where I_1, I_2 are coprime integral ideals.*

Proof. Use the prime decomposition of I : let I_1 be the product of the terms with a non-negative exponents in the decomposition and I_2^{-1} be the product of the rest. \square

Recall the size of field elements from Definition 2.53. Lemma 2.59 allows defining a natural extension for ideals.

Definition 2.60. Let a fractional ideal $I = I_1 I_2^{-1}$ as in Lemma 2.59; we define

$$\text{size}(I) = \log \mathcal{N}(I_1) + \log \mathcal{N}(I_2).$$

The two-element representation of a fractional ideal (see the third property of Theorem 2.58) is often useful. However, integral ideals also have another convenient representation. For an extended version of the following result, as well as a constructive proof, see Lemma A.1.

Lemma 2.61. *Let K be a number field with the ring of integers \mathcal{R} . Then, any integral ideal $I \subseteq \mathcal{R}$ can be written as*

$$I = \langle z \rangle \cap \mathcal{R}$$

for some $z \in K$.

We finish this section by proving a series of results that will be needed later in the thesis.

Lemma 2.62. *Let I, J be fractional ideals of a number field K . Then, $(I \cap J)^{-1} = I^{-1} + J^{-1}$.*

Proof. Write I, J using their prime decompositions; i.e., let $I = \prod_i \mathfrak{p}_i^{e_i}$ and $J = \prod_i \mathfrak{p}_i^{f_i}$ where $e_i, f_i \in \mathbb{Z}$ for all i . We have

$$(I \cap J)^{-1} = \left(\prod_i \mathfrak{p}_i^{\max\{e_i, f_i\}} \right)^{-1} = \prod_i \mathfrak{p}_i^{-\max\{e_i, f_i\}} = \prod_i \mathfrak{p}_i^{\min\{-e_i, -f_i\}} = I^{-1} + J^{-1},$$

where the first and last equalities follow from Lemma 5.8 of [36]. \square

Lemma 2.63. *Let I be an integral ideal of \mathcal{R} . Then, $\mathcal{N}(I) \in I$ and as a corollary, $\mathcal{N}(I) \cdot I^{-1} \subseteq \mathcal{R}$.*

Proof. For the former part, see e.g. Theorem 5.14 of [36]. For the latter part let us assume the opposite; that is, there exists $x \in I^{-1}$ such that $\mathcal{N}(I)x \notin \mathcal{R}$. Applying the first part, this contradicts $II^{-1} = \mathcal{R}$. \square

Lemma 2.64. *Let $z \in K$. Then, for all $k \in \mathbb{N}$,*

$$\langle z^{k+1} \rangle \cap \mathcal{R} \subseteq \langle z^k \rangle \cap \mathcal{R}.$$

Proof. Let $x \in \langle z^{k+1} \rangle \cap \mathcal{R}$, i.e. $x \in \mathcal{R}$ and there exists $r \in \mathcal{R}$ such that $x = z^{k+1}r$. We want to show that $x \in \langle z^k \rangle \cap \mathcal{R}$. Since $x \in \mathcal{R}$, it suffices to prove that $zr \in \mathcal{R}$ because this implies $x = z^k(zr) \in \langle z^k \rangle$.

Let I, J be coprime integral ideals such that $\langle z \rangle = IJ^{-1}$ (recall Lemma 2.59). Since $x \in \mathcal{R}$, we have

$$\langle x \rangle = I^{k+1}J^{-(k+1)}\langle r \rangle \subseteq \mathcal{R}.$$

By the coprimeness of I, J this implies $J^{-(k+1)}\langle r \rangle \subseteq \mathcal{R}$, and multiplying this by $IJ^k \subseteq \mathcal{R}$ we obtain $IJ^{-1}\langle r \rangle = \langle zr \rangle \subseteq \mathcal{R}$ which is equivalent to $zr \in \mathcal{R}$. \square

Lemma 2.65. *Let $z \in K$. Then, for all $k \in \mathbb{N}$,*

$$(\langle z \rangle \cap \mathcal{R})^k \subseteq \langle z^k \rangle \cap \mathcal{R}.$$

Proof. Applying Lemma 2.62 we obtain

$$(\langle z \rangle \cap \mathcal{R})^{-k} = \left(\langle z^{-1} \rangle + \mathcal{R} \right)^k = \langle z^{-k} \rangle + \dots + \mathcal{R} \supseteq \langle z^{-k} \rangle + \mathcal{R} = (\langle z^k \rangle \cap \mathcal{R})^{-1},$$

and the claim follows after taking the inverse of both sides. \square

2.4.5 Example: Cyclotomic fields

To illustrate the concepts covered so far, let us look at a special case of number fields; namely, *cyclotomic fields*. They are important in practice since they are well understood and possess nice computational properties. Therefore, many cryptographic schemes have been instantiated over cyclotomic fields in particular. In addition to this, they are important in theory; historically, they played a major role in the proof of Fermat's Last Theorem, for instance.

Cyclotomic fields are obtained by adjoining a *primitive k th root of unity*, $\zeta_k = e^{2\pi i/k}$, to the field of rationals.

Definition 2.66 (Cyclotomic fields). Let $k \in \mathbb{N}$ and define $\zeta = \zeta_k$. We call $K(\zeta)$ the k th cyclotomic field. The degree of $K(\zeta)$ over \mathbb{Q} is given by the Euler's totient function: $[K(\zeta) : \mathbb{Q}] = \phi(k) = |\{a \in [k] \mid \gcd(a, k) = 1\}|$.

Cyclotomic fields have particularly simple expressions for the ring of integers and the \mathbb{Z} -basis.

Lemma 2.67. *Use the same definitions as in Definition 2.66; furthermore, denote $d = \phi(k)$. The ring of integers of K is given by $\mathbb{Z}[\zeta] = \{ \sum_{i=0}^{d-1} a_i \zeta^i \mid a_i \in \mathbb{Z} \forall i \}$, and $\{1, \zeta, \dots, \zeta^{d-1}\}$ is an integral basis of K .*

Observe that, for a cyclotomic field $K = \mathbb{Q}(\zeta)$, the complex conjugation corresponds to the automorphism defined by the extension of $\zeta \mapsto \zeta^{-1}$. Thus, for $x \in K$, we will write \bar{x} to denote the conjugate element in K . In particular, this means that we can inherit the standard inner product from \mathbb{C}^n to K^n without ambiguity. This will be useful in Section 5 where we consider GSO over K^n .

Lastly, we consider an important special case. In sections 5 and 6, we focus on the case where k is a power of two; such fields are simply called *power-of-2 cyclotomic fields*. They form a tower of fields,

$$\mathbb{Q} \subseteq \mathbb{Q}(\zeta_4) \subseteq \mathbb{Q}(\zeta_8) \subseteq \dots,$$

where each field is a Galois extension of the previous ones.

2.4.6 Splitting of primes

For completeness, we will briefly discuss the splitting behavior of primes in cyclotomic fields since it is often important for parameter selection in applications. Let $q \in \mathbb{N}$ be a prime; note that this does not imply that the ideal $\langle q \rangle$ is prime in the ring of integers \mathcal{R} of a number field K . Instead, $\langle q \rangle$ might factorize further as a product of prime ideals. When $\langle q \rangle$ has $[K : \mathbb{Q}]$ distinct prime factors, we say that q *splits completely*. Characterizing such primes is easy for cyclotomic fields.

Lemma 2.68 (Part of Theorem 2.13 of [39]). *Let $k \in \mathbb{N}$, $K = \mathbb{Q}(\zeta_k)$, $d = \phi(k)$ and $q \in \mathbb{N}$ be a prime. Then, the principal ideal $\langle q \rangle$ can be written as*

$$\langle q \rangle = \mathfrak{p}_1 \cdots \mathfrak{p}_d$$

where \mathfrak{p}_i are prime ideals in the ring of integers of K if and only if $q \equiv 1 \pmod{k}$.

Let us adopt the notation of Lemma 2.68 and let \mathcal{R} be the ring of integers of K . Suppose that q is chosen such that $q \equiv 1 \pmod{k}$; by Dirichlet's theorem on arithmetic progressions, there exist infinitely many such q . This choice is often used in practice, since by the Chinese remainder theorem we have

$$\mathcal{R}_q = \mathcal{R}/\langle q \rangle \cong (\mathcal{R}/\mathfrak{p}_1) \times \cdots \times (\mathcal{R}/\mathfrak{p}_d).$$

This identification allows more efficient multiplication in \mathcal{R}_q . Moreover, the units of \mathcal{R}_q can be easily characterized as

$$\mathcal{R}_q^\times \cong (\mathcal{R}/\mathfrak{p}_1)^\times \times \cdots \times (\mathcal{R}/\mathfrak{p}_d)^\times.$$

Since \mathfrak{p}_i are non-zero prime ideals, $\mathcal{R}/\mathfrak{p}_i$ is a field for all i (see, e.g., Lemma 5.1 and Theorem 5.3 of [36]). This often implies that a significant proportion of the elements of \mathcal{R}_q are units.

2.4.7 Embeddings and ideal lattices

In this section, we once again consider general number fields. We examine how the fractional ideals of a number field of degree d correspond to d -dimensional full-rank lattices in a real vector space; such lattices are called *ideal lattices*. This identification is done via mappings that we call *embeddings*. The two commonly used embeddings are called the *coefficient embedding* and the *canonical embedding*. We start with the coefficient embedding as it is somewhat simpler conceptually.

Definition 2.69 (Coefficient embedding). Let K be a number field of degree d and $B = \{b_i\}_{i \in [d]}$ be an integral basis of K . We define the *coefficient embedding* of K (with respect to the integral basis) as the mapping

$$\begin{aligned} \tau : K &\rightarrow \mathbb{R}^d \\ x &\mapsto (x_1, \dots, x_d) \end{aligned}$$

where the x_i satisfy $x = \sum_{i \in [d]} x_i b_i$. Note that the mapping is unique due to properties of the \mathbb{Z} -basis. The ℓ_2 -norm of x under the coefficient embedding is denoted and defined as

$$\|x\| = \|\tau(x)\|,$$

and similarly for other norms – most importantly, $\|\cdot\|_\infty$.

We extend the notations coefficient-wise to elements of K^n ; that is, for $\mathbf{y} = (y_1, \dots, y_n) \in K^n$, we define

$$\tau(\mathbf{y}) = \begin{bmatrix} \tau(y_1) \\ \vdots \\ \tau(y_n) \end{bmatrix}$$

and $\|\mathbf{y}\| = \|\tau(\mathbf{y})\|$.

The following proposition demonstrates how the ideals of K correspond to lattices under the mapping τ .

Proposition 2.70. *Use the same definitions as in Definition 2.69; moreover, let I be a fractional ideal of K . Then, $\tau(I)$ is a full-rank lattice of dimension d .*

Proof. First, observe that τ is a linear mapping between vector spaces; that is, it respects addition and scalar multiplication. For the first one, notice that for $x = \sum_{i \in [d]} x_i b_i$ and $y = \sum_{i \in [d]} y_i b_i$ we have

$$\tau(x + y) = (x_1 + y_1, \dots, x_d + y_d) = \tau(x) + \tau(y).$$

For the second one, we can similarly show how $\tau(cx) = c\tau(x)$ for all $c \in \mathbb{Q}$ and $x \in K$.

Now, let $\{c_i\}_{i \in [d]}$ be a \mathbb{Z} -basis of I . This implies

$$I = \left\{ \sum_{i \in [d]} a_i c_i \mid a_i \in \mathbb{Z} \forall i \in [d] \right\}.$$

By the linearity of τ ,

$$\tau(I) = \left\{ \sum_{i \in [d]} a_i \tau(c_i) \mid a_i \in \mathbb{Z} \forall i \in [d] \right\}.$$

Therefore, $\tau(I)$ is a lattice with a basis given by $[\tau(c_1) \ \cdots \ \tau(c_d)] \in \mathbb{R}^{d \times d}$. \square

Since addition under τ is coefficient-wise, we have $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in K$. For multiplication, we can define a field-dependent *expansion factor* γ_K for which we have $\|xy\| \leq \gamma_K \cdot \|x\| \cdot \|y\|$ for all $x, y \in K$ [40].

Under τ , multiplication by a field element can be expressed as a linear transformation. We denote the matrix corresponding to $x \in K$ as $M(x)$. The elements of the matrix are determined by the identity

$$xy = z \Leftrightarrow M(x)\tau(y) = \tau(z) \forall x, y, z \in K.$$

We give the following definition for concreteness.

Definition 2.71. Let K, B be as in Definition 2.69. For $x \in K$, we define the matrix $M(x) \in \mathbb{R}^{d \times d}$ as

$$M(x) = [\tau(xb_1) \ \cdots \ \tau(xb_d)].$$

Similarly as in Definition 2.69, we extend the notation coefficient-wise to vectors in K^n and matrices in $K^{m \times n}$.

The coefficient embedding allows us to perform rounding in the field K , that is, map an arbitrary field element to a nearby element contained in the ring of integers. The concept is a generalization of rounding in \mathbb{Q} .

Definition 2.72 (Rounding). For $x \in \mathbb{Q}$, we call the result of rounding of x to the nearest integer (rounding half towards zero) the *integral part* of x and denote it as $\lfloor x \rfloor$. Furthermore, we denote $\{x\} = x - \lfloor x \rfloor$ and call it the *fractional part* of x .

We extend this notation to arbitrary number fields as follows. Let K, B be as in Definition 2.69 and $x = \sum_{i \in [d]} x_i b_i \in K$. We denote $\lfloor x \rfloor = \sum_{i \in [d]} \lfloor x_i \rfloor b_i$ and $\{x\} = \sum_{i \in [d]} \{x_i\} b_i$ (or equivalently, $\{x\} = x - \lfloor x \rfloor$).

Observe that in the above definition, B being an integral basis guarantees that the integral part of $x \in K$ is in the ring of integers of K . For \mathbb{Q} , the absolute value of the fractional part is guaranteed to be at most $1/2$. More generally, for all $x \in K$ we have $\|\{x\}\|_\infty \leq 1/2$ and, by equivalence of norms, $\|\{x\}\| \leq \sqrt{d}/2$.

Let us then look into the other natural choice of embeddings, the canonical embedding. It is defined using the monomorphisms $\sigma_i : K \rightarrow \mathbb{C}$.

Definition 2.73 (Canonical embedding). Let K be a number field of degree d and $\sigma_1, \dots, \sigma_d$ be the embeddings of K . We define the *canonical embedding* of K as the mapping

$$\begin{aligned} \sigma : K &\rightarrow \mathbb{C}^d \\ x &\mapsto (\sigma_1(x), \dots, \sigma_d(x)). \end{aligned}$$

The ℓ_2 -norm of $x \in K$ under the canonical embedding is given by the standard norm in \mathbb{C}^d (i.e., $\|\sigma(x)\| = \sqrt{\sigma(x)^\dagger \sigma(x)}$), and similarly for the infinity norm.

Remark 2.74. To be strict, observe that σ does not directly map fractional ideals to lattices in \mathbb{R}^d . However, there is a closely connected mapping that does satisfy this property. Let us divide the monomorphisms σ_i into two categories, *real* and *complex*, depending on if $\sigma_i(K) \subseteq \mathbb{R}$ or not. We write $d = s + 2t$ where s is the number of real monomorphisms and $2t$ the number of complex monomorphisms. Since the complex conjugation is an automorphism of \mathbb{C} , the complex monomorphisms come in conjugate pairs. Therefore, there are only t independent complex monomorphisms. If we only choose one of each such pair, we get an embedding $\sigma' : K \rightarrow \mathbb{R}^s \times \mathbb{C}^t \cong \mathbb{R}^d$.

It is easy to check that under σ' , the fractional ideals of K correspond to lattices and hence it is sometimes used as the definition of canonical embedding. However, the two are merely two sides of the same coin; one can easily equip \mathbb{R}^d with a quadratic form that corresponds to $\|\sigma(\cdot)\|$. Therefore, making a distinction is not necessary for our purposes.

The canonical embedding is often easier to analyze and yields better bounds than the coefficient embedding [41]. One reason for this is that using the canonical embedding eliminates the need of an expansion factor. Since multiplication under the canonical embedding is coefficient-wise, we have $\|xy\| \leq \|x\|_\infty \cdot \|y\|$ for all $x, y \in K$.

Similarly to the coefficient embedding, we can upper bound the norm of fractional parts under σ .

Proposition 2.75. Let K be a number field of degree d and $B = \{b_i\}_{i \in [d]}$ be a \mathbb{Z} -basis of K . Then, we have

$$\|\sigma(\{x\})\|_\infty \leq d \cdot \frac{1}{2} \cdot \delta_K,$$

where $\delta_K = \max_i \{\|\sigma(b_i)\|_\infty\}$.

Proof. We have

$$\|\sigma(\{x\})\|_\infty \leq \sum_{i \in [D]} \|\sigma(\{x_i\} b_i)\|_\infty \leq \sum_{i \in [D]} \|\sigma(\{x_i\})\|_\infty \|\sigma(b_i)\|_\infty \leq d \cdot \frac{1}{2} \cdot \delta_K,$$

where the first inequality follows from triangle inequality and the second is a consequence of multiplication under σ being coefficient-wise. \square

Remark 2.76. Throughout the remainder of the thesis, we assume that a \mathbb{Z} -basis of a number field K is always implicitly known and denote the largest element of that basis (measured in infinity norm) as δ_K .

2.4.8 Computational problems over ideal lattices and rings

Now that we have established the concept of ideal lattices, a natural next step is to consider the lattice problems defined in Section 2.3.2 in the special case of ideal lattices. Despite the additional algebraic structure, all of the problems are still commonly assumed to be hard for $\mu = \text{poly}(d)$. We only formally write down the ideal-HSVP problem; the ideal lattice versions of the rest can be defined analogously.

Definition 2.77 ((Approximate) Ideal Hermite shortest vector problem (ideal-HSVP $_{\mu}$, id-HSVP $_{\mu}$)). Let K be a number field; given a non-zero fractional ideal $I \subseteq K$ as input (represented by a \mathbb{Z} -basis), id-HSVP $_{\mu}$ problem asks to find $x \in I \setminus \{0\}$ satisfying

$$\|\sigma(x)\| \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}}.$$

What comes to average-case problems, both the SIS and LWE problems have natural “ring-based” analogues where \mathbb{Z} is replaced with the ring of integers of a number field [12, 13]. Both can be reduced from problems over ideal lattices.

In the following, we once again assume that the parameters are functions of λ .

Definition 2.78 (Ring-SIS). Let \mathcal{R} be a ring of integers and $m, q, \beta \in \mathbb{N}$ such that $q \geq 2$. Given random $\mathbf{a} \leftarrow_{\$} \mathcal{R}_q^m$ and some target $v \in \mathcal{R}_q$, the \mathcal{R} -SIS $_{m,q,\beta,v}$ problem asks to find $\mathbf{u} \in \mathcal{R}^m$ satisfying

$$\mathbf{a}^T \mathbf{u} = v \pmod{q} \quad \text{and} \quad 0 < \|\sigma(\mathbf{u})\| \leq \beta.$$

Definition 2.79 (Ring-LWE, Search). Let \mathcal{R} be a ring of integers, $m, q \in \mathbb{N}$ such that $q \geq 2$ and χ be a distribution over \mathcal{R} . Let $\mathbf{a} \leftarrow_{\$} \mathcal{R}_q^m$, $\mathbf{e}^T \leftarrow_{\$} \chi^m$ and $s \in \mathcal{R}_q$. Furthermore, define

$$\mathbf{b}^T = s\mathbf{a}^T + \mathbf{e}^T \pmod{q}.$$

Given (\mathbf{a}, \mathbf{b}) the \mathcal{R} -sLWE $_{m,q,\chi}$ problem asks to recover s .

Definition 2.80 (Ring-LWE, Decision). Let \mathcal{R} be a ring of integers and \mathbf{b} be either generated as in Definition 2.79 or sampled uniformly in \mathcal{R}_q^m . Given \mathbf{b} , the \mathcal{R} -dLWE $_{m,q,\chi}$ problem asks to distinguish how it was generated.

2.4.9 NTRU

In addition to the problems mentioned above, there is one more standard problem over rings: NTRU; it was originally presented by Hoffstein et al. in [15]. Let us begin by defining an *NTRU instance*. Our definitions closely follow those presented in [18], although we slightly adapt them to better suit our purposes.

Definition 2.81 (NTRU instance). Let \mathcal{R} be a ring of integers, $q \in \mathbb{N}$ such that $q \geq 2$ and $\alpha \in \mathbb{N}$. Then, an (α, q) -NTRU instance is an element $h \in \mathcal{R}_q$ such that there exists $(f, g) \in \mathcal{R}^2 \setminus \{(0, 0)\}$ satisfying

$$gh + f = 0 \pmod{q} \quad (2.2)$$

and $\|\sigma(f)\|, \|\sigma(g)\| \leq \alpha$.

For a distribution \mathcal{D} over NTRU instances to be useful in practice, we need to have an efficient sampling algorithm for \mathcal{D} . We call such algorithm an *NTRU setup*.

Definition 2.82 (NTRU setup). Use the same definitions as in Definition 2.81. Furthermore, let \mathcal{D} be a distribution over (α, q) -NTRU instances. A (\mathcal{D}, α, q) -NTRU setup is a PPT algorithm (with respect to $\log q$ and $\log \Delta_K$) that samples tuples $(h, f, g) \in \mathcal{R}_q \times \mathcal{R}^2$ such that

- the marginal distribution of h is \mathcal{D} ,
- $(f, g) \neq (0, 0)$ and $\|\sigma(f)\|, \|\sigma(g)\| \leq \alpha$, and
- $gh + f = 0 \pmod{q}$.

Like the LWE problems, NTRU comes in search and decision variants.

Definition 2.83 (NTRU, Search). Use the same definitions as in Definition 2.81. Furthermore, let \mathcal{D} be a distribution over (α, q) -NTRU instances and $\beta \in \mathbb{N}$ such that $\beta \geq \alpha$. Given $h \leftarrow \mathcal{D}$, the $\text{sNTRU}_{\mathcal{D}, \alpha, \beta, q}$ problem asks to recover $(f, g) \in \mathcal{R}^2 \setminus \{(0, 0)\}$ satisfying

$$gh + f = 0 \pmod{q} \quad \text{and} \quad \|\sigma(f)\|, \|\sigma(g)\| \leq \alpha.$$

Definition 2.84 (NTRU, Decision). Use the same definitions as in Definition 2.83. Furthermore let S be a subset of \mathcal{R}_q and let h be sampled either from \mathcal{D} or uniformly in S . Given h , the $\text{dNTRU}_{\mathcal{D}, \alpha, q, S}$ problem asks to distinguish how it was generated.

Remark 2.85. Instead of $gh + f = 0 \pmod{q}$, a more standard way of defining the NTRU problems is to require $h = f/g \pmod{q}$. Since the latter implies the former (up to different signs), we have a simple NTRU setup:

- Sample $f, g \in \mathcal{R}_q^\times$ such that $\|\sigma(f)\|, \|\sigma(g)\| \leq \alpha$.
- Define $h = -f/g \pmod{q}$.

Denote the resulting distribution of NTRU instances as \mathcal{D} . Assuming that $\text{dNTRU}_{\mathcal{D}, \alpha, q, \mathcal{R}_q^\times}$ is hard (and choosing α to be small), the above provides a method of sampling seemingly uniform elements in \mathcal{R}_q^\times that have the additional property of being a quotient of two short elements.

For a fixed h, q , the solutions to equation (2.2) lie in a certain rank-2 module.

Lemma 2.86 (Special case of Lemma 5.2). *Let $h \in \mathcal{R}$. Then, solutions to equation (2.2) lie in a free \mathcal{R} -module of rank 2, generated by the matrix*

$$\mathbf{B} = \begin{bmatrix} 1 & \\ -h & q \end{bmatrix} \in \mathcal{R}^{2 \times 2}.$$

The module of Lemma 2.86 is called an *NTRU module* (see [16]), and finding a solution to search-NTRU can equivalently be viewed as finding a short non-zero element in the module.

3 Connections between different vSIS variants

In this section, we will introduce the Vanishing-SIS problem and present a number of connections between different variants of the problem. This will be useful in building a better understanding of how the choice of parameters affects the hardness of the problem.

First, let us recall a formal definition of the vSIS problem from [21]. We introduce an additional parameter, η , that clarifies which norm is used for the ring elements. In practice, we take η to be either $\|\cdot\|$ or $\|\sigma(\cdot)\|$, i.e., the norm over the coefficient or the canonical embedding.

Definition 3.1 (Vanishing-SIS). Let $n, D, w, q, \beta \in \mathbb{N}$ and \mathcal{G} , a set of w -variate monomials of degree at most D , be functions of λ . Furthermore, let $\eta : \mathcal{R} \rightarrow \mathbb{R}$ be a norm for the elements of \mathcal{R} . The $\text{vSIS}_{\mathcal{R}, \mathcal{G}, n, q, \beta, \eta}$ problem is, given a set $V = \{\mathbf{v}_i\}_{i=1}^n$ of n uniformly random points in $(\mathcal{R}_q^\times)^w$, find a non-zero polynomial $p \in \mathcal{R}[X_1, \dots, X_w]$ with monomial support over \mathcal{G} such that

$$p(\mathbf{v}_i) = 0 \pmod{q} \quad \forall i \in [n]$$

and $\eta(p_j) \leq \beta$ for all coefficients p_j . The $\text{vSIS}_{\mathcal{R}, \mathcal{G}, n, q, \beta, \eta}$ assumption states that, for any PPT adversary \mathcal{A} , the probability of \mathcal{A} solving a uniformly random instance of $\text{vSIS}_{\mathcal{R}, \mathcal{G}, n, q, \beta}$ is negligible in λ .

Individual parameters are omitted when they are clear from the context. If \mathcal{G} is the set of all w -variate monomials of individual degree at most D , we denote the problem by $\text{vSIS}_{\mathcal{R}, D, w, n, q, \beta}$. To emphasize certain parameters, e.g. $n = n^*$ and $w = w^*$, we sometimes write $\text{vSIS}_{(n, w) = (n^*, w^*)}$. If η is omitted and is not clear from the context, the choice of η does not matter (as long as one is consistent with the choice).

As one can see, there are many possibilities when choosing the parameters of the vSIS problem. In the following sections, we propose reductions between some of the different variants; we are by no means claiming that our list is comprehensive. We leave it as an open problem to find further connections or improve the parameters of our reductions.

3.1 Different degree of the polynomial

Here, one direction is obvious: a lower-degree problem is always at least as hard as a higher-degree problem.

Claim 3.2. *Let $D, D' \in \mathbb{N}$ such that $D \geq D'$. Then,*

$$\text{vSIS}_D \leq \text{vSIS}_{D'}.$$

Proof. Let V be any vSIS_D instance. Give it to the $\text{vSIS}_{D'}$ oracle and receive a short vanishing polynomial p . Since $\deg(p) \leq D' \leq D$, p is a valid solution to the vSIS_D -instance. \square

Remark 3.3. Notice that in the above proof, if $D > D'$ the oracle output p actually yields several linearly independent solutions to vSIS_D . This is because we can “shift” the coefficients up or, more concretely, multiply the polynomial by sufficiently low-degree monomials in \mathcal{G} . To elaborate, all polynomials of form

$$X_1^{e_1} \cdots X_w^{e_w} p,$$

where $e_1 + \dots + e_w \leq D - D'$, are perfectly valid solutions. Corollary 2.9 says that there is a total of $\binom{w+D-D'}{w}$ such polynomials.

This is a trivial notion, but it will be useful for constructing vSIS trapdoors.

The other direction, i.e. reducing from a lower-degree problem to a higher-degree one, is not that trivial. To resolve this difficulty, we will also consider the problem of finding several, \mathcal{R} -linearly independent solutions with respect to a problem instance. Let us give a formal definition to simplify the upcoming notations.

Definition 3.4. Use the same definitions as in Definition 3.1; in addition, let $m \in \mathbb{N}$. We define the problem

$$\text{vSIS}_{\mathcal{R}, \mathcal{G}, n, q, \beta, \eta}^m$$

that asks, given \mathbf{v} , to find m \mathcal{R} -linearly independent solutions $p^{(1)}, \dots, p^{(m)}$, each of which must be a valid solution to $\text{vSIS}_{\mathcal{R}, \mathcal{G}, n, q, \beta, \eta}$.

Remark 3.5. Notice that the problem of Definition 3.4 essentially asks to find a short basis for a submodule of the kernel of the mapping $(p_0, \dots, p_D) \mapsto \sum_{i=0}^D p_i \mathbf{v}^i \bmod q$. Note that this problem may be considerably harder than the problem of finding only one solution. It can even be ill-posed: for a given \mathbf{v} there often exists only a limited number of short linearly independent solutions.

The reason for considering such a problem is that finding several linearly independent solutions to the higher degree problem can imply a solution to a lower degree problem. A drawback is that the solution norm grows exponentially in the difference of the degrees.

First, we examine the univariate case, $w = 1$. Let $D > 1$, \mathbf{v} be a vSIS instance, and $p^{(1)}, p^{(2)}$ be two degree- D linearly independent polynomials satisfying $p^{(1)}(\mathbf{v}) = p^{(2)}(\mathbf{v}) = 0 \bmod q$ and $\|\sigma(p^{(1)})\|, \|\sigma(p^{(2)})\| \leq \beta$. Using these, we can find a relatively short vanishing polynomial p of degree at most $D - 1$ by eliminating the leading coefficients: set

$$p = p_D^{(2)} p^{(1)} - p_D^{(1)} p^{(2)}.$$

It is easy to verify that $p(\mathbf{v}) = 0 \bmod q$, $\deg(p) \leq D - 1$ and $\|\sigma(p)\| \leq 2\beta^2$.

This elimination approach can be generalized to larger jumps between degrees: given $k + 1$ linearly independent solutions, we can eliminate the coefficients of k different monomials.

Claim 3.6. Let $D, k, \beta, \beta' \in \mathbb{N}$ such that $\beta \geq (k + 1)! (\beta')^{k+1}$. Then,

$$\text{vSIS}_{D, w=1, \beta, \|\sigma(\cdot)\|} \leq \text{vSIS}_{D+k, w=1, \beta', \|\sigma(\cdot)\|}^{k+1}$$

Proof. Suppose that PPT adversary \mathcal{A} can find $k + 1$ linearly independent solutions to a random $\text{vSIS}_{D+k,w=1,\beta'}$ instance with non-negligible probability. Then, we claim that the following algorithm solves $\text{vSIS}_{D,w=1,\beta}$ with non-negligible probability.

vSIS_{D+k,w=1,β'}-to-vSIS_{D,w=1,β}(v**)**

$(p^{(1)}, \dots, p^{(k+1)}) \leftarrow \mathcal{A}(\mathbf{v})$

$\mathbf{P} := \begin{bmatrix} p_{D+k}^{(1)} & \cdots & p_{D+k}^{(k+1)} \\ \vdots & \ddots & \vdots \\ p_{D+1}^{(1)} & \cdots & p_{D+1}^{(k+1)} \end{bmatrix} \in \mathcal{R}_q^{(k) \times (k+1)}$

for $i \in [k + 1]$ **do**

$x_i := (-1)^{i+1} M_i$

// here M_i denote $k \times k$ minors of \mathbf{P}

$p := [p^{(1)} \ \dots \ p^{(k+1)}] \mathbf{x}$

return p

We will show that if \mathcal{A} succeeds, then our reduction succeeds. Since we assumed that the success probability of \mathcal{A} is non-negligible, this implies that our reduction has a non-negligible success probability.

Thus, assume \mathcal{A} is successful. Since p is an \mathcal{R} -linear combination of $p^{(1)}, \dots, p^{(k+1)}$ and $p^{(i)}(\mathbf{v}) = 0 \pmod q \ \forall i \in [k + 1]$, p obviously satisfies $p(\mathbf{v}) \pmod q$. p is also non-zero since $p^{(i)}$ are linearly independent.

Since $p^{(i)}$ satisfy $\|\sigma(p^{(i)})\| \leq \beta'$ for all $i \in [k + 1]$, we have

$$\begin{aligned} \|\sigma(p)\| &= \max_{i \in [k+1]} \{\|\sigma(p_i)\|\} = \max_{i \in [k+1]} \left\{ \left\| \sigma \left(\sum_{j \in [k+1]} p_i^{(j)} x_j \right) \right\| \right\} \\ &\leq \max_{i \in [k+1]} \left\{ \sum_{j \in [k+1]} \|\sigma(p_i^{(j)} x_j)\| \right\} \leq \max_{i \in [k+1]} \left\{ \sum_{j \in [k+1]} \|\sigma(p_i^{(j)})\| k! (\beta')^k \right\} \\ &\leq (k + 1)! (\beta')^{k+1} \leq \beta. \end{aligned}$$

It remains to show that $\deg(p) \leq D$. Observe that for any $i \in \{D + 1, \dots, D + k\}$ we have

$$\left[p_i^{(1)} \ \dots \ p_i^{(k+1)} \right] \mathbf{x} = \sum_{j \in [k+1]} (-1)^{j+1} p_i^{(j)} M_j = \left| \frac{p_i^{(1)} \ \dots \ p_i^{(k+1)}}{\mathbf{P}} \right| = 0$$

where we used the Laplace expansion (Definition 2.2) for the second equality; the last equality is a consequence of the matrix having the same row twice. This implies $\mathbf{P}\mathbf{x} = \mathbf{0}$ like we wanted. \square

The above can be generalized for multivariate vSIS. The only difference is that, instead of having to eliminate the coefficients of k monomials, we need to eliminate the coefficients of $\binom{w+D+k}{D} - \binom{w+D}{D}$ monomials (as implied by Corollary 2.9).

Claim 3.7. Let $D, k, \beta, \beta' \in \mathbb{N}$. Denote $k' = \binom{w+D+k}{D} - \binom{w+D}{D}$ and let $\beta \geq (k' + 1)!(\beta')^{k'+1}$. Then,

$$\text{vSIS}_{D,\beta,\|\sigma(\cdot)\|} \leq \text{vSIS}_{D+k,\beta',\|\sigma(\cdot)\|}^{k'+1}.$$

As the proof is — apart from different parameters — essentially the same as in the proof of Claim 3.6, we will omit it.

3.2 Different number of variables

Although choosing the space of our vSIS problem to be univariate polynomials is conceptually simple and computationally efficient, for some applications we might want to consider multivariate polynomials instead. In the following, we will comment on the connections between different choices of w , the number of variables.

Reduction-wise, the situation here is extremely similar as with problems of different degrees. Adding more variables can be seen as increasing degrees of freedom, which only makes the problem easier; similarly to increasing the allowed degree D . Thus, we have a trivial reduction from any problem with more variables to a problem with fewer variables. In contrast, decreasing the number of variables generally makes the problem harder. As a result, if $w > w'$ we need several linearly independent vSIS_w solutions to find a solution for $\text{vSIS}_{w'}$.

These findings are formalized by the following claims. Since their proofs are almost identical to those presented in Section 3.1, they will be omitted.

Claim 3.8. Let $w, w' \in \mathbb{N}$ such that $w \geq w'$. Then,

$$\text{vSIS}_w \leq \text{vSIS}_{w'}.$$

Claim 3.9. Let $D, w, w', \beta, \beta' \in \mathbb{N}$ such that $w \leq w'$. Denote $k = \binom{w+D}{w} - \binom{w'+D}{w'}$ and let $\beta \geq (k + 1)!(\beta')^{k+1}$. Then,

$$\text{vSIS}_{D,w,\beta,\|\sigma(\cdot)\|} \leq \text{vSIS}_{D,w',\beta',\|\sigma(\cdot)\|}^{k+1}.$$

3.3 Different number of points

The number of points that the polynomial has to vanish at is another parameter that can be freely chosen, depending on the application. Once again, one direction of reductions is trivial here: if an adversary can solve vSIS for more points, it can also solve it for fewer points. In other words, adding more points can only make the problem harder.

Claim 3.10. Let $n, n' \in \mathbb{N}$ such that $n \leq n'$. Then

$$\text{vSIS}_n \leq \text{vSIS}_{n'}.$$

Proof. Given a vSIS_n instance V , the reduction samples $n' - n$ uniformly random points in $(\mathcal{R}_q^\times)^w$; let us denote that set by V' . Then, it calls $\text{vSIS}_{n'}$ oracle on input $V \cup V'$ and returns the oracle output. \square

The other direction, i.e. how to reduce from more points to fewer points, is not as clear cut. However, we can apply a “divide and conquer” approach here. Suppose that we have two polynomials, $p^{(1)}$ and $p^{(2)}$ such that $p^{(1)}(\mathbf{v}) = 0 \pmod q \forall \mathbf{v} \in V_1$ and $p^{(2)}(\mathbf{v}) = 0 \pmod q \forall \mathbf{v} \in V_2$; this implies

$$(p^{(1)}p^{(2)})(\mathbf{v}) = 0 \pmod q \forall \mathbf{v} \in V_1 \cup V_2.$$

Obviously, there is a trade-off: both the norm and the degree blow up in polynomial multiplication.

Claim 3.11. *Let $D, D', n, n', \beta, w \in \mathbb{N}$ such that $n \geq n'$ and denote $N = \lceil n/n' \rceil$. Assume $D \geq ND'$ and $\beta \geq v_{w,D,N}(\beta')^N$ where $v_{w,D,N}$ is as in Definition 2.10. Then,*

$$vSIS_{D,w,n,\beta,w,\|\sigma(\cdot)\|} \leq vSIS_{D',w,n',\beta',\|\sigma(\cdot)\|}.$$

Proof. Let \mathcal{A} be a PPT $vSIS_{D',w,n',\beta',\|\sigma(\cdot)\|}$ oracle with non-negligible success probability. Given a $vSIS_{D,w,n,\beta,w,\|\sigma(\cdot)\|}$ instance V , the reduction partitions it into N disjoint subsets V_i such that $|V_i| \leq n' \forall i \in [N]$. Then, for $i \in [N]$ it calls \mathcal{A} with input V_i ; denote the outputs $p^{(i)}$. Finally, it returns $p = p^{(1)} \dots p^{(N)}$.

With a non-negligible probability, \mathcal{A} succeeds; this means that we have $p(\mathbf{v}) = 0 \pmod q \forall \mathbf{v} \in V$ and $\deg(p) \leq \sum_{i \in [N]} \deg(p^{(i)}) \leq ND'$. Also, since the coefficients of p are sums of products of N elements and the maximal number of summands is described by $v_{w,D,N}$, we have $\|\sigma(p)\| \leq m_{w,D,N}(\beta')^N \leq \beta$. \square

3.4 Different moduli

Brakerski et al. showed in [42] that decision-LWE with modulus q can be reduced to decision-LWE with a smaller modulus. One might wonder if such modulus reduction technique exists for $vSIS$. Due to the limitations of the main result of Section 4, we are particularly interested in obtaining a reduction from $vSIS_{q^e}$ to $vSIS_{q^{e'}}$ for $e' < e$.

We can do this by exponentiating a solution. To demonstrate the basic idea, suppose that PPT adversary \mathcal{A} can solve $vSIS_{D=1,w=1,q,2\beta^2,\|\sigma(\cdot)\|}$. Then, given an instance \mathbf{v} of the problem $vSIS_{D=2,w=1,q^2,4\beta^4,\|\sigma(\cdot)\|}$, we can query \mathcal{A} for a degree-1 solution p . Now, for some $r \in \mathcal{R}$ we have $p(\mathbf{v}) = rq$ and squaring both sides yields that p^2 is a solution to the original problem. Therefore, we obtain

$$vSIS_{D=2,w=1,q^2,4\beta^4,\|\sigma(\cdot)\|} \leq vSIS_{D=1,w=1,q,2\beta^2,\|\sigma(\cdot)\|}.$$

Together with Claim 3.6, this implies

$$vSIS_{D=2,w=1,q^2,4\beta^4,\|\sigma(\cdot)\|} \leq vSIS_{D=2,w=1,q,\beta,\|\sigma(\cdot)\|}^2.$$

This approach can easily be generalized to handle a range of parameters. It should be noted that the parameters of the reduction are not very good; in addition to requiring several linearly independent solutions, the solution norm of the original problem grows super-exponentially with respect to the difference $e - e'$.

3.5 From worst-case vSIS to average-case vSIS

In the following we will show that, under certain parameters, the $\text{vSIS}_{n=1}$ problem is random self-reducible if we assume that decision-NTRU is hard. This reduction was sketched in [21] but here we state it formally.

To explain the high-level idea of the reduction, consider the univariate case: let v^* be any fixed $\text{vSIS}_{(n,w)=(1,1)}$ instance. We can rerandomize sample an NTRU instance $h = f/g \bmod q$ and let $v = v^*h$. Now, v is indistinguishable from uniform by the decision-NTRU assumption and hence we can query an average-case oracle for a short polynomial p vanishing at v .

Notice that by absorbing the powers of h in the coefficients, we get another polynomial p' that satisfies $p'(v^*) = p(v) = 0 \bmod q$, i.e. we get a polynomial vanishing at v^* just as we wanted. If we further multiply p' by $g^{\deg(p)}$ we can cancel out all the denominators. All of the coefficients of the resulting polynomial are products of short elements and therefore somewhat short themselves. This intuition is formalized in the following theorem.

Theorem 3.12. *Let $\alpha \in \mathbb{N}$ and χ be a distribution over \mathcal{R}_q^\times such that any element x in the support satisfies $\|\sigma(x)\| \leq \alpha$; also, denote the distribution $\{f/g \bmod q \mid f, g \leftarrow \chi\}$ by χ' . Moreover, let $w, \beta, \beta^* \in \mathbb{N}$ such that $\beta^* \geq \alpha^{Dw} \beta$. Then, assuming the hardness of $\text{dNTRU}_{\chi', \alpha, q, \mathcal{R}_q^\times}$, there exists a PPT reduction from worst-case $\text{vSIS}_{D, n=1, \beta^*, \|\sigma(\cdot)\|}$ to average-case $\text{vSIS}_{D, n=1, \beta, \|\sigma(\cdot)\|}$.*

Proof. Suppose \mathcal{A} solves $\text{vSIS}_{D, n=1, \beta, \|\sigma(\cdot)\|}$ with non-negligible probability. Then, we claim that the following algorithm solves any $\text{vSIS}_{D, n=1, \beta^*, \|\sigma(\cdot)\|}$ instance $\mathbf{v}^* \in (\mathcal{R}_q^\times)^w$ with non-negligible probability in probabilistic polynomial time.

WorstCase-to-AverageCase-vSIS $^{\mathcal{A}}(\mathbf{v}^*)$

for $i \in [w]$ **do**

$f_i, g_i \leftarrow \chi$

$h_i := f_i/g_i \bmod q$

$\mathbf{v} := (v_i^* h_i)_{i \in [w]} \bmod q$

$p \leftarrow \mathcal{A}(\mathbf{v})$

for $(i_1, \dots, i_w) \in \{0, \dots, D\}^w$ **do**

$p'_{(i_1, \dots, i_w)} := p_{(i_1, \dots, i_w)} \prod_{j \in [w]} h_j^{i_j} \bmod q$

// now $p'(\mathbf{v}^*) = p(\mathbf{v}) = 0 \bmod q$

$p^* := p' \prod_{i \in [w]} g_i^D$

return p^*

Firstly, observe that the algorithm makes is PPT and makes one call to average-case

oracle. Secondly, assuming p is a valid solution, we have

$$\begin{aligned} p^*(\mathbf{v}^*) &= \left(\sum_{(i_1, \dots, i_w) \in \{0, \dots, D\}^w} \left(p_{(i_1, \dots, i_w)} \prod_{j \in [w]} (v_j^* h_j)^{i_j} \right) \right) \prod_{i \in [w]} g_i^D \\ &= p(\mathbf{v}) \prod_{i \in [w]} g_i^D = 0 \pmod q \end{aligned}$$

because $p(\mathbf{v}) = 0 \pmod q$. Also, since multiplying with $\prod_{i \in [w]} g_i^D$ cancels out all of the denominators, we get

$$\|\sigma(p^*)\| \leq \|\sigma(p)\| \alpha^{Dw} = \beta^*.$$

Thus, p^* is a valid solution to the $\text{vSIS}_{D, n=1, \beta^*, \|\sigma(\cdot)\|}$ instance \mathbf{v}^* .

By the $\text{dNTRU}_{\chi', \alpha, q, \mathcal{R}_q^\times}$ assumption, \mathbf{v} is computationally indistinguishable from a random vSIS instance. Hence, we can assume that \mathcal{A} succeeds with a non-negligible probability, implying that the success probability of the reduction is also non-negligible. \square

We note that the parameters of the reduction are not great since the solution norm is exponential in both D and w . This means that the quality of the reduction rapidly decreases as these parameters increase.

However, the condition $n = 1$ does not pose a restriction. This is because adding more points only makes the problem harder, as discussed in Section 3.3.

4 A reduction from ideal-HSVP to vSIS

The goal of this section is to generalize the results of [18], where the authors reduce search-NTRU from ideal-HSVP. As a result, we obtain a reduction from a special distribution of ideal-HSVP to vSIS, under a specific parameter regime.

4.1 The reduction

Before going into the main theorem of the section, let us first state and prove a lemma that represents the core of the reduction.

Lemma 4.1 (Transforming ideal-HSVP instance to vSIS instance). *Let $q, D \in \mathbb{N}$ such that $q \geq 2$. Also, let $I \subseteq \mathcal{R}$ be a non-zero ideal of form $I = \langle z^D \rangle \cap \mathcal{R}$ where $z \in K$, and define $v = \lfloor q/z \rfloor \bmod q^D$. Then, for every such v we have*

- (i) *there exists a non-zero polynomial $p \in \mathcal{R}[X]$ such that $\deg(p) \leq D$, $p(v) = 0 \bmod q^D$ and*

$$\|\sigma(p)\| \leq d^{D+\frac{1}{2}} \cdot \Delta_K^{\frac{1}{2d}} \cdot \delta_K^D \cdot \mathcal{N}(I)^{\frac{1}{d}},$$

and

- (ii) *for any non-zero polynomial $p' \in \mathcal{R}[X]$ satisfying $\deg(p') \leq D$, $p'(v) = 0 \bmod q^D$ and*

$$\|\sigma(p')\|_\infty < \frac{q}{\Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}} \cdot \max \left\{ \|\sigma(z^{-1})\|_\infty^D, \left(\frac{\delta_K \cdot d}{2}\right)^D \right\} \cdot (2^{D+1} - 2)},$$

the leading coefficient of p' is in $I \setminus \{0\}$.

Proof. Towards (i), let α be the shortest non-zero element in J , measured in the infinity norm. Let

$$p(X) = \alpha \left(X + \left\{ \frac{q}{z} \right\} \right)^D.$$

We propose that p is a short degree- D polynomial in $\mathcal{R}[X]$ vanishing at v modulo q^D .

Observe that $\deg(p) = D$ by construction. To check the vanishing property, notice that $v = \frac{q}{z} - \left\{ \frac{q}{z} \right\}$, and thus

$$p(v) = \alpha \left(\frac{q}{z} \right)^D = 0 \bmod q^D$$

where the second equality follows from our assumptions: since $\alpha \in \langle z^D \rangle$ there exist $r \in \mathcal{R}$ such that $\alpha = rz^D$ and thus $\alpha/z^D \in \mathcal{R}$.

Next, show that the coefficients of p are in \mathcal{R} . Observe that they are given by the binomial expansion

$$p_i = \alpha \binom{D}{i} \left\{ \frac{q}{z} \right\}^{D-i}, \quad i \in \{0, \dots, D\}.$$

We have

$$\alpha \left\{ \frac{q}{z} \right\}^{D-i} = \alpha \left(\frac{q}{z} - v \right)^{D-i} = \sum_{j=0}^{D-i} \alpha \binom{D-i}{j} \left(\frac{q}{z} \right)^j (-v)^{D-i-j}.$$

Since $\alpha \in \langle z^D \rangle$, all of the summands are in \mathcal{R} and hence $\alpha \{q/z\}^{D-i} \in \mathcal{R}$ for all $i \in \{0, \dots, D\}$. This implies $p \in \mathcal{R}[X]$.

Lastly, let us compute an upper bound on the norm of the solution. Minkowski's bound implies $\|\sigma(\alpha)\|_\infty \leq \Delta_K^{1/(2d)} \cdot \mathcal{N}(I)^{1/d}$ and by equivalence of norms, $\|\sigma(\alpha)\| \leq \sqrt{d} \cdot \Delta_K^{1/(2d)} \cdot \mathcal{N}(I)^{1/d}$. Also, by Proposition 2.75 we have $\|\sigma(\{q/z\})\|_\infty \leq \delta_K \cdot d/2$. Thus,

$$\begin{aligned} \|\sigma(p)\| &\leq \|\sigma(\alpha)\| \cdot \max_{i \in \{0, \dots, D\}} \left\{ \binom{D}{i} \right\} \cdot \left\| \sigma \left(\left\{ \frac{q}{z} \right\} \right) \right\|_\infty^D \\ &\leq \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}} \cdot \max_{i \in \{0, \dots, D\}} \left\{ \binom{D}{i} \right\} \cdot \left(\frac{\delta_K \cdot d}{2} \right)^D \\ &\leq d^{D+\frac{1}{2}} \cdot \Delta_K^{\frac{1}{2d}} \cdot \delta_K^D \cdot \mathcal{N}(I)^{\frac{1}{d}} \end{aligned}$$

where we used the bound $\max_{i \in \{0, \dots, D\}} \left\{ \binom{D}{i} \right\} \leq \sum_{i=0}^D \binom{D}{i}$, and by the binomial theorem

$$\sum_{i=0}^D \binom{D}{i} = (1+1)^D = 2^D. \quad (4.1)$$

Next, prove (ii). Without loss of generality we may assume that $\deg(p') = D$; if $\deg(p') = D^* < D$, simply consider the polynomial $X^{D-D^*} p'$ instead. We have

$$p'(v) = \sum_{i=0}^D p'_i \left(\frac{q}{z} - \left\{ \frac{q}{z} \right\} \right)^i = \sum_{i=0}^D p'_i \sum_{j=0}^i \binom{i}{j} \left(\frac{q}{z} \right)^j \left(- \left\{ \frac{q}{z} \right\} \right)^{i-j} = q^D r$$

for some $r \in \mathcal{R}$. Multiplying both sides by α/q^D and reordering yields

$$\frac{\alpha p'_D}{z^D} = \alpha r - \frac{\alpha}{q} \left(p'_D \sum_{j=0}^{D-1} \binom{D}{j} \frac{q^{j-D+1}}{z^j} \left(- \left\{ \frac{q}{z} \right\} \right)^{D-j} + \sum_{i=0}^{D-1} p'_i \sum_{j=0}^i \binom{i}{j} \frac{q^{j-D+1}}{z^j} \left(- \left\{ \frac{q}{z} \right\} \right)^{i-j} \right)$$

Denote the second term on the right-hand side as θ and define

$$M = \max \left\{ \left\| \sigma \left(z^{-1} \right) \right\|_\infty^D, \left\| \sigma \left(\left\{ \frac{q}{z} \right\} \right) \right\|_\infty^D \right\}.$$

Then, observe that

$$\begin{aligned} \|\sigma(\theta)\|_\infty &\leq \frac{\|\sigma(\alpha)\|_\infty}{q} \cdot \|\sigma(p')\|_\infty \cdot M \cdot \left(\sum_{j=0}^{D-1} \binom{D}{j} + \sum_{i=0}^{D-1} \sum_{j=0}^i \binom{i}{j} \right) \\ &= \frac{\Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}}}{q} \cdot \|\sigma(p')\|_\infty \cdot \max \left\{ \left\| \sigma \left(z^{-1} \right) \right\|_\infty^D, \left(\frac{\delta_K \cdot d}{2} \right)^D \right\} \cdot (2^{D+1} - 2) \end{aligned}$$

where we used Minkowski's bound on $\|\sigma(\alpha)\|_\infty$, as well as (4.1) and properties of geometric sums to get

$$\sum_{j=0}^{D-1} \binom{D}{j} + \sum_{i=0}^{D-1} \sum_{j=0}^i \binom{i}{j} = 2^D - 1 + \sum_{i=0}^{D-1} 2^i = 2^D - 1 + \frac{1-2^D}{1-2} = 2^{D+1} - 2.$$

Therefore, by the assumption on $\|\sigma(p')\|_\infty$, we have that $\|\sigma(\theta)\|_\infty < 1$. Because $\alpha \in \langle z^D \rangle$, $\alpha p'_D / z^D \in \mathcal{R}$. We also have $\alpha r \in \mathcal{R}$, and \mathcal{R} being an additive group hence implies $\theta \in \mathcal{R}$. Since the infinity norm of any non-zero element in \mathcal{R} is greater than or equal to 1, we conclude that $\theta = 0$.

As a result, $\alpha p'_D / z^D = \alpha r$. Dividing both sides by α we get that $\frac{p'_D}{z^D} = r \in \mathcal{R}$ and therefore $p'_D \in \langle z^D \rangle$, concluding the proof. \square

Remark 4.2. One could obtain a tighter bound for $\|\sigma(p)\|$ by using a less naive upper bound for the binomial coefficients. Writing

$$\max_{i \in \{0, \dots, D\}} \left\{ \binom{D}{i} \right\} = \binom{D}{\lfloor D/2 \rfloor} = \frac{D!}{\lfloor D/2 \rfloor! \lceil D/2 \rceil!}$$

and using the inequalities

$$\sqrt{2\pi D} \left(\frac{D}{e} \right)^D e^{\frac{1}{12D+1}} < D! < \sqrt{2\pi D} \left(\frac{D}{e} \right)^D e^{\frac{1}{12D}}$$

appears to yield a good bound. However, the resulting expression is much more complicated than the naive bound and using it does not seem to yield significantly stronger results. Thus, we choose to avoid it in this thesis.

Now we are ready to prove the main theorem.

Theorem 4.3. *Let $D, N_0, N_1, q, \beta \in \mathbb{N}$ such that $N_0 \leq N_1$,*

$$\beta \geq d^{D+\frac{1}{2}} \cdot \Delta_K^{\frac{1}{2d}} \cdot \delta_K^D \cdot N_1^{\frac{1}{d}}$$

and

$$q > \beta \cdot \Delta_K^{\frac{1}{2d}} \cdot N_1^{\frac{1}{d}} \cdot \max \left\{ \left\| \sigma(z^{-1}) \right\|_\infty^D, \left(\frac{\delta_K \cdot d}{2} \right)^D \right\} \cdot (2^{D^*+1} - 2).$$

Also, define

$$\mu = \frac{\beta}{\sqrt{d} \cdot \Delta_K^{1/(2d)} \cdot N_0^{1/d}}.$$

There is a PPT (with respect to $\text{size}(z)$, $\log q$ and D) reduction from worst-case id-HSVP_μ to worst-case $\text{vSIS}_{D,q^D,\beta}$ for ideals $I \subseteq \mathcal{R}$ that

- satisfy $\mathcal{N}(I) \in [N_0, N_1]$ and
- are of form $I = \langle z^D \rangle \cap \mathcal{R}$ where $z \in K$.

Moreover, let $\mathcal{D}^{\text{id-HSVP}}$ be a distribution over ideals satisfying the above conditions. Then, there exists a distribution $\mathcal{D}^{\text{vSIS}}$ over $\text{vSIS}_{D,q^D,\beta}$ instances and a PPT (w.r.t. $\text{size}(z)$, $\log q$ and D) reduction from average-case id-HSVP_μ (for ideals sampled from $\mathcal{D}^{\text{id-HSVP}}$) to average-case $\text{vSIS}_{D,q^D,\beta}$ (for instances sampled from $\mathcal{D}^{\text{vSIS}}$ over $\text{vSIS}_{D,q^D,\beta}$).

Proof. Let \mathcal{A} be a PPT worst-case $\text{vSIS}_{D,q^D,\beta}$ oracle. Define the reduction $\text{id-HSVP-to-vSIS}_{D,q^D,\beta}^{\mathcal{A}}$ that takes as input an ideal $I = \langle z^D \rangle \cap \mathcal{R}$ satisfying $\mathcal{N}(I) \in [N_0, N_1]$.

$\text{id-HSVP-to-vSIS}_{D,q^D,\beta}^{\mathcal{A}}(I)$

$v := \left\lfloor \frac{q}{z} \right\rfloor \bmod q^D$

$p \leftarrow \mathcal{A}(v)$

Let p^* be the leading coefficient of the polynomial p

return p^*

Due to the lower bound on β , v is a valid $\text{vSIS}_{D,q^D,\beta}$ instance by the first claim of Lemma 4.1. Therefore p is a non-zero polynomial in $\mathcal{R}[X]$ satisfying $\deg(p) \leq D$, $p(v) = 0 \bmod q^D$ and $\|\sigma(p)\| \leq \beta$. Since $\|\sigma(p)\|_\infty \leq \|\sigma(p)\|$ and thanks to q being bounded from below, the second claim of Lemma 4.1 implies that $p^* \in I \setminus \{0\}$. Observe that

$$\|\sigma(p^*)\| \leq \beta \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot N_0^{\frac{1}{d}} \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}}$$

and hence p^* is a solution to the id-HSVP_μ instance I .

To conclude the proof of the first part, it remains to bound the running time of the reduction. Notice that we query the oracle once and the rest of the operations consist of division, rounding, taking residue and finding the leading coefficient. All of these can be done in time $\text{poly}(\text{size}(z), \log q, D)$.

To prove the second part of the theorem, consider the same reduction as before but let \mathcal{A} now be a PPT average-case oracle with non-negligible success probability. By a similar argument that was used when proving the first part, the reduction does not decrease the success probability and the running time is still polynomial. \square

The next corollary further emphasises the approximation factor of the reduction and follows from combining the definition of μ and the lower bound on β in the previous theorem.

Corollary 4.4. *Use the same notations as in Theorem 4.3. Also, suppose that there exists a set of ideals of \mathcal{R} satisfying the conditions of the theorem and denote that set by S .*

There exists a reduction from worst-case (respectively, average-case) id-HSVP_μ for ideals in S to worst-case (resp. average-case) $\text{vSIS}_{D,q^D,\beta}$, with

$$\mu = O\left(d^D \cdot \delta_K^D \cdot \left(\frac{N_1}{N_0}\right)^{\frac{1}{d}}\right).$$

Assuming that $(N_1/N_0)^{1/d} = \text{poly}(d)$, the approximation factor is polynomial in d (for a constant D).

Remark 4.5. In the previous corollary, assuming $(N_1/N_0)^{1/d}$ to be polynomial in d is not restrictive by itself. Note that we can choose a lower bound N_0 such that there exists sufficiently many ideals $I \subseteq \mathcal{R}$ with $\mathcal{N}(I) \leq N_0$. Then, we claim that if we set $N_1 = 2^d N_0$ (such that $(N_1/N_0)^{1/d} = 2$), there are at least as many ideals with norm in $[N_0, N_1]$. This is because we can scale ideals with small norm up using the idea discussed in Section 4.1 of [18].

In more detail, for any non-zero integral ideal I such that $\mathcal{N}(I) \leq N_0$ we define

$$I' = \left\lfloor 2 \left(\frac{N_0}{\mathcal{N}(I)} \right)^{\frac{1}{d}} \right\rfloor \cdot I$$

I' is obviously integral; moreover, we claim that its norm is in $[N_0, 2^d N_0]$. Firstly, observe that because $N_0/\mathcal{N}(I) \geq 1$, $(N_0/\mathcal{N}(I))^{1/d} \geq 1$ and hence $\left\lfloor 2(N_0/\mathcal{N}(I))^{1/d} \right\rfloor \geq (N_0/\mathcal{N}(I))^{1/d}$. Thus

$$\mathcal{N}(I') = \left\lfloor 2 \left(\frac{N_0}{\mathcal{N}(I)} \right)^{\frac{1}{d}} \right\rfloor^d \cdot \mathcal{N}(I) \geq N_0.$$

Secondly,

$$\mathcal{N}(I') = \left\lfloor 2 \left(\frac{N_0}{\mathcal{N}(I)} \right)^{\frac{1}{d}} \right\rfloor^d \cdot \mathcal{N}(I) \leq \left(2 \left(\frac{N_0}{\mathcal{N}(I)} \right)^{\frac{1}{d}} \right)^d \cdot \mathcal{N}(I) \leq 2^d N_0,$$

proving our claim.

However, when combined with the other two conditions for the ideals, the situation is not as clear; we will discuss this further in the next section.

4.2 On the restrictions

Let us then study the restrictions of our reduction more closely and discuss why they do not pose a problem in the ideal-HSVP to search-NTRU reduction of [18]. First, we consider ideal-HSVP only for integral ideals. However, this is not a true restriction since we can efficiently compute an element $k \in K$ such that $kI \subseteq \mathcal{R}$, i.e. any non-integral ideal can be scaled up so that it becomes integral (a simple approach is to take the least common multiple of the denominators of the \mathbb{Z} -basis coefficients of I). If we define $I' = kI$ and have $x' \in I'$, it is easy to verify that $x'/k \in I$. Also, if x' satisfies $\|\sigma(x')\| \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{1/(2d)} \cdot \mathcal{N}(I')^{(1/d)}$, then

$$\left\| \sigma \left(\frac{x'}{k} \right) \right\| \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \frac{\mathcal{N}(I')^{\frac{1}{d}}}{k} = \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}}.$$

In other words, the solutions for ideal-HSVP are invariant under scaling and, as a corollary, ideal-HSVP for all fractional ideals is equivalent to ideal-HSVP for integral ideals only.

Second, we have an actual restriction: we require the ideal I to be of form $\langle z^D \rangle \cap \mathcal{R}$ for some $z \in K$. This restriction seems to be an inherent side-product our techniques since in the proof we make use the property that for any element x in such ideal, x/z^D is in \mathcal{R} . However, if $D > 1$, not all ideals — even integral ones — can be written in such a way. For example, consider $K = \mathbb{Q}$, $I = \langle 2 \rangle$ and $D = 2$; there do exist exactly two real possibilities for z , $\pm\sqrt{2}$, but neither are in \mathbb{Q} . While it is true that such z always exists within the field of algebraic numbers \mathbb{A} (since it is the algebraic closure of K), this is not helpful in practice since the degree $[\mathbb{A} : \mathbb{Q}]$ is not finite.

If $D = 1$, there is no restriction as for any $I \subseteq \mathcal{R}$ one can efficiently compute z such that $I = \langle z \rangle \cap \mathcal{R}$, as demonstrated by Lemma 4.2 of [18]. Moreover, z can be chosen such that

$$\|\sigma(z^{-1})\| \leq 2^{2(d+1)} d^2 \Delta_K^{2/d} \delta_K^4 \mathcal{N}(I)^3,$$

as stated in Lemma A.1. The bound is interesting since $\|\sigma(z^{-1})\|_\infty$ affects the lower bound of q in our reduction. Unfortunately, the same approach does not seem to apply when $D > 1$, and hence we do not know if $\|\sigma(z^{-1})\|$ can be given a similar upper bound in that case.

Third, the norm of the ideal I is assumed to be bounded both from above and from below. In [18], the authors handle the lower bound by scaling the ideal up; as discussed earlier, the solutions to ideal-HSVP are invariant under scaling. This does not work in our context since for $k \in K$,

$$k \cdot (\langle z^D \rangle \cap \mathcal{R}) = \langle k \cdot z^D \rangle \cap \langle k \rangle$$

which means that we lose the desired form when scaling.

For the upper bound on the norm, one could consider using the ideal-HSVP self-reduction presented in [43] and subsequently adapted in [18]. In more detail, there is a distribution a reduction from worst-case ideal-HSVP (for all fractional ideals) to average-case ideal-HSVP for ideals sampled from $\mathcal{D}_N^{\text{id-HSVP}}$ where $\mathcal{D}_N^{\text{id-HSVP}}$ is a distribution over non-zero integral ideals of norm at most N . The reduction runs in polynomial time if $N \geq (2^d \cdot 6d^{1.5} \log(d) \Delta_K^{1/(2d)} \cdot \delta_K)^d$. However, it is not clear how many ideals lie in the intersection of the support of $\mathcal{D}_N^{\text{id-HSVP}}$ and the set of ideals of form $\langle z^D \rangle \cap \mathcal{R}$.

Finally, let us discuss an approach that does not work but is enlightening nevertheless. Lemma 2.64 implies that $\langle z^D \rangle \cap \mathcal{R} \subseteq \langle z \rangle \cap \mathcal{R}$, and thus it is tempting to exploit this fact using the following idea. For an ideal sampled from $\mathcal{D}_N^{\text{id-HSVP}}$, scale it up if its norm is below $N/2^d$. Then, compute z such that $I = \langle z \rangle \cap \mathcal{R}$ (and $\|\sigma(z^{-1})\|$ is somewhat small). Then, define $I' = \langle z \rangle \cap \mathcal{R} \subseteq I$ and input that to our reduction. By Lemma 2.65, $I^D \subseteq I'$ and therefore $\mathcal{N}(I') \leq \mathcal{N}(I^D) = \mathcal{N}(I)^D$. Notice that the inequality is in fact an equality for any principal integral ideal, and hence the bound is tight.

Now, suppose our reduction returns $p^* \in I' \setminus \{0\}$ such that

$$\|\sigma(p^*)\| \leq \mu' \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I')^{\frac{1}{d}}.$$

By the previous inequality of the norms of the ideals, this implies

$$\|\sigma(p^*)\| \leq \mu \cdot \sqrt{d} \cdot \Delta_K^{\frac{1}{2d}} \cdot \mathcal{N}(I)^{\frac{1}{d}}$$

for $\mu = \mu' \cdot \mathcal{N}(I)^{\frac{D-1}{d}}$. Since we needed $N = \Omega(2^{d^2})$ for the ideal-HSVP self-reduction to run in polynomial time, we have $\mu = \Omega(2^d)$ even if μ' is polynomial in d . This is no better than what can be achieved using polynomial-time block Korkine-Zolotarev (BKZ) lattice reduction [44]. If we do as [18] and consider also more expensive reductions (instead of purely polynomial-time ones), we can get a smaller bound for N . Unfortunately, we still do not get a meaningful reduction: it takes exponential time (w.r.t. d) to achieve a polynomial-size approximation factor, similarly to what happens with BKZ.

Thus, we get

$$\mathbf{B}^{-1}\mathbf{T} = \begin{bmatrix} g & & & & & a_0 \\ & vg-f & & & & va_0+a_1 \\ & & \ddots & & & \vdots \\ & & & v^{D-2}(vg-f) & \cdots & vg-f \\ & & & & & g \\ v^{D-1}(vg-f)/q & \cdots & \cdots & v(vg-f)/q & (vg-f)/q & \left(\sum_{i=0}^{D-1} v^{D-1-i} a_i \right) / q \\ & & & & & \left(\sum_{i=0}^D v^{D-i} a_i \right) / q \end{bmatrix}$$

The elements of the first d rows are trivially in \mathcal{R} so let us focus on the last row. For the first D elements of that row, notice that $v = f/g$ implies $vg - f = 0 \pmod q$ and therefore $(vg - f)/q \in \mathcal{R}$. For the last element, by our assumptions $\sum_{i=0}^D f^{D-i} g^i a_i = 0 \pmod q$. Multiplying both sides by g^{-D} implies $\sum_{i=0}^D v^{D-i} a_i = 0 \pmod q$, thus the last element is also in \mathcal{R} and we have proven (i).

Towards (ii), notice that the determinant of \mathbf{A} is easy to compute using the Laplace expansion along the last column. This gives

$$\det(\mathbf{B}) = \sum_{i=0}^D f^{D-i} g^i a_i = q.$$

Also, since \mathbf{B}^{-1} is a lower triangular matrix, its determinant is simply the product of the diagonal elements (by Laplace expansion), i.e. $\det(\mathbf{B}^{-1}) = 1/q$. Now, using the multiplicativity of the determinant we get

$$\det(\mathbf{B}^{-1}\mathbf{T}) = \det(\mathbf{B}^{-1}) \det(\mathbf{T}) = 1$$

which concludes the proof. \square

5.1.3 Solving for the last column

To construct trapdoors we need to be able to solve for the last column. Concretely, given f, g , we need to find $a_0, \dots, a_D \in \mathcal{R}$ satisfying (5.1). For the special case $D = 1$ (i.e., NTRU), there exist at least three different classes of algorithms: the classic resultant-based algorithm [16], a Hermite normal form based algorithm [45], and the field norm based algorithms of [46]. Since the last approach offers the best asymptotic performance, we will generalize that to the vSIS context. In [46], it is presented in two variants, recursive and iterative. For simplicity, we will only define the recursive variant, but we note that the iterative approach could also be generalized with relatively little effort.

The high-level idea of the algorithm is to utilize a “project-then-lift” paradigm. Since it can be hard to directly find solutions over \mathcal{R} , we map the equation down to a smaller subring, e.g. \mathbb{Z} and solve the equation there. That solution is then used to deduce a solution for the original problem.

To be more concrete, to map (5.1) onto \mathbb{Z} , we take the field norm of the terms $f^D, f^{D-1}g, \dots, fg^{D-1}, g^D$. This yields a linear Diophantine equation,

$$\sum_{i=0}^D \mathcal{N}(f^{D-i} g^i) a'_i = q,$$

where we look for suitable coefficients $a'_i \in \mathbb{Z}$. If $\gcd(\mathcal{N}(f^D), \mathcal{N}(f^{D-1}g), \dots, \mathcal{N}(g^D))$ divides q , we can find a solution (a'_1, \dots, a'_D) using the extended Euclidean algorithm (see Claim 2.18). Now, define $f^\times = \prod_{\sigma_i \in \text{Gal}(K/\mathbb{Q}) \setminus \{\text{id}\}} \sigma_i(f)$ (where id denotes the identity automorphism) and g^\times in a similar way. Moreover, for $i \in \{0, \dots, D\}$ define

$$a_i = (f^\times)^{D-i} (g^\times)^i a'_i.$$

Since $\mathcal{N}(f) = f f^\times$ and $\mathcal{N}(g) = g g^\times$, we can verify that (a_0, \dots, a_D) is indeed a solution to (5.1).

However, this is a naive approach and has little benefit over using the resultant mapping for the projection. The real benefit of using the field norm is being able to exploit the tower of fields structure of power-of-2 cyclotomic fields. Recall from Lemma 2.48 that the field norm is transitive in towers; thus, we can perform the projecting and lifting via the intermediate fields. This is favorable in terms of both time and space complexity of the algorithm, as explained in [46].

In the following, we outline a recursive version of the algorithm that takes as input f, g in the ring of integers of K and the degree D . We assume that the cyclotomic field $K = \mathbb{Q}(\zeta_k)$ is implicitly known.

```

TowerSolverR( $f, g, D$ )
if  $f, g \in \mathbb{Z}$  then
   $G := \gcd(f, g)^D$ 
  if  $G \nmid q$  then
    abort
  Use extended Euclidean algorithm to compute  $(a_0, \dots, a_D) \in \mathbb{Z}^{D+1}$  s.t.
    
$$\sum_{i=0}^D f^{D-i} g^i a_i = G$$

  return  $(\frac{a_0 \cdot q}{G}, \dots, \frac{a_D \cdot q}{G})$ 
 $(a'_0, \dots, a'_D) \leftarrow \text{TowerSolverR}(\mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(f), \mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(g), D)$ 
 $f^\times := \prod_{\sigma_i \in \text{Gal}(\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})) \setminus \{\text{id}\}} \sigma_i(f)$ 
 $g^\times := \prod_{\sigma_i \in \text{Gal}(\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})) \setminus \{\text{id}\}} \sigma_i(g)$ 
for  $i \in \{0, \dots, D\}$  do
   $a_i := (f^\times)^{D-i} (g^\times)^i a'_i$ 
return  $(a_0, \dots, a_D)$ 

```

Proposition 5.4. *Let $f, g \in \mathcal{R}$ where \mathcal{R} is the ring of integers of a power-of-2 cyclotomic field $K = \mathbb{Q}(\zeta_k)$. If $\gcd(\mathcal{N}(f), \mathcal{N}(g))^D$ divides q , then $\text{TowerSolverR}(f, g, D)$ returns a solution to (5.1).*

Proof. For $k \in \{1, 2\}$, observe that if $\gcd(\mathcal{N}(f), \mathcal{N}(g))^D = \gcd(f, g)^D \mid q$, Lemma 2.19 and Claim 2.18 together imply that EEA successfully finds integers a_i such that $\sum_{i=0}^D f^{D-i} g^i a_i = G$. Moreover, since we require $G \mid q$, q/G must be integral and hence $a_i q/G$ are integers satisfying $\sum_{i=0}^D f^{D-i} g^i a_i q/G = q$.

Next, consider $k \geq 4$. By Lemma 2.48 we have $(\mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})} \circ \dots \circ \mathcal{N}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}})(f) = \mathcal{N}(f)$ and the same is true for g . Therefore, if $\gcd(\mathcal{N}(f), \mathcal{N}(g))^D \mid q$, EEA finds a solution at the deepest level of recursion.

Finally, verify that the lifting works correctly in the tower of fields for $k \geq 4$. Towards this, if a'_i are ring elements satisfying

$$\sum_{i=0}^D \mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(f)^{D-i} \mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(g)^i a'_i = q,$$

then substituting $\mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(f) = f f^\times$, $\mathcal{N}_{\mathbb{Q}(\zeta_k)/\mathbb{Q}(\zeta_{k/2})}(g) = g g^\times$ implies

$$\sum_{i=0}^D (f f^\times)^{D-i} (g g^\times)^i a'_i = q,$$

and using the definition of a_i completes the statement. A similar argument shows that exactly the same happens at the deeper levels of recursion. This completes the proof. \square

5.2 Preimage sampling

Next, let us see how to perform preimage sampling using vSIS trapdoors. Recall from Section 2.4.7 how, for $a \in K$ (where $[K : \mathbb{Q}] = d$), we defined $M(a) \in \mathbb{R}^{d \times d}$ as the matrix that represents multiplication by a under the coefficient embedding τ . Let \mathbf{T} be as in Theorem 5.3. For every $\mathbf{x} = (x_1, \dots, x_{D+1}) \in \mathcal{R}^{D+1}$, the multiplication $\mathbf{T}\mathbf{x}$ can be written under the coefficient embedding as

$$M(\mathbf{T})\tau(\mathbf{x}) = \begin{bmatrix} M(g) & & & & M(a_0) \\ M(-f) & \ddots & & & M(a_1) \\ & \ddots & \ddots & & \vdots \\ & & \ddots & M(g) & M(a_{D-1}) \\ & & & M(-f) & M(a_D) \end{bmatrix} \begin{bmatrix} \tau(x_1) \\ \tau(x_2) \\ \vdots \\ \tau(x_D) \\ \tau(x_{D+1}) \end{bmatrix}$$

Thus, if we can show that the GS-norm of $M(\mathbf{T})$ is reasonably short, we can apply the GPV sampling technique from Section 2.3.4 to sample vSIS preimages; we dedicate Section 5.3 to that.

For certain choices of K (in particular, power-of-2 cyclotomic fields), it is possible to take a more sophisticated approach and leverage the fast Fourier sampling algorithm [47, 17]. It exploits the tower-of-fields structure to produce faster sampling, the downside being that one needs to do preprocessing in the trapdoor generation phase. Since a trapdoor is typically used repeatedly to sample several preimages, this is usually a beneficial tradeoff.

Note that the fast Fourier sampling algorithm is not different from GPV sampling in the sense that it requires the GS-norm of $M(\mathbf{T})$ to be short as well.

5.3 Bounding the Gram-Schmidt norm

In this section, we study the Gram-Schmidt norm of $M(\mathbf{T})$. Obtaining an upper bound seems infeasible, as such a result has not been established even in the less complicated case of NTRU trapdoors. This is why we can ultimately only provide heuristic results for the expected behavior.

5.3.1 Lower bound for the norm

From now on, we assume $K = \mathbb{Q}(\zeta)$ to be a power-of-2 cyclotomic field. The following lemma illustrates why this restriction is useful.

Lemma 5.5 (Adapted from Lemma 2.2 of [48]). *Let $K = \mathbb{Q}(\zeta)$ be a power-of-2 cyclotomic field of degree d and $n \in \mathbb{N}$. Furthermore, let $\mathbf{A} \in K^{n \times n}$ and denote $M(\mathbf{A}) = [\mathbf{m}_1 \ \dots \ \mathbf{m}_{nd}]$. Then,*

- for all $i \in [n]$, $\tau(\tilde{\mathbf{a}}_i) = \tilde{\mathbf{m}}_{(i-1)d+1}$, and
- $\|M(\mathbf{A})\|_{\text{GS}} = \max_{i \in [n]} \{\|\tilde{\mathbf{a}}_i\|\} = \|\mathbf{A}\|_{\text{GS}}$.

Remark 5.6. If we could generalize the above lemma to handle a larger range of number fields, it would allow generalizing most of our upcoming results as well. However, that would most likely require weakening the result (e.g., for the latter part, bounding $\|M(\mathbf{A})\|_{\text{GS}}$ from above using $\|\mathbf{A}\|_{\text{GS}}$ while allowing some slack) as it is easy to come up with counterexamples even for prime cyclotomic fields. We leave it as an open problem to consider such generalizations.

In light of Lemma 5.5, to estimate the GS-norm of $M(\mathbf{T})$ (with GSO performed over $\mathbb{R}^{(D+1)d}$), it suffices to consider the GS-norm of \mathbf{T} (with GSO performed over K). This makes the task easier. Also, thanks to the lemma, we can talk about the GS-norm of \mathbf{T} interchangeably with that of $M(\mathbf{T})$ without ambiguity.

With that out of the way, let us study $\|\mathbf{T}\|_{\text{GS}}$. During this process, we generalize a series of results from [22]. We begin by noticing that as long as f, g are short ring elements (as in NTRU), $\tilde{\mathbf{t}}_1, \dots, \tilde{\mathbf{t}}_D$ are indeed short. However, we also need to estimate the norm of $\tilde{\mathbf{t}}_{D+1}$. The following lemma formalizes this observation.

Lemma 5.7. *Use the same definitions as in Theorem 5.3; in addition, let K, d be as in Lemma 5.5. We have*

$$\|\mathbf{T}\|_{\text{GS}} = \max\{\|\mathbf{t}_1\|, \|\tilde{\mathbf{t}}_{D+1}\|\}.$$

Proof. Denote $M(\mathbf{T}) = [\mathbf{m}_1 \ \dots \ \mathbf{m}_{(D+1)d}]$; by the structure of \mathbf{T} , we have $\|\mathbf{m}_1\| = \dots = \|\mathbf{m}_{(D-1)d+1}\|$. By orthogonality, the vectors in the GSO cannot have norm greater than the corresponding original vectors. Therefore, we have $\|\tilde{\mathbf{m}}_i\| \leq \|\mathbf{m}_i\|$ for all $i \in [(D+1)d]$. Combining this with the previous fact and using $\tilde{\mathbf{m}}_1 = \mathbf{m}_1$ yields

$$\|\tilde{\mathbf{m}}_i\| \leq \|\mathbf{m}_1\| \ \forall i \in \{1, d+1, \dots, (D-1)d+1\}.$$

Using the first claim of Lemma 5.5, we conclude that

$$\|\tilde{\mathbf{t}}_i\| \leq \|\mathbf{t}_1\| \ \forall i \in [D].$$

The claim follows from the definition of GS-norm. □

Next, we derive an explicit expression for $\tilde{\mathbf{t}}_{D+1}$.

Lemma 5.8. *Use the same definitions as in Lemma 5.7. Furthermore, let $\tilde{\mathbf{T}} = [\tilde{\mathbf{t}}_1 \ \cdots \ \tilde{\mathbf{t}}_{D+1}]$ be the GSO of \mathbf{T} . We have*

$$\tilde{\mathbf{t}}_{D+1} = \frac{q}{\sum_{i=0}^D f^{D-i} g^i \overline{f^{D-i} g^i}} \left(\overline{f^D}, \overline{f^{D-1} g}, \dots, \overline{f g^{D-1}}, \dots, \overline{g^D} \right).$$

Proof. Denote $\mathbf{c} = \frac{q}{\sum_{i=0}^D f^{D-i} g^i \overline{f^{D-i} g^i}} \left(\overline{f^D}, \overline{f^{D-1} g}, \dots, \overline{f g^{D-1}}, \dots, \overline{g^D} \right)$. Recall that an orthogonal projection onto a subspace is always unique. Therefore, to prove that $\tilde{\mathbf{t}}_D = \mathbf{c}$ it suffices to show that \mathbf{c} is of form $\mathbf{t}_{D+1} - \mathbf{p}$ where \mathbf{p} is the orthogonal projection $\text{proj}_{\text{span}_K\{\mathbf{t}_1, \dots, \mathbf{t}_D\}}(\mathbf{t}_{D+1})$. This can be split into two separate statements:

- (i) $\mathbf{p} = \mathbf{t}_{D+1} - \mathbf{c} \in \text{span}_K\{\mathbf{t}_1, \dots, \mathbf{t}_D\}$, i.e. \mathbf{p} is a projection onto the correct subspace, and
- (ii) for all $j \in \{1, \dots, D\}$, $\langle \mathbf{c}, \tilde{\mathbf{t}}_j \rangle = 0$, i.e. the projection is orthogonal.

Since $\{\mathbf{t}_1, \dots, \mathbf{t}_D\}$ is a linearly independent set by construction, (i) is equivalent to showing that the determinant of $[\mathbf{t}_1 \ \cdots \ \mathbf{t}_D \ \mathbf{t}_{D+1} - \mathbf{c}]$ is zero. Applying the Laplace expansion yields

$$\begin{aligned} \det([\mathbf{t}_1 \ \cdots \ \mathbf{t}_D \ \mathbf{t}_{D+1} - \mathbf{c}]) &= \sum_{j=0}^D \left(a_j - \frac{q}{\sum_{i=0}^D f^{D-i} g^i \overline{f^{D-i} g^i}} \overline{f^{D-j} g^j} \right) f^{D-j} g^j \\ &= \sum_{j=0}^D a_j f^{D-j} g^j - \frac{q}{\sum_{i=0}^D f^{D-i} g^i \overline{f^{D-i} g^i}} \sum_{j=0}^D \overline{f^{D-j} g^j} f^{D-j} g^j = 0 \end{aligned}$$

where in the last equality we used (5.1).

Towards (ii), observe that each of $\mathbf{t}_1, \dots, \mathbf{t}_D$ can be written as a linear combination of $\{\tilde{\mathbf{t}}_1, \dots, \tilde{\mathbf{t}}_D\}$. By the construction of $\mathbf{t}_1, \dots, \mathbf{t}_D$ (see Theorem 5.3), we can verify that each of them is orthogonal to \mathbf{c} . Hence, $\langle \mathbf{c}, \tilde{\mathbf{t}}_j \rangle = 0$ for all $j \in \{1, \dots, D\}$; this concludes the proof. \square

The following result provides an easy way to calculate the GS-norm without actually computing the GSO. In practice, it provides a fast way to identify bad choices of f, g (that is, those that result in a large GS-norm) without going through the GSO process. The result is a direct corollary of lemmas 5.7 and 5.8.

Corollary 5.9. *Use the same definitions as in Lemma 5.7. We have*

$$\|\tilde{\mathbf{T}}\| = \max \left\{ \|\mathbf{t}_1\|, \left\| \frac{q}{\sum_{i=0}^D f^{D-i} g^i \overline{f^{D-i} g^i}} \left(\overline{f^D}, \overline{f^D g}, \dots, \overline{f g^D}, \dots, \overline{g^D} \right) \right\| \right\}.$$

We also obtain a nice lower bound for the norm of $\tilde{\mathbf{t}}_{D+1}$ that will aid in the analysis.

Lemma 5.10. *Use the same definitions as in Lemma 5.7. We have*

$$\|\tilde{\mathbf{t}}_{D+1}\| \geq \frac{q}{\|\mathbf{t}_1\|^D}.$$

Proof. By orthogonality of $\{\tilde{\mathbf{t}}_i\}_{i=1}^{D+1}$,

$$q = |\det(\mathbf{T})| = |\det(\tilde{\mathbf{T}})| = \prod_{i=1}^{D+1} \|\tilde{\mathbf{t}}_i\|.$$

By Lemma 5.7 we have $\prod_{i=1}^{D+1} \|\tilde{\mathbf{t}}_i\| \leq \|\mathbf{t}_1\|^D \|\tilde{\mathbf{t}}_{D+1}\|$ and the claim follows. \square

The following corollary combines lemmas 5.7 and 5.10.

Corollary 5.11. *Use the same definitions as in Lemma 5.7. We have*

$$\|\mathbf{T}\|_{\text{GS}} \geq \max\left\{\|\mathbf{t}_1\|, \frac{q}{\|\mathbf{t}_1\|^D}\right\}.$$

5.3.2 Numerical results

The results of the previous section provide a solid basis for understanding the GS norm of \mathbf{T} . In what follows, we attempt to complete the picture by providing results from numerical experiments. It will turn out that, with careful choice of parameters, we can sample the trapdoor from a distribution where $\|M(\mathbf{T})\|_{\text{GS}}$ is (heuristically) expected to be of order $q^{1/(D+1)}$.

Our experiments were motivated by the numerical results of [22] for NTRU trapdoors, as well as those of [49, 48] for module-NTRU trapdoors. In all cases, the norm of the last column of the trapdoor was found to be not much greater than the theoretical lower bound. We expected to observe a similar pattern for the vSIS trapdoors; thus, we anticipated that the norm of $\tilde{\mathbf{t}}_{D+1}$ would depend on $\|\mathbf{t}_1\|$ and be somewhat close to the lower bound given in Lemma 5.10.

We conducted our experiments using SageMath [50]; the source code and numerical results can be found at <https://github.com/kjyrkin/VanishingSIS>. In the experiments, we sampled several independent $f, g \in \mathcal{R}_q^\times$ by picking their coefficient embedding vectors from a discrete Gaussian distribution. The Gaussian parameter was varied to obtain a wide range of different values for $\|\mathbf{t}_1\|$. Then, for each pair f, g , we computed $\|\tilde{\mathbf{t}}_{D+1}\|$ using Lemma 5.8. Finally, we plotted $\|\tilde{\mathbf{t}}_{D+1}\|$ against $\|\mathbf{t}_1\|$ to analyze how the former depends on the latter.

This process was repeated for four different values of D , sampling 2000 independent pairs f, g for each of them. The field $K = \mathbb{Q}(\zeta_{64})$ was kept the same for each D , as well as the parameter $q = 1048897$. This specific choice of q was made to ensure that q is a prime satisfying $q = 1 \pmod{64}$, which implies that a random element of \mathcal{R}_q is a unit with a high probability (see Section 2.4.6).

The results of the experiments are shown in Figure 2. It is in line with our expectations, since most samples result in $\|\tilde{\mathbf{t}}_{D+1}\|$ close to the lower bound. Some

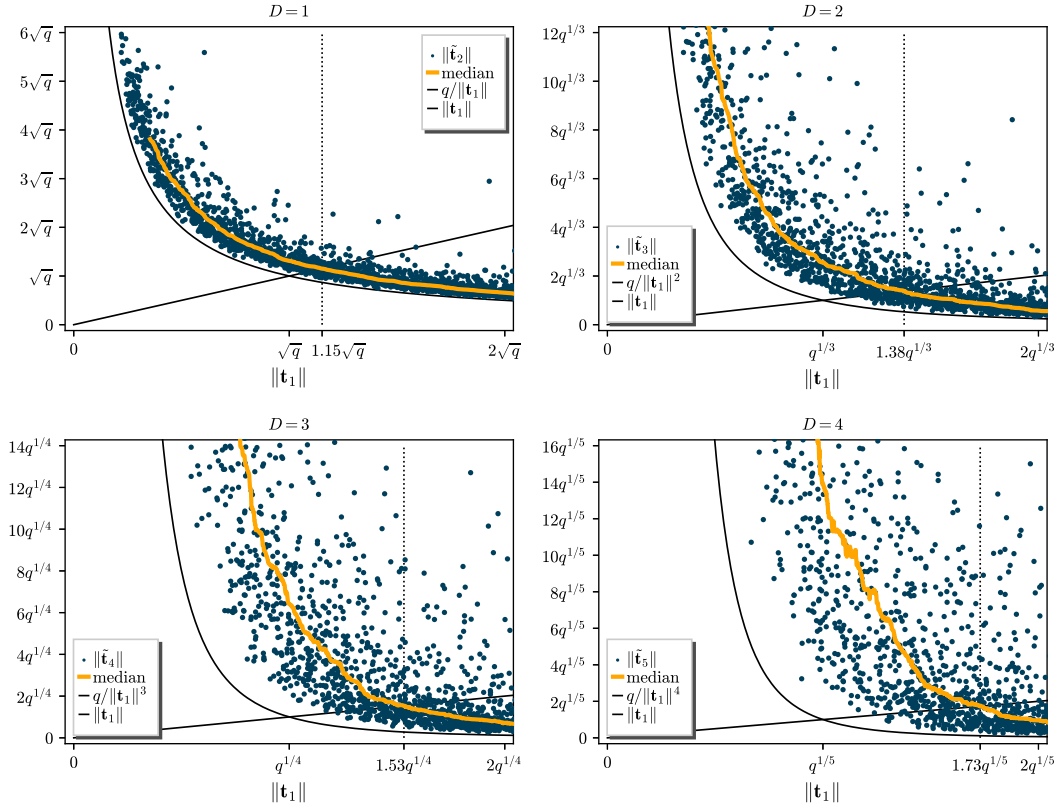


Figure 2: Norm of the last column of the Gram-Schmidt orthogonalization as a function of $\|\mathbf{t}_1\|$, for different values of D . The optimal choice of $\|\mathbf{t}_1\|$ (as given by our moving median -based heuristic) is indicated by the dotted vertical line.

$\|\tilde{\mathbf{t}}_{D+1}\|$ are still significantly larger (especially for larger D), which might seem alarming. However, this does not pose a problem if we can get evidence that the proportion of such outliers is sufficiently small. This is because we can simply reject such f, g during trapdoor sampling, without impairing the performance of the algorithm too much. This is similar to what is done in Algorithm 2 of [22].

To see if the cases resulting in a large $\|\tilde{\mathbf{t}}_{D+1}\|$ are really in the minority, we computed a heuristic which we call the moving median – specifically, of window size 250. This means that we first ordered the $(\|\mathbf{t}_1\|, \|\tilde{\mathbf{t}}_{D+1}\|)$ tuples with respect to their first coordinate to obtain a sequence $(\|\mathbf{t}_1\|, \|\tilde{\mathbf{t}}_{D+1}\|)_1, \dots, (\|\mathbf{t}_1\|, \|\tilde{\mathbf{t}}_{D+1}\|)_{2000}$. Then, for $s \in \{1, \dots, 1751\}$, we defined the consecutive subsequence

$$s_i = (\|\mathbf{t}_i\|, \|\tilde{\mathbf{t}}_{D+1}\|)_i, \dots, (\|\mathbf{t}_i\|, \|\tilde{\mathbf{t}}_{D+1}\|)_{i+249}.$$

Finally, for each i , we took coordinate-wise medians of s_i . The resulting line is drawn in yellow.

Regarding this heuristic and the lower bound of Corollary 5.11, the optimal $\|\mathbf{t}_1\|$ is where the yellow moving median line and the line $y = \|\mathbf{t}_1\|$ intersect. This is because for such $\|\mathbf{t}_1\|$, we expect around one half of the resulting $\|\tilde{\mathbf{t}}_{D+1}\|$ to be at most $\|\mathbf{t}_1\|$. Hence, approximately one half of resulting $\|\mathbf{T}\|_{\text{GS}}$ are expected to be equal to $\|\mathbf{t}_1\|$.

A dotted vertical line marks the location of the leftmost intersection point; we consider the corresponding value on the x -axis as an estimate of the optimal $\|\mathbf{T}\|_{\text{GS}}$. These values are illustrated in Figure 3 as a function of the parameter D . The figure demonstrates that $\|\mathbf{T}\|_{\text{GS}}$ decreases rapidly as D increases.

The (estimated) optimal $\|\mathbf{T}\|_{\text{GS}}$ seem to be slightly larger than the theoretical lower bound, $q^{1/(1+D)}$. Inspired by [48], we call the ratio of the optimal $\|\mathbf{T}\|_{\text{GS}}$ and $q^{1/(1+D)}$ as GS_SLACK since it describes the “slack” between the theoretical lower bound and what is practically achievable. These coefficients appear to grow roughly linearly with respect to D , as seen in Figure 4.

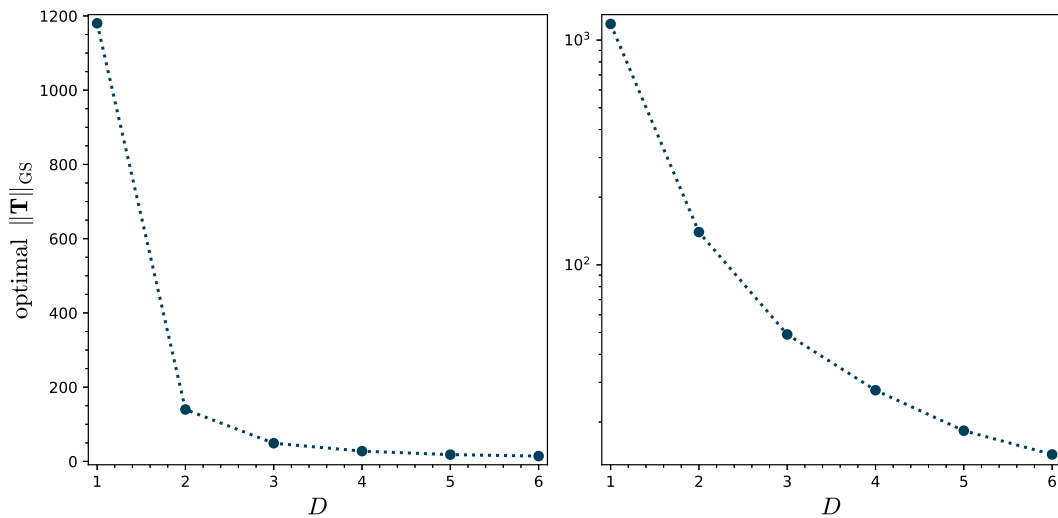


Figure 3: An estimate of the optimal $\|\mathbf{T}\|_{\text{GS}}$ for $D \in \{1, \dots, 6\}$. Both figures present the same data, but the right one uses logarithmic scale on the y-axis.

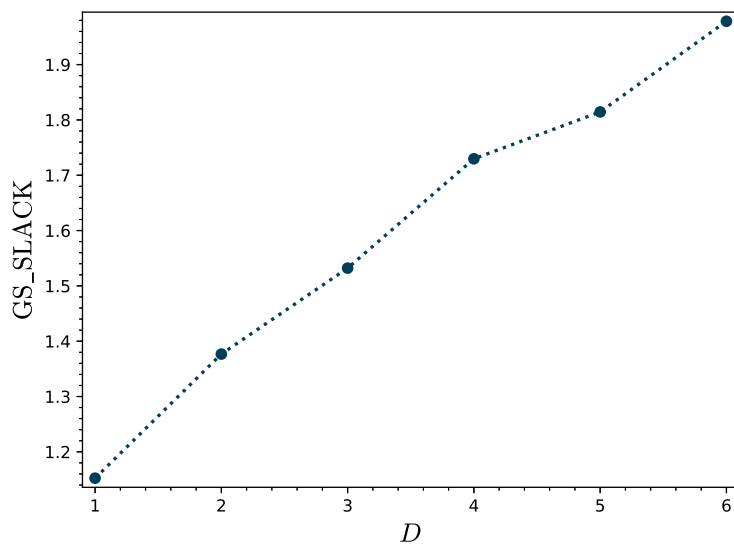


Figure 4: Growth of GS_SLACK with respect to the parameter D .

5.4 Nearest plane algorithm

Note that even when $\|\mathbf{T}\|_{\text{GS}}$ is small, the length of the last column, \mathbf{t}_{D+1} , could still be arbitrarily large. Geometrically, this happens when \mathbf{t}_{D+1} is close to parallel to some of the other columns. In this scenario, the size of the preimages remains unaffected. However, storing the trapdoor would require a significant amount of memory. Luckily, there exists a clever way to reduce its norm. We note that a similar approach is used with NTRU trapdoors in [16] and [46].

Consider the trapdoor basis $\mathbf{T} = [\mathbf{t}_1 \ \dots \ \mathbf{t}_{D+1}] \in \mathcal{R}^{(D+1) \times (D+1)}$, as defined in Theorem 5.3. Notice that we can add any \mathcal{R} -multiple of a column to another to obtain a different basis for the same module. Equivalently, we can replace any \mathbf{t}_i with $\mathbf{t}_i - \mathbf{c}$ where $\mathbf{c} \in \text{span}_{\mathcal{R}}\{\mathbf{t}_1, \dots, \mathbf{t}_{i-1}, \mathbf{t}_{i+1}, \dots, \mathbf{t}_{D+1}\}$. This motivates us to consider the problem of finding a \mathbf{c} such that $\|\mathbf{t}_{D+1} - \mathbf{c}\|$ is nearly minimal. Notice that this is, essentially, an instance of approximate CVP in the \mathcal{R} -module setting.

Babai's nearest plane algorithm [51] is a commonly used algorithm for solving approximate CVP. It takes as input a basis $\mathbf{A} \in \mathbb{R}^{n \times m}$ and a target vector $\mathbf{y} \in \mathbb{R}^n$. The algorithm first runs the Lenstra–Lenstra–Lovász (LLL) reduction [52] on \mathbf{A} and computes the GSO basis $\tilde{\mathbf{A}}$. Then it recursively breaks the problem down to finding a series of hyperplanes, decreasing in dimension, that are close to the target \mathbf{y} . For a more comprehensive explanation, we refer the reader to [51] as well as the lecture notes of Oded Regev [37].

In the following, we describe an algorithm that we will call Babai's reduction. It is essentially an \mathcal{R} -module analogue of the original algorithm. In our case, the basis does not need to be reduced since we expect the input to be $\mathbf{t}_1, \dots, \mathbf{t}_D$ which are short whenever f, g are short.

Definition 5.12 (Babai's reduction). Let $m, n \in \mathbb{Z}$ and $\mathbf{A} = [\mathbf{a}_1 \ \dots \ \mathbf{a}_m] \in \mathcal{R}^{n \times m}$ be a basis of some \mathcal{R} -module. We denote the GSO of \mathbf{A} as $[\tilde{\mathbf{a}}_1 \ \dots \ \tilde{\mathbf{a}}_m]$. For a target $\mathbf{y} \in \mathcal{R}^n$, we define the algorithm BabaiReduce as follows.

BabaiReduce(\mathbf{A}, \mathbf{y})
$\mathbf{b} := \mathbf{y}$ $i := n$ while $i \geq 1$ do $c_i := \left\lfloor \frac{\langle \mathbf{b}, \tilde{\mathbf{a}}_i \rangle}{\langle \tilde{\mathbf{a}}_i, \tilde{\mathbf{a}}_i \rangle} \right\rfloor$ $\mathbf{b} \leftarrow \mathbf{b} - c_i \mathbf{a}_i; \quad i \leftarrow i - 1$ return $\mathbf{y} - \mathbf{b}$

Now, given the trapdoor basis \mathbf{T} , we can make it more compact by replacing \mathbf{t}_{D+1} by $\mathbf{t}'_{D+1} = \mathbf{t}_{D+1} - \mathbf{x}$ where $\mathbf{x} = \text{BabaiReduce}([\mathbf{t}_1 \ \dots \ \mathbf{t}_D], \mathbf{t}_{D+1})$. The following lemma states that the resulting \mathbf{t}'_{D+1} is nearly optimal if \mathbf{t}_1 is short.

Lemma 5.13. Let \mathbf{T} be as in Theorem 5.3. Denote $\mathbf{A} = [\mathbf{t}_1 \ \cdots \ \mathbf{t}_D]$ and let

$$\mathbf{t}'_{D+1} = \mathbf{t}_{D+1} - \text{BabaiReduce}(\mathbf{A}, \mathbf{t}_{D+1}).$$

Then, we have

$$\|\mathbf{t}'_{D+1}\| \leq \sqrt{\frac{d}{4} \cdot D \cdot \|\tilde{\mathbf{t}}_1\|^2 + \|\tilde{\mathbf{t}}_{D+1}\|^2}.$$

Proof. Denote $\mathbf{y} = \mathbf{t}_{D+1}$. First, suppose that \mathbf{y} is in the K -span of the columns of \mathbf{A} . Furthermore, let $\mathbf{x} = \text{BabaiReduce}(\mathbf{A}, \mathbf{y})$. We want to estimate the distance $\|\mathbf{x} - \mathbf{y}\|$. This is similar to the analysis of [37]: without loss of generality, we may rotate our coordinate axes such that \mathbf{A} is of form

$$\begin{bmatrix} \|\tilde{\mathbf{t}}_1\| & * & \cdots & * \\ & \|\tilde{\mathbf{t}}_2\| & \cdots & * \\ & & \ddots & \vdots \\ & & & \|\tilde{\mathbf{t}}_D\| \end{bmatrix}$$

where $*$ represents a potentially non-zero element. From this, we quite easily obtain that, for every $i \in [D]$, the i th component of the difference satisfies

$$\|x_i - y_i\| \leq \frac{\sqrt{d}}{2} \|\tilde{\mathbf{t}}_i\|. \quad (5.2)$$

Only the coefficient arising from the rounding error (as given in Section 2.4.7) differs from the analysis of [37], the rest of the details are similar. Equation (5.2) yields

$$\|\mathbf{x} - \mathbf{y}\|^2 \leq \frac{d}{4} \sum_{i \in [D]} \|\tilde{\mathbf{t}}_i\|^2 \leq \frac{d}{4} D \|\tilde{\mathbf{t}}_1\|^2 \quad (5.3)$$

Now, let us see what happens if \mathbf{y} is not in the K -span of the columns of \mathbf{A} . Look at the algorithm BabaiReduce; observe that for target \mathbf{y} , the only part that contributes to the coefficients c_i (and hence the output) is the part that lies in the column span of \mathbf{A} . Therefore, $\text{BabaiReduce}(\mathbf{A}, \mathbf{t}_{D+1}) = \text{BabaiReduce}(\mathbf{A}, \text{proj}_{\text{span}_K\{\mathbf{t}_1, \dots, \mathbf{t}_D\}}(\mathbf{t}_{D+1}))$. Applying equation (5.3) and using properties of the orthogonal projection imply that

$$\|\mathbf{t}'_{D+1}\|^2 \leq \frac{d}{4} \cdot D \cdot \|\tilde{\mathbf{t}}_1\|^2 + \|\tilde{\mathbf{t}}_{D+1}\|^2.$$

□

Remark 5.14. Suppose that \mathbf{x} is the closest point to \mathbf{t}_{D+1} in $\text{span}_{\mathcal{R}}\{\mathbf{t}_1, \dots, \mathbf{t}_D\}$; obviously, $\|\mathbf{x} - \mathbf{t}_{D+1}\| \geq \|\tilde{\mathbf{t}}_{D+1}\|$. In this sense, the bound given in Lemma 5.13 is good in the case where $\|\mathbf{t}_1\|$ is small. In practice, this is not a particularly exciting setting. This is due to the behavior of the expected norm of $\tilde{\mathbf{t}}_{D+1}$ as a function of $\|\mathbf{t}_1\|$ (which we studied in Sections 5.3); the former decreases as the latter grows.

However, we concluded that the optimal $\|\mathbf{t}_1\|$ was such that we expect $\|\mathbf{t}_1\| \approx \|\tilde{\mathbf{t}}_{D+1}\|$. In this case, the lemma implies that the norm of \mathbf{t}'_{D+1} will be of a similar order of magnitude as $\|\mathbf{t}_1\|$.

Remark 5.15. Let \mathbf{t}_{D+1} be the last column of \mathbf{B} before Babai's reduction, $\mathbf{x} = \text{BabaiReduce}([\mathbf{t}_1 \ \dots \ \mathbf{t}_D], \mathbf{t}_{D+1})$, and $\mathbf{t}'_{D+1} = \mathbf{t}_{D+1} - \mathbf{x}$ be the last column after reduction. Then, since $\mathbf{x} \in \text{span}_K\{\mathbf{t}_0, \dots, \mathbf{t}_D\}$,

$$\begin{aligned}\tilde{\mathbf{t}}'_{D+1} &= \mathbf{t}'_{D+1} - \text{proj}_{\text{span}_K\{\mathbf{t}_0, \dots, \mathbf{t}_D\}}(\mathbf{t}'_{D+1}) \\ &= (\mathbf{t}_{D+1} - \mathbf{x}) - \text{proj}_{\text{span}_K\{\mathbf{t}_0, \dots, \mathbf{t}_D\}}(\mathbf{t}_{D+1} - \mathbf{x}) = \tilde{\mathbf{t}}_{D+1},\end{aligned}$$

i.e. the reduction does not change the Gram-Schmidt norm of the matrix and therefore does not affect the shortness of the preimages. This is a trivial observation, but it underlines the fact that Babai's reduction is done solely to reduce the memory needed to store the trapdoor.

5.5 vSIS trapdoor sampling algorithm

Now, we have acquired all the details needed for sampling vSIS trapdoors. The following definition demonstrates this.

Definition 5.16 (vSIS trapdoor generation). Let K be a power-of-2 cyclotomic field parameterized by λ and \mathcal{R} be its ring of integers. Also, let $D, q, \text{GS_SLACK}$ be functions of λ and χ be an NTRU distribution over \mathcal{R}_q , (also parameterized by λ). We define the vSIS trapdoor generation algorithm `TrapGen` as follows.

```
TrapGen( $1^\lambda$ )
repeat
   $f, g \leftarrow \chi$ 
until  $g \in \mathcal{R}_q^\times$ ,  $\text{gcd}(\mathcal{N}(f), \mathcal{N}(g))^D$  divides  $q$  and  $\|\tilde{\mathbf{t}}_{D+1}\| \leq \text{GS\_SLACK} \cdot q^{\frac{1}{D+1}}$ 
//  $\|\tilde{\mathbf{t}}_{D+1}\|$  computed as in Lemma 5.8
 $v = fg^{-1} \pmod q$ 
 $(a_0, \dots, a_D) \leftarrow \text{TowerSolverR}(f, g, D)$ 
Define  $(\mathbf{t}_1, \dots, \mathbf{t}_{D+1})$  as in Thm. 5.3
 $\mathbf{t}'_{D+1} \leftarrow \text{BabaiReduce}([\mathbf{t}_1 \ \dots \ \mathbf{t}_D], \mathbf{t}_{D+1})$ 
 $\text{td} := (f, g, \mathbf{t}'_{D+1})$  // uniquely determines the trapdoor matrix
return  $(v, \text{td})$ 
```

Remark 5.17. In section 5.3.2, we discussed the selection of the parameter `GS_SLACK`. There is a tradeoff: for `TrapGen` to be efficient, `GS_SLACK` needs to be large enough. However, a smaller value means that we can sample shorter preimages.

5.6 Hiding the NTRU secret

In our trapdoor construction, sharing the trapdoor basis \mathbf{T} with someone also leaks the NTRU secret, i.e. the short f, g such that $v = f/g \pmod q$. This gives the other party

more power than what is necessary. Indeed, we would like them to be able to sample short, degree- D vanishing polynomials. Yet, knowing f, g , they essentially have access to a vanishing linear function. In some settings this might not be a problem, but there could be other cases where this is not desirable. Therefore, let us demonstrate how to avoid that by exploiting the notion of “gadget matrix”. It was introduced by Micciancio and Peikert in [53] and has appeared regularly in the lattice-based cryptography literature ever since.

To provide some background, let us explain the concept of binary expansion in \mathcal{R} . Let $q \geq 2$ be an integer and recall that any element in \mathbb{Z}_q can be uniquely represented as a sequence of $\lceil \log q \rceil$ bits. We refer to this mapping from \mathbb{Z}_q to $\mathbb{Z}_2^{\lceil \log q \rceil}$ as the binary expansion, and denote the binary expansion operator as $(\mathbf{g}^T)^{-1}$ (for reasons that will become evident soon). We use the convention of writing the least significant bit first; for example, the binary expansion of the element $6 \in \mathbb{Z}_8$ is given by

$$(\mathbf{g}^T)^{-1}(6) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

To generalize the previous definition, we utilize the fact that $\mathcal{R} \cong \mathbb{Z}[X_1, \dots, X_n]/\mathfrak{p}$ for some $n \in \mathbb{N}$ and a prime ideal $\mathfrak{p} \subseteq \mathbb{Z}[X_1, \dots, X_n]$; thus, $\mathcal{R}_q \cong \mathbb{Z}_q[X_1, \dots, X_n]/\mathfrak{p}$. As a result, we can define the binary expansion

$$(\mathbf{g}^T)^{-1} : \mathcal{R}_q \rightarrow \mathcal{R}_2^{\lceil \log q \rceil}$$

by first mapping r to $\mathbb{Z}_q[X_1, \dots, X_n]/\mathfrak{p}$, applying the $(\mathbf{g}^T)^{-1}$ over \mathbb{Z}_q to each element and mapping the result back from $(\mathbb{Z}_2[X_1, \dots, X_n]/\mathfrak{p})^{\lceil \log q \rceil}$ to $\mathcal{R}_2^{\lceil \log q \rceil}$.

We define the gadget matrix as

$$\mathbf{g}^T = [1 \quad 2 \quad \dots \quad 2^{\lceil \log q \rceil - 1}] \in \mathcal{R}_q^{\lceil \log q \rceil}.$$

Observe that it gives a linear mapping from a binary expansion of a ring element to the ring element itself: for any $r \in \mathcal{R}_q$ it holds that $\mathbf{g}^T (\mathbf{g}^T)^{-1}(r) = r$. Conversely, for any $\mathbf{r} \in \mathcal{R}_2^{\lceil \log q \rceil}$ we have $(\mathbf{g}^T)^{-1}(\mathbf{g}^T \mathbf{r}) = \mathbf{r}$.

The gadget matrix has a nice property: both SIS and LWE are easy with respect to \mathbf{g}^T , as observed in [53]. We are mainly interested in SIS, so let us focus on that however. To show that SIS is easy, notice that $\Lambda_q^\perp(\mathbf{g}^T)$ has a short basis, given by

$$\mathbf{T}_{\mathbf{g}^T} = \begin{bmatrix} 2 & & & & q_0 \\ -1 & 2 & & & q_1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & 2 & q_{\lceil \log q \rceil - 2} \\ & & & -1 & q_{\lceil \log q \rceil - 1} \end{bmatrix}$$

where

$$(q_0, \dots, q_{\lceil \log q \rceil - 1}) = \begin{cases} (0, \dots, 0, 2), & q \text{ is a power of } 2 \\ (\mathbf{g}^T)^{-1}(q), & \text{otherwise.} \end{cases}$$

Equipped with this machinery, we are ready to tackle the problem of sharing a vSIS trapdoor without leaking the NTRU secret. The idea is rather simple: we use the trapdoor \mathbf{T} to sample a short random $\mathbf{X} \in \mathcal{R}^{(D+1) \times \lceil \log q \rceil}$ that satisfies

$$\begin{bmatrix} v^D & \cdots & v & 1 \end{bmatrix} \mathbf{X} = \mathbf{g}^T \pmod{q}.$$

This can be done one column at a time using standard techniques, e.g., GPV sampling. Then, instead of sharing \mathbf{T} , we give out \mathbf{X} to the other party; the properties of the trapdoor sampling guarantee that this does not leak f, g . When asked to find \mathbf{p} such that $\begin{bmatrix} v^D & \cdots & v & 1 \end{bmatrix} \mathbf{p} = t \pmod{q}$ (for some $t \in \mathcal{R}$), they can first use $\mathbf{T}_{\mathbf{g}^T}$ to sample \mathbf{p}' such that $\mathbf{g}^T \mathbf{p}' = t \pmod{q}$. They subsequently compute $\mathbf{p} = \mathbf{X} \mathbf{p}' \pmod{q}$ and return that. We can check that

$$\begin{bmatrix} v^D & \cdots & v & 1 \end{bmatrix} \mathbf{p} = \begin{bmatrix} v^D & \cdots & v & 1 \end{bmatrix} \mathbf{X} \mathbf{p}' = \mathbf{g}^T \mathbf{p}' = t \pmod{q}.$$

5.7 Trapdoors for multivariate vSIS

In the previous sections, we only considered univariate trapdoors. Hence, the goal of this section is to discuss how this could be generalized to construct trapdoors for multivariate vSIS. As it will turn out, a lot of what was discussed earlier seems to still apply.

However, we face some new technical difficulties. For univariate vSIS of degree D , the first $D - 1$ columns of the trapdoor basis \mathbf{T} form a lower bidiagonal Toeplitz matrix. This greatly simplifies the computations of the determinant. We lose this structure when introducing new variables, which makes it difficult to state any general results. Let us therefore study a simple example.

Example 5.18 (Trapdoors for degree-2 bivariate vSIS). Let $q \in \mathbb{N}$, $v_1, v_2 \in \mathcal{R}$ and an index set $J = \{(i, j) \in \mathbb{Z}_{\geq 0}^2 \mid i + j \leq 2\}$. Define the degree-2 vSIS module

$$\mathcal{M} = \left\{ (p_{i,j})_{(i,j) \in J} \mid \sum_{(i,j) \in J} p_{i,j} v_1^i v_2^j = 0 \pmod{q} \right\}.$$

In the lexicographic order (the monomial basis ordered as $(X^2, X_1 X_2, X_1, X_2^2, X_2, 1)$), \mathcal{M} is generated by the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -v_1 & -v_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -v_2 & 1 & 0 \\ 0 & 0 & -v_1 & 0 & -v_2 & q \end{bmatrix}.$$

This can be verified using a similar argument as in the proof of Lemma 5.2: the first

inclusion is simple and the reverse inclusion follows by choosing

$$\mathbf{x} = \begin{bmatrix} p_{2,0} \\ p_{1,1} \\ p_{2,0}v + p_{1,1}w + p_{1,0} \\ p_{0,2} \\ p_{0,2}w + p_{0,1} \\ r \end{bmatrix}$$

where $r \in \mathcal{R}$ satisfies $p_{i,j}v_1^i v_2^j = rq$.

If we assume that for $i \in \{1, 2\}$, $v_i = f_i/g_i \bmod q$ for some short ring elements f_i, g_i , another basis (the trapdoor basis) is then given by

$$\mathbf{T} = \begin{bmatrix} g_1 & 0 & 0 & 0 & 0 & a_0 \\ 0 & g_2 & 0 & 0 & 0 & a_1 \\ -f_1 & -f_2 & g_1 & 0 & 0 & a_2 \\ 0 & 0 & 0 & g_2 & 0 & a_3 \\ 0 & 0 & 0 & -f_2 & g_2 & a_4 \\ 0 & 0 & -f_1 & 0 & -f_2 & a_5 \end{bmatrix}$$

if the a_i satisfy

$$\det(\mathbf{T}) = a_0 f_1^2 g_2^3 + a_1 f_1 f_2 g_1 g_2^2 + a_2 f_1 g_1 g_2^3 + a_3 f_2^2 g_1^2 g_2 + a_4 f_2 g_1^2 g_2^2 + a_5 g_1^2 g_2^3 = q. \quad (5.4)$$

This follows from the fact that the change-of-basis matrix $\mathbf{A}^{-1}\mathbf{T}$ has its elements in the ring and its determinant is equal to 1; both of these properties can be shown by a routine calculation.

Let us discuss how good the trapdoor basis is. First of all, observe that

$$\max_{i \in \{1,2,3,4,5\}} \{\|\tilde{\mathbf{t}}_i\|\} \leq \max\{\|\mathbf{t}_1\|, \|\mathbf{t}_2\|\}.$$

This is because $\mathbf{t}_3, \mathbf{t}_4$ and \mathbf{t}_5 have their norm bounded by $\max\{\|\mathbf{t}_1\|, \|\mathbf{t}_2\|\}$ and, similarly as in the proof of Lemma 5.7, GSO cannot increase their norm. This directly implies

$$\|\mathbf{T}\|_{\text{GS}} = \max\{\|\mathbf{t}_1\|, \|\mathbf{t}_2\|, \|\tilde{\mathbf{t}}_6\|\}.$$

$\|\tilde{\mathbf{t}}_1\|$ and $\|\tilde{\mathbf{t}}_2\|$ are easy to bound due to a priori bound $\|f_i\|, \|g_i\| \leq \beta$. However, an expression for $\tilde{\mathbf{t}}_6$ in terms of q and f_1, f_2, g_1, g_2 exists, but is far from being as simple as the one we derived in Lemma 5.8. What we do have, similarly as in Lemma 5.10, is a simple expression for $\|\tilde{\mathbf{t}}_6\|$ in terms of $\|\tilde{\mathbf{t}}_i\|$ for $i \in \{1, \dots, 5\}$. This also gives a lower bound:

$$\|\tilde{\mathbf{t}}_6\| = \frac{q}{\prod_{i=1}^5 \|\tilde{\mathbf{t}}_i\|} \geq \frac{q}{\max\{\|\mathbf{t}_1\|, \|\mathbf{t}_2\|\}^5}. \quad (5.5)$$

Also, observe that the Babai's reduction algorithm as described in Definition 5.12 is general enough to apply directly also in the multivariate case.

Remark 5.19. Let us take another look at equation (5.4). It illustrates another new difficulty: g_2 divides $\det(\mathbf{T})$. This implies that $\mathcal{N}(g_2)$ divides $\gcd(\mathcal{N}(f_1^2 g_2^3), \dots, \mathcal{N}(g_1^2 g_2^3))$. Thus, if $\mathcal{N}(g_2)$ does not divide q (which is true with a high probability for a randomly sampled g_2), we cannot use EEA to solve for (a_0, \dots, a_5) . The problem is not specific to using the field norm approach of [46]; due to multiplicativity of the resultants, we would face it even if we were to use the traditional resultant-based algorithm. Currently, we do not know how to work around this without negatively affecting the security (for example, taking g_2 to be a random unit would solve the problem but also ruin the distribution of v_2). Therefore, it limits the practical use of multivariate vSIS trapdoors.

Notice that working through even a small example was laborious without the bidiagonal structure of the basis matrix. This is the reason why deriving general results for multivariate vSIS trapdoors seems to be difficult. However, one idea would be to try to find a favorable permutation of the rows and columns; i.e. finding permutation matrices \mathbf{P} , \mathbf{P}' such that \mathbf{PAP}' and \mathbf{PTP}' would have a nice structure. However, it is unclear whether this is possible or how it would be done.

Another approach stems from the observation that, compared to the univariate case, there is some additional freedom when picking a “natural” basis for the module. For instance, in the above example one could equivalently choose \mathbf{A} to be

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -v_1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -v_1 & 0 & -v_2 & 1 & 0 \\ 0 & 0 & -v_1 & 0 & -v_2 & q \end{bmatrix}.$$

Surprisingly, also

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -v_1^2 & -v_1 v_2 & -v_1 & -v_2^2 & -v_2 & q \end{bmatrix}$$

gives a basis for the module. The last one is by far the simplest option structure-wise. We also have a trapdoor basis with similar structure:

$$\mathbf{T} = \begin{bmatrix} g_1^2 & 0 & 0 & 0 & 0 & a_0 \\ 0 & g_1 g_2 & 0 & 0 & 0 & a_1 \\ 0 & 0 & g_1 & 0 & 0 & a_2 \\ 0 & 0 & 0 & g_2^2 & 0 & a_3 \\ 0 & 0 & 0 & 0 & g_2 & a_4 \\ -f_1^2 & -f_1 f_2 & -f_1 & -f_2^2 & -f_2 & a_5 \end{bmatrix}.$$

Choosing \mathbf{A}, \mathbf{T} as above would make the theory significantly easier to develop. There is also a downside, though. Since the first 5 columns of \mathbf{T} contain higher powers and cross-terms, one could expect $\|\mathbf{T}\|_{\text{GS}}$ to be large. On the other hand, there also exists a counterargument. Recall equation (5.5): as $\|\tilde{\mathbf{t}}_i\|$ for $i \in \{1, \dots, 5\}$ increase, $\|\tilde{\mathbf{t}}_6\|$ decreases. This may cancel the growth of $\|\mathbf{T}\|_{\text{GS}}$ at least to some extent. Running numerical experiments could show which of the effects is more prevalent, but that turns out to be hard due to what was discussed in Remark 5.19.

6 Applications

6.1 Replacing SIS trapdoors

There are plenty of existing constructions that are based on SIS trapdoors. To name a few examples, consider the digital signatures and identity-based encryption of [38] and the key-policy attribute-based encryption of [54]. Such schemes can be modified to the vSIS setting in an extremely straight-forward manner: replace the SIS problem by the vSIS problem and use vSIS trapdoors instead of the SIS trapdoors. The security reductions of the schemes are similarly simple to modify by using the vSIS assumption instead of the SIS assumption.

The rationale behind doing this is mainly to build more efficient schemes. First of all, over SIS or ring-SIS we achieve significantly more compact public parameters. The SIS-based schemes feature the public matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, and likewise the schemes based on ring-SIS feature a vector of ring elements, $\mathbf{a} \in \mathcal{R}^m$. If we use single-point vSIS instead, the problem instance can be described by a single ring element $v \in \mathcal{R}$. This saves a factor of m over ring-SIS and even more over SIS.

NTRU trapdoors are similar to vSIS trapdoors in the sense that the public parameters are compact. However, the Gram-Schmidt norm of an NTRU trapdoor, $\|\mathbf{T}\|_{\text{GS}}$, is limited to being of order \sqrt{q} . Considering higher degree polynomials (that is, vSIS), we can achieve a significantly better $\|\mathbf{T}\|_{\text{GS}}$, as shown in Figure 3. The main benefit of this is that we can sample shorter preimages without leaking information about the trapdoor, which results in better efficiency.

Module-NTRU trapdoors [49, 48] are another type that provides a Gram-Schmidt norm similar to that of vSIS trapdoors. However, to achieve a norm of order $q^{1/(D+1)}$ with module-NTRU trapdoors, the public parameters need to include D ring elements, which is worse than what we have for vSIS trapdoors.

6.2 Single-data homomorphic signatures from vSIS trapdoors

The extent of what we can do with vSIS trapdoors does not seem to be limited to simply replacing SIS trapdoors in existing schemes. Indeed, observe that vanishing polynomials have a peculiar property: they allow both homomorphic addition and multiplication.

This is not a given; to illustrate that, let us consider a standard SIS trapdoor. Let \mathbf{A} be a matrix in $\mathbb{Z}^{n \times m}$; a trapdoor with respect to \mathbf{A} allows us, given $\mathbf{y} \in \mathbb{Z}^n$, to compute short preimages $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$. These preimages are additively homomorphic, i.e. if in addition we have $\mathbf{A}\mathbf{x}' = \mathbf{y}' \bmod q$, then $\mathbf{A}(\mathbf{x} + \mathbf{x}') = \mathbf{y} + \mathbf{y}' \bmod q$. However, the same does not hold for (coefficient-wise) multiplication. The situation is similar with ring-SIS.

Compare this to vSIS: suppose we have sampled short, degree- D polynomials $f_1, f_2 \in \mathcal{R}[X]$ such that $f_i(v) = r_i$ for $i \in \{1, 2\}$. We clearly have $(f_1 + f_2)(v) = r_1 + r_2$ but also $(f_1 f_2)(v) = r_1 r_2$. Furthermore, the constant coefficients behave nicely: $(f_1 + f_2)(0) = f_1(0) + f_2(0)$ and $(f_1 f_2)(0) = f_1(0) f_2(0)$, and exactly the same

happens with the leading coefficients as well. Finally, $(f_1 + f_2)$ and $(f_1 f_2)$ are somewhat short, and we can easily bound their degree.

These observations motivate us to consider applying vSIS trapdoors for homomorphic computation. There could be other applications as well, but a homomorphic signature (HS) scheme seems like a natural choice.

6.2.1 Homomorphic signatures

Let us begin by defining the notion of a single-data HS scheme. Throughout this section we will utilize the definitions from [23].

Definition 6.1 (Single-data HS). *A single-data homomorphic signature (HS) scheme for message space \mathcal{X} consists of a tuple of algorithms (PrmsGen, KeyGen, Sign, Eval, Process, Verify) with the following syntax.*

- $\text{pp} \leftarrow \text{PrmsGen}(1^\lambda, 1^N)$: Gets the security parameter λ and a data-size bound N . Generates the public parameters pp .
- $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda, \text{pp})$: Gets the security parameter along with the public parameters. Generates the public key and the secret key.
- $(\mathbf{s}_1, \dots, \mathbf{s}_N) \leftarrow \text{Sign}_{\text{sk}}(x_1, \dots, x_N)$: Signs a tuple of data $(x_1, \dots, x_N) \in \mathcal{X}^N$.
- $\mathbf{s}^* \leftarrow \text{Eval}_{\text{pp}}(g, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$: Homomorphically evaluates function g , outputting a new signature \mathbf{s}^* .
- $\alpha_g \leftarrow \text{Process}_{\text{pp}}(g)$: Computes a “public key” of g that is later used in the verification step.
- $b \leftarrow \text{Verify}_{\text{pk}}(\alpha_g, y, \mathbf{s}^*)$: Uses the signature \mathbf{s}^* to check that y is equal to $g(x_1, \dots, x_l)$; outputs 1 if that is the case, 0 otherwise.

Remark 6.2. Some authors choose to merge the definitions of PrmsGen and KeyGen together, as well as Process and Verify. We will define them separately, similar to the definitions used in [23]. The benefit of doing this is that some schemes provide what we call *efficient verification with preprocessing* in some schemes. This means that generating $\alpha_g \leftarrow \text{Process}_{\text{pp}}(g)$ might be expensive but Verify runs fast. Since α_g does not depend on the data (x_1, \dots, x_l) , this can be beneficial in some situations – especially when considering multi-data schemes.

To explain the essence of a HS scheme, in addition to the syntax we also need to describe what these algorithms are supposed to do. More concretely, we need to define what we mean by correctness and security. Towards that, we introduce the concept of an *admissible function*.

In the schemes that we are studying, each signature \mathbf{s} possesses a property called *noise-level* and denoted by β . A scheme is parameterised by the maximal initial noise level for fresh signatures produced by Sign, and the maximal amount of noise that a signature may have while still being verifiable. We will denote these β_{init}

and β_{\max} . We say that a function g is admissible on (x_1, \dots, x_l) if the noise of $\mathbf{s}^* = \text{Eval}_{\text{pp}}(g, ((x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l)))$ is at most β_{\max} whenever all \mathbf{s}_i have noise level at most β_{init} .

Remark 6.3. In the context of this thesis we assume that β_{\max} is fixed. Such schemes fall under the (quite generic) class of *somewhat homomorphic signature* schemes. In [23], β_{\max} can be arbitrary but must be set at the initialization phase; this makes the construction a *leveled fully-homomorphic signature (FHS)*.

Now we are ready to present the correctness requirements and two different security notions.

Definition 6.4 (Correctness, single-data). Let Σ be a single-data HS scheme as defined in Definition 6.1. We say that Σ is correct if it satisfies the following two notions for any $(\text{pk}, \text{sk}) \in \text{KeyGen}(1^\lambda, 1^N)$:

- *Signing correctness.* Let $\text{id}_i : \mathcal{X}^N \rightarrow \mathcal{X}$ be defined as the canonical extension of $\text{id}_i(x_1, \dots, x_N) = x_i$. We require that for any $i \in [N]$ and $(x_1, \dots, x_N) \in \mathcal{X}^N$,

$$\text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(\text{id}_i), x_i, \mathbf{s}_i) = 1.$$

- *Evaluation correctness.* For any h_1, \dots, h_l with $h_i : \mathcal{X}^N \rightarrow \mathcal{X}$ and any $g : \mathcal{X}^l \rightarrow \mathcal{X}$, define the composition $\bar{g} : \mathcal{X}^N \rightarrow \mathcal{X}$ as $\bar{g} := (g \circ (h_1, \dots, h_l))$. We require that if \bar{g} is admissible on (x_1, \dots, x_l) , then for any $(x_1, \dots, x_l) \in \mathcal{X}^l$ and $(\mathbf{s}_1, \dots, \mathbf{s}_l)$

$$\text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(h_i), x_i, \mathbf{s}_i) = 1 \quad \forall i \in [l]$$

implies

$$\text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(\bar{g}), g(x_1, \dots, x_l), \mathbf{s}^*) = 1$$

if \mathbf{s}^* is generated as $\mathbf{s}^* \leftarrow \text{Eval}_{\text{pp}}(g, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$.

Remark 6.5. The above definition of evaluation correctness is quite general since it we require correctness of compositions. A more intuitive notion is to consider *evaluation correctness of a single function*: for any $(\text{pk}, \text{sk}) \in \text{KeyGen}(1^\lambda, 1^N)$, $(x_1, \dots, x_N) \in \mathcal{X}^N$, $(\mathbf{s}_1, \dots, \mathbf{s}_N) \in \text{Sign}_{\text{sk}}(x_1, \dots, x_N)$, we require that

$$\text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(g), g(x_1, \dots, x_N), \mathbf{s}^*) = 1$$

if \mathbf{s}^* is generated as $\mathbf{s}^* \leftarrow \text{Eval}_{\text{pp}}(g, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$ and $g : \mathcal{X}^l \rightarrow \mathcal{X}$ is admissible on (x_1, \dots, x_l) . This is a weaker notion than evaluation correctness since it is implied by signing and evaluation correctness.

Definition 6.6 (Security, single data). Let N be a function of λ . We define the security of single-data HS via the following security games:

Selective-security $_{\Sigma, \mathcal{A}}(1^\lambda)$	Full-security $_{\Sigma, \mathcal{A}}(1^\lambda)$
$(x_1, \dots, x_N) \leftarrow \mathcal{A}$	$\text{pp} \leftarrow \text{PrmsGen}(1^\lambda, 1^N)$
$\text{pp} \leftarrow \text{PrmsGen}(1^\lambda, 1^N)$	$(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda, \text{pp})$
$(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda, \text{pp})$	$(x_1, \dots, x_N) \leftarrow \mathcal{A}(\text{pp}, \text{pk})$
$(\mathbf{s}_1, \dots, \mathbf{s}_N) \leftarrow \text{Sign}_{\text{sk}}(x_1, \dots, x_N)$	$(\mathbf{s}_1, \dots, \mathbf{s}_N) \leftarrow \text{Sign}_{\text{sk}}(x_1, \dots, x_N)$
$(g, y, \mathbf{s}) \leftarrow \mathcal{A}(\text{pp}, \text{pk}, (\mathbf{s}_1, \dots, \mathbf{s}_N))$	$(g, y, \mathbf{s}) \leftarrow \mathcal{A}((\mathbf{s}_1, \dots, \mathbf{s}_N))$
$b_0 = \text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(g), y, \mathbf{s})$	$b_0 = \text{Verify}_{\text{pk}}(\text{Process}_{\text{pp}}(g), y, \mathbf{s})$
$b_1 = (g(x_1, \dots, x_N) \neq y)$	$b_1 = (g(x_1, \dots, x_N) \neq y)$
$b_2 = \text{"}g \text{ is admissible on } (x_1, \dots, x_N)\text{"}$	$b_2 = \text{"}g \text{ is admissible on } (x_1, \dots, x_N)\text{"}$
return $b_0 \wedge b_1 \wedge b_2$	return $b_0 \wedge b_1 \wedge b_2$

An HS scheme is said to be *selectively secure* if, for any polynomially bounded adversary \mathcal{A} , the probability $\Pr [\text{Selective-security}_{\Sigma, \mathcal{A}}(1^\lambda) = 1]$ is negligible in λ . The scheme is called *fully secure* if the same holds for $\Pr [\text{Full-security}_{\Sigma, \mathcal{A}}(1^\lambda) = 1]$.

Observe that the difference between the two security games is that in the first one the adversary \mathcal{A} is required to declare the challenge messages (x_1, \dots, x_N) before the generation of pk and sk . In the second one \mathcal{A} can decide after seeing pk . Since the latter offers the adversary more freedom without anything else changing, full security is a strictly stronger notion than selective security.

To motivate why HS schemes can be useful, we consider an example application in certified data analysis, as proposed in [23]. Suppose that there is a trusted authority (a government institute, for example) who has some data $\mathbf{x} = (x_1, \dots, x_N)$. Furthermore, assume that the authority provides a researcher access to \mathbf{x} in order for them to conduct analysis on it (in our model, evaluate a function g) and publish the result, y , along with the methodology used, g , to the public.

We might not trust the researcher; they may have incentives to lie about the outcome y . This is not a problem for the authority since they can easily check if $g(x_1, \dots, x_N)$ is indeed equal to y . However, assume that they do not want to publish the data (for example, it might be too large to be conveniently distributed). Then, a third party (e.g. a member of the public) does not have any way of checking if the researcher lied or not.

This is where homomorphic signatures come in: in addition to sending the researcher \mathbf{x} , the authority sends $(\mathbf{s}_1, \dots, \mathbf{s}_N)$, the signature of \mathbf{x} . The researcher can then homomorphically evaluate g on $(\mathbf{s}_1, \dots, \mathbf{s}_N)$ to compute a homomorphic signature $\mathbf{s}_{(g,y)}$ and publish this along with y and g . These, along with the public key (generated by the authority), provide the third party the means to verify if the result is to be trusted.

6.2.2 vSIS-based construction

We propose a HS construction that is essentially an adaptation of the SIS-based leveled FHS scheme of Gorbunov, Vaikuntanathan, and Wichs [23] to the vSIS context. For

$N \in \mathbb{N}$, let the public key be a tuple $(v, (r_i)_{i \in [N]}) \in \mathcal{R} \times \mathcal{R}_q^N$ and $(x_i)_{i \in [N]} \in \mathcal{R}^N$ be a short message. Then, given a degree- $(D-1)$ trapdoor with respect to v , we can sample N degree- D polynomials $(f_i)_{i \in [N]}$ such that for all $i \in [N]$, $\|f_i\| \leq \beta$ and

$$\begin{cases} f_i(v) = r_i \pmod{q} \\ f_i(0) = x_i \pmod{q}. \end{cases}$$

The tuple $(f_i)_{i \in [N]}$ can now be viewed as a signature of the message $(x_i)_{i \in [N]}$. Let $l \leq N$; given an l -variate polynomial g , anyone can now derive a somewhat short, bounded-degree polynomial $h = g \circ (f_1, \dots, f_l)$ that satisfies

$$\begin{cases} h(v) = g(r_1, \dots, r_l) \pmod{q} \\ h(0) = g(x_1, \dots, x_l) \pmod{q}, \end{cases} \quad (6.1)$$

i.e. h is a signature of $g(x_1, \dots, x_l)$.

We use the following lemma to bound the degree and the norm of h .

Lemma 6.7. *For $i \in [l]$, let $f_i \in \mathcal{R}[X]$ such that $\|f_i\| \leq \beta$. Moreover, let $g \in \mathcal{R}[X_1, \dots, X_l]$ and $h = g \circ (f_1, \dots, f_l)$. Then,*

$$\deg(h) \leq \deg(g) \cdot \max_i \{\deg(f_i)\}$$

and

$$\|h\| \leq \binom{\deg(g) + l}{l} \cdot \|g\| \cdot \beta^{\deg(g)} \cdot \nu_{1, \deg(f), \deg(g)} \cdot \gamma_K^{\deg(g)},$$

where $\nu_{1, \deg(f), \deg(g)}$ is as in Definition 2.10.

Proof. Denote $f_i = \sum_j f_{ij} X^j$ and $g = \sum_\alpha g_\alpha X^\alpha$. Then,

$$h = \sum_\alpha g_\alpha (f_1, \dots, f_l)^\alpha = \sum_\alpha g_\alpha \left(\sum_j f_{1j} X^j \right)^{\alpha_1} \cdots \left(\sum_j f_{lj} X^j \right)^{\alpha_l}$$

The degree bound is immediate when h is written in this form. Furthermore, if n is the number of non-zero terms in g , we can bound the norm as

$$\begin{aligned} \|h\| &\leq n \cdot \max_\alpha \{g_\alpha\} \cdot \max_\alpha \left\{ \left(\sum_j f_{1j} X^j \right)^{\alpha_1} \cdots \left(\sum_j f_{lj} X^j \right)^{\alpha_l} \right\} \\ &\leq n \cdot \|g\| \cdot \gamma_K \cdot \max_\alpha \left\{ \left(\beta \sum_j X^j \right)^{\alpha_1} \cdots \left(\beta \sum_j X^j \right)^{\alpha_l} \right\} \\ &\leq n \cdot \|g\| \cdot \gamma_K \cdot \left(\beta \sum_j X^j \right)^{\deg(g)} \leq n \cdot \|g\| \cdot \beta^{\deg(g)} \cdot \nu_{1, \deg(f), \deg(g)} \cdot \gamma_K^{\deg(g)}. \end{aligned}$$

By Corollary 2.9 we know that $n \leq \binom{\deg(g)+l}{l}$. □

Verification of a signature h therefore simply amounts to checking that

$$\begin{cases} \deg(h) \leq \deg(g) \cdot D \\ \|h\| \leq \binom{\deg(g)+l}{l} \cdot \|g\| \cdot \beta^{\deg(g)} \cdot \nu_{1,D,\deg(g)} \cdot \gamma_K^{\deg(g)} \\ h(v) = g(r_1, \dots, r_l) \bmod q \end{cases} \quad (6.2)$$

Let us now write down the algorithms formally.

Definition 6.8. Let \mathcal{R} be a ring of integers and $\mathcal{X} \subseteq \mathcal{R}$ be a message space, both parameterized by λ . We define a vSIS-based single-data HS scheme Σ_{vSIS} over message space \mathcal{X} , consisting of the following algorithms.

$\text{PrmsGen}(1^\lambda, 1^N)$ for $i \in [N]$ do $r_i \leftarrow \mathcal{R}_q$ $\text{pp} := (r_1, \dots, r_N)$ return pp	$\text{Eval}_{\text{pp}}(g, (x_1, f_1), \dots, (x_N, f_N))$ $h := g \circ (f_1, \dots, f_l)$ return h
$\text{KeyGen}(1^\lambda, \text{pp})$ $(v, \text{td}) \leftarrow \text{TrapGen}(1^\lambda)$ $(\text{pk}, \text{sk}) := (v, \text{td})$ return (pk, sk)	$\text{Process}_{\text{pp}}(g)$ $\alpha_g := (l, \ g\ , \deg(g), g(r_1, \dots, r_l) \bmod q)$ return α_g
$\text{Sign}_{\text{sk}}((x_i)_{i \in [N]})$ for $i \in [N]$ do $f'_i \leftarrow \text{SampPre}\left(\text{td}, \frac{r_i - x_i}{v}\right)$ $f_i := v \cdot f'_i + x_i$ return (f_1, \dots, f_N)	$\text{Verify}_{\text{pk}}(\alpha_g, y, h)$ parse α_g as $(l, \ g\ , \deg(g), z)$ $b_1 := (\deg(h) \leq \deg(g) \cdot D)$ $b_2 := (\ h\ \leq b_{g,\beta,D})$ $b_3 := (h(v) = z \bmod q)$ $b_4 := (h(0) = y)$ return $b_1 \wedge b_2 \wedge b_3 \wedge b_4$

Here, SampPre denotes a PPT algorithm that, given the target $(r_i - x_i)/v$, samples a preimage f'_i satisfying $f'_i(v) = (r_i - x_i)/v \bmod q$. It can be implemented using, for example, GPV sampling, as explained in Section 2.3.4.

Moreover, we used the shorthand $b_{g,\beta,D} = \binom{\deg(g)+l}{l} \cdot \|g\| \cdot \beta^{\deg(g)} \cdot \nu_{1,D,\deg(g)} \cdot \gamma_K^{\deg(g)}$.

Remark 6.9. In the above construction, we must require that v is a unit in \mathcal{R}_q . This means that we have to use a slightly modified version of the algorithm TrapGen . However, over a suitable choice of \mathcal{R} and q — for example, \mathcal{R} is the ring of integers of $K(\zeta_k)$ for k a power-of-2 and q a prime congruent to 1 mod k — the majority of elements in \mathcal{R}_q are units. Thus, neither the output distribution nor the efficiency changes drastically.

Remark 6.10. In our construction, the parameter y of Verify is redundant since it is equal to the constant coefficient of h . We keep it only to be somewhat consistent with the definitions of [23].

The above vSIS HS scheme is selectively secure under a certain vSIS assumption.

Lemma 6.11. *Let Σ_{vSIS} be as in Definition 6.8; then, Σ_{vSIS} is correct. Moreover, suppose $\beta, D \in \mathbb{N}$ such that all elements in \mathcal{X} have norm at most β and td allows sampling preimages of norm β and degree $D-1$. Then, if the admissible functions of Σ_{vSIS} consist of polynomials g satisfying $\|g\| \leq \beta'$ and $\deg(g) \leq D'$, the scheme is selectively secure under the $\mathcal{R}\text{-dLWE}_{1,q,\mathcal{X}}$ and $\text{vSIS}_{D^*,\beta^*}$ assumptions, where $D^* = D \cdot D'$,*

$$\beta^* = 2 \cdot \binom{D' + N}{N} \cdot \beta' \cdot \beta^{D'} \cdot \nu_{1,D,D'} \cdot \gamma_K^{D'}$$

and $\nu_{1,D,D'}$ is defined as in Definition 2.10.

Proof. For signing correctness, we require that $f_i(v) = r_i \bmod q$ for all i . Since f'_i is a preimage of $(r_i - x_i)/v$,

$$f_i(v) = v f'_i(v) + x_i = r_i \bmod q.$$

Lemma 6.7 directly implies that the first two conditions of **Verify** are satisfied. For evaluation correctness, observe that if we let $e : \mathcal{X}^l \rightarrow \mathcal{X}$ and g_i, h_i such that $g_i : \mathcal{X}^N \rightarrow \mathcal{X}$, $h_i(v) = g_i(r_1, \dots, r_N)$ for all $i \in [l]$, then

$$(e \circ (h_1, \dots, h_l))(v) = (e \circ (g_1, \dots, g_l))(r_1, \dots, r_N) \bmod q.$$

Thus, the third verification condition is satisfied; the first two are once again implied by Lemma 6.7, and checking the fourth one is trivial.

Towards the security proof, suppose that \mathcal{A} is a PPT adversary that breaks the selective security of the scheme. In particular, if \mathcal{A} has chosen messages $(x_i)_{i \in [N]}$ and received their signatures $(f_i)_{i \in [N]}$ along with $\text{pp} = (r_i)_{i \in [N]}$ and $\text{pk} = v$, suppose that it can (with a non-negligible probability) find a forgery $(g^*, h^*) \in (\mathcal{R}[X] \times \mathcal{R}[X_1, \dots, X_l])$ for some $l \leq N$. The forgery is such that it passes the verification but $g^*(x_1, \dots, x_l) \neq h^*(0)$. Then, we claim that the following reduction solves $\text{vSIS}_{D^*,\beta^*}$ with a non-negligible probability.

```

 $\mathcal{R}_q^{\mathcal{A}}(v)$ 


---


 $(x_1, \dots, x_l) \leftarrow \mathcal{A}()$ 
Sample degree- $D$   $f_1, \dots, f_N \in \mathcal{R}[X]$  with  $f_i(0) = x_i$  and the rest of the coefficients in  $\mathcal{X}$ 
 $v \leftarrow \mathcal{R}_q^{\times}$ 
for  $i \in [N]$  do
   $r_i := f_i(v) \bmod q$ 
 $(g^*, h^*) \leftarrow \mathcal{A}((f_i)_{i \in [N]}, v, (r_i)_{i \in [N]})$ 
 $p := h^* - g^* \circ (f_1, \dots, f_l)$ 
return  $p$ 

```

Observe that r_i can be written as $f_{i,D}v^D + \dots + f_{i,1}v + x_i$; by the $\mathcal{R}\text{-dLWE}_{1,q,\mathcal{X}}$ assumption, the term $f_{i,1}v$ is computationally indistinguishable from uniform in

\mathcal{R}_q . Since adding anything to a uniformly distributed term will result in a uniform distribution, we conclude that the distribution of $r_i = f_i(v)$ is also computationally indistinguishable from uniform.

Thus, we may assume that, with a non-negligible probability, (g^*, h^*) is a successful forgery. In that case,

$$\begin{cases} h^*(0) \neq g^*(x_1, \dots, x_l) \\ \|h^*\| \leq \binom{\deg(g^*)+l}{l} \cdot \|g^*\| \cdot \beta^{\deg(g^*)} \cdot \nu_{1,D, \deg(g^*)} \cdot \gamma_K^{\deg(g^*)} \\ \deg(h^*) \leq \deg(g^*) \cdot D \\ h^*(v) = g^*(r_1, \dots, r_l) \pmod q \end{cases} \quad (6.3)$$

where the last three conditions follow from the verification equation. The last of them implies

$$p(v) = h^*(v) - g^*(f_1(v), \dots, f_l(v)) = h^*(v) - g^*(r_1, \dots, r_l) = 0 \pmod q,$$

therefore p is a polynomial vanishing at v . Moreover, p is non-zero since

$$(g^* \circ (f_1, \dots, f_l))(0) = g^*(f_1(0), \dots, f_l(0)) = g^*(x_1, \dots, x_l) \neq h^*(0)$$

by the first condition of (6.3). Finally, by admissibility of g^* , Lemma 6.7 and the second and third conditions of (6.3), $\|p\|$ and $\deg(p)$ are sufficiently small. \square

6.3 From single-data to multi-data

The HS scheme that we constructed in the previous section is a single-data scheme, i.e. the signer can sign only one set of data (x_1, \dots, x_l) . This is limiting since in many applications we need to be able to sign several different data sets, possibly of different sizes. Moreover, we could only show that the scheme is selectively secure, as opposed to being fully secure. We will see how both of these limitations can be handled by using two generic transformations introduced in [23]. The idea of both transformations is to combine a regular (that is, non-homomorphic) signature scheme and a selectively secure single-data HS scheme.

Before going into the details, let us briefly overview the definition of multi-data HS, once again following the one given in [23]. The main addition, compared to a single-data scheme, is that each data set is given a label $\psi \in \{0, 1\}^*$. We assume that the verifier knows the label of the data set being verified.

Definition 6.12 (Multi-data HS). A *multi-data HS scheme* for message space \mathcal{X} consists of the algorithms (PrmsGen, KeyGen, Sign, Eval, Process, Verify) with the following syntax.

- $\text{pp} \leftarrow \text{PrmsGen}(1^\lambda, 1^N)$: Gets the security parameter λ and a data-size bound N . Generates the public parameters pp .
- $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^\lambda, \text{pp})$: Gets the security parameter along with the public parameters. Generates the public key and the secret key.

- $(\mathbf{s}_\psi, \mathbf{s}_1, \dots, \mathbf{s}_N) \leftarrow \text{Sign}_{\text{sk}}((x_1, \dots, x_N), \psi)$: Signs the data $(x_1, \dots, x_N) \in \mathcal{X}^N$ under a label $\psi \in \{0, 1\}^*$.
- $\mathbf{s}^* \leftarrow \text{Eval}_{\text{pp}}(g, \mathbf{s}_\psi, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$: Homomorphically evaluates function g on the signatures, outputting a new signature \mathbf{s}^* .
- $\alpha_g \leftarrow \text{Process}_{\text{pp}}(g)$: Computes a “public key” of g that is used in the verification.
- $b \leftarrow \text{Verify}_{\text{pk}}(\alpha_g, y, \psi, (\mathbf{s}_\psi, \mathbf{s}^*))$: Using the signatures $(\mathbf{s}_\psi, \mathbf{s}^*)$ to verify that y is the output of g evaluated at the data (x_1, \dots, x_l) under the label ψ ; outputs 1 if that is the case, 0 otherwise.

The correctness of multi-data HS is defined analogously to that of single-data HS. Also, the security notions are similar for the most part, but there are two differences. Firstly, \mathcal{A} can select arbitrarily many data sets along with a label, as long as all labels only appear once. The data sets are allowed to be of different sizes. All data sets are signed and the signatures given to \mathcal{A} . Secondly, there is a new way for \mathcal{A} to win. In the game, \mathcal{A} returns $g, \psi, y, \mathbf{s}_\psi, \mathbf{s}$ i.e. the forgery. As before, \mathcal{A} wins (that is, the game returns 1) if (i) the forgery is accepted by `Verify`, and (ii) g is admissible on the corresponding data set but g evaluated at the data set is not equal to y . This is called a type I forgery. However, for multi-data security, there is an alternative condition to (ii): \mathcal{A} wins also if ψ was not in the data set labels that \mathcal{A} chose initially, or if the size of the data set corresponding to ψ is not equal to the number of arguments of g . We call that a type II forgery.

The full definitions of the correctness and security are omitted; we refer the interested reader to look up the details from [23].

6.3.1 A selectively secure multi-data HS scheme

Theorem 5.1 of [23] states that one can transform a fully secure single-data HS scheme into a fully secure multi-data HS scheme. This does not directly apply to our vSIS HS since we could only prove selective security. However, we observe that the result can be straight-forwardly adapted to transform a selectively secure single-data scheme into a selectively secure multi-data scheme.

Theorem 6.13 (Adapted from Theorem 5.1 of [23]). *Let $\Sigma' = (\text{PrmsGen}', \text{KeyGen}', \text{Sign}', \text{Eval}', \text{Process}', \text{Verify}')$ be a single-data HS scheme, and $\Sigma^{\text{nh}} = (\text{NH.KeyGen}, \text{NH.Sign}, \text{NH.Verify})$ be a regular (not homomorphic) signature scheme. Construct a multi-data HS scheme $\Sigma = (\text{PrmsGen}, \text{KeyGen}, \text{Sign}, \text{Eval}, \text{Process}, \text{Verify})$ as follows.*

$\text{PrmsGen}(1^\lambda, 1^N)$ <hr/> $\text{pp} \leftarrow \text{PrmsGen}'(1^\lambda, 1^N)$ return pp	$\text{Eval}_{\text{pp}}(g, \mathbf{s}_\psi, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$ <hr/> $\text{parse } \mathbf{s}_\psi \text{ as } (\text{pk}_2, \psi, N, \rho)$ $\mathbf{s}^* \leftarrow \text{Eval}'_{\text{pp}}(g, (x_1, \mathbf{s}_1), \dots, (x_l, \mathbf{s}_l))$ $\text{return } \mathbf{s}^*$
$\text{KeyGen}(1^\lambda, \text{pp})$ <hr/> $(\text{pk}_1, \text{sk}_1) \leftarrow \text{NH.KeyGen}(1^\lambda, \text{pp})$ $\text{pk} := \text{pk}_1$ $\text{sk} := (\text{sk}_1, \text{pp})$ $\text{return } (\text{pk}, \text{sk})$	$\text{Process}_{\text{pp}}(g)$ <hr/> $\alpha_g \leftarrow \text{Process}'_{\text{pp}}(g)$ $\text{return } \alpha_g$
$\text{Sign}_{\text{sk}}((x_1, \dots, x_N), \psi)$ <hr/> $(\text{pk}_2, \text{sk}_2) \leftarrow \text{KeyGen}'(1^\lambda, \text{pp})$ $\rho \leftarrow \text{NH.Sign}_{\text{sk}_1}((\text{pk}_2, \psi, N))$ $\mathbf{s}_\psi := (\text{pk}_2, \psi, N, \rho)$ $(\mathbf{s}_1, \dots, \mathbf{s}_N) \leftarrow \text{Sign}'_{\text{sk}_2}(x_1, \dots, x_N)$ $\text{return } (\mathbf{s}_\psi, \mathbf{s}_1, \dots, \mathbf{s}_N)$	$\text{Verify}_{\text{pk}}(\alpha_g, y, \psi, (\mathbf{s}_\psi, \mathbf{s}^*))$ <hr/> $\text{parse } \mathbf{s}_\psi \text{ as } (\text{pk}_2, \psi, N, \rho)$ $b_0 \leftarrow \text{NH.Verify}_{\text{pk}_1}((\text{pk}_2, \psi, N), \rho)$ $b_1 \leftarrow \text{Verify}'_{\text{pk}_2}(\alpha_g, y, \mathbf{s}^*)$ $\text{return } b_0 \wedge b_1$

If Σ' is selectively secure and Σ^{nh} is secure, then Σ is selectively secure.

The rough idea of the proof is to observe that to win the game Selective-security, adversary \mathcal{A} either needs to break the security of Σ^{nh} or find a type II forgery. The latter implies an adversary \mathcal{B} that uses \mathcal{A} to break the single-data selective security. We omit the full proof since it overlaps with that of [23]; we only need to swap the full security games to selective security games, but everything else works similarly.

The following corollary is immediate.

Corollary 6.14. *There exists a vSIS-based selectively secure multi-data HS scheme.*

6.3.2 A fully secure multi-data HS scheme

[23] also presents another single-data to multi-data transformation. It is similar to the first one, apart from the fact that this time there are no public parameters, i.e., no pp generated by PrmsGen at the beginning. Instead, Sign generates fresh pp on every invocation, computes a signature $\rho \leftarrow \text{NH.Sign}_{\text{sk}_1}((\text{pp}, \text{pk}_2, \psi, N))$, and defines $\mathbf{s}_\psi := (\text{pp}, \text{pk}_2, \psi, N, \rho)$.

The motivation for this approach is that we can start from a selectively secure scheme and produce a fully secure one. A drawback is that not having one commonly agreed set of public parameters means that we no longer benefit from preprocessing a function g by Process . It can as well be done in the verification phase, and as such, we no longer have efficient verification with preprocessing.

A second drawback is that the public parameters pp of the scheme are required to be short. That is not the case with our vSIS HS scheme, but there is a workaround

borrowed from [23]: if we use a hash function, we obtain a scheme that is selectively secure in the random oracle model.

Proposition 6.15. *There exists a vSIS-based single-data HS scheme that is selectively secure in ROM.*

Proof. Let $H : \mathcal{R} \rightarrow \mathcal{R}$ be a hash function. Take the scheme of Definition 6.8 and modify it in the following way. In PrmsGen, instead of sampling r_i uniformly in \mathcal{R} , sample them from a distribution consisting of short ring elements. Then, replace r_i with $H(r_i)$ in the algorithms Sign and Process.

Now, pp consists of short elements only. Moreover, if H is modeled as a random oracle, $H(r_i)$ are uniformly random elements in \mathcal{R} . Hence, the proof of Lemma 6.11 works exactly as before. \square

The next statement follows directly from combining Proposition 6.15 with Theorem A.1 of [23].

Corollary 6.16. *There exists a vSIS-based multi-data HS scheme that is fully secure in ROM.*

References

- [1] R. E. Crandall and C. Pomerance, *Prime numbers: a computational perspective*, vol. 2. Springer, 2005.
- [2] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.
- [3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
- [4] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, *et al.*, “Strong quantum computational advantage using a superconducting quantum processor,” *Physical review letters*, vol. 127, no. 18, p. 180501, 2021.
- [5] IBM Research Communications, “IBM Unveils 400 Qubit-Plus Quantum Processor and Next-Generation IBM Quantum System Two.” <https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two>, 2022. Accessed: January 26, 2024.
- [6] Atom Computing, “Quantum startup Atom Computing first to exceed 1,000 qubits.” <https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/>, 2023. Accessed: January 26, 2024.
- [7] IBM Research Communications, “IBM Quantum System Two: the era of quantum utility is here.” <https://research.ibm.com/blog/ibm-quantum-system-two>, 2023. Accessed: January 26, 2024.
- [8] M. Ajtai, “Generating hard instances of lattice problems,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 99–108, 1996.
- [9] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.
- [10] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, *et al.*, “Status report on the third round of the NIST post-quantum cryptography standardization process,” *US Department of Commerce, NIST*, 2022.
- [11] C. Peikert *et al.*, “A decade of lattice cryptography,” *Foundations and trends® in theoretical computer science*, vol. 10, no. 4, pp. 283–424, 2016.

- [12] D. Micciancio, “Generalized compact knapsacks, cyclic lattices, and efficient one-way functions,” *computational complexity*, vol. 16, pp. 365–411, 2007.
- [13] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” in *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pp. 1–23, Springer, 2010.
- [14] A. Langlois and D. Stehlé, “Worst-case to average-case reductions for module lattices,” *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.
- [15] J. Hoffstein, J. Pipher, and J. H. Silverman, “NTRU: a new high speed public key cryptosystem.” Preprint; presented at the rump session of Crypto’96, 1996.
- [16] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte, “NTRUSIGN: Digital signatures using the NTRU lattice,” in *Cryptographers’ track at the RSA conference*, pp. 122–140, Springer, 2003.
- [17] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, “Falcon: Fast-Fourier lattice-based compact signatures over NTRU,” *Submission to the NIST’s post-quantum cryptography standardization process*, vol. 36, no. 5, 2018.
- [18] A. Pellet-Mary and D. Stehlé, “On the hardness of the NTRU problem,” in *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*, pp. 3–35, Springer, 2021.
- [19] J. Felderhoff, A. Pellet-Mary, D. Stehlé, and B. Wesolowski, “Ideal-SVP is Hard for Small-Norm Uniform Prime Ideals,” in *Theory of Cryptography Conference*, pp. 63–92, Springer, 2023.
- [20] J. Felderhoff, A. Pellet-Mary, and D. Stehlé, “On Module Unique-SVP and NTRU,” in *Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part III*, pp. 709–740, Springer, 2023.
- [21] V. Cini, R. W. F. Lai, and G. Malavolta, “Lattice-based succinct arguments from vanishing polynomials,” in *Annual International Cryptology Conference*, pp. 72–105, Springer, 2023.
- [22] L. Ducas, V. Lyubashevsky, and T. Prest, “Efficient identity-based encryption over NTRU lattices,” in *Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, ROC, December 7–11, 2014, Proceedings, Part II 20*, pp. 22–41, Springer, 2014.

- [23] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, “Leveled fully homomorphic signatures from standard lattices,” in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 469–477, 2015.
- [24] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, 2009.
- [25] C. Gentry, A. Sahai, and 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*, pp. 75–92, Springer, 2013.
- [26] S. Lang, *Algebra*, vol. 211. Springer Science & Business Media, 2012.
- [27] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*. Oxford university press, 1979.
- [28] R. W. F. Lai, “Lecture notes for the Aalto University course Advanced Topics in Cryptography.” Unpublished, 2023.
- [29] R. C. Bollinger, “Extended pascal triangles,” *Mathematics Magazine*, vol. 66, no. 2, pp. 87–94, 1993.
- [30] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.
- [31] O. Goldreich, *Foundations of Cryptography, Volume 1 (Basic Tools)*. Cambridge university press, 2001.
- [32] O. Goldreich, “Computational complexity: a conceptual perspective,” *ACM Sigact News*, vol. 39, no. 3, pp. 35–39, 2008.
- [33] C. Brzuska and V. Lipiäinen, “Companion to Cryptographic Primitives, Protocols and Proofs.” <https://github.com/cryptocompanion/cryptocompanion>, 2021. Accessed: December 12, 2023.
- [34] J. W. S. Cassels, *An Introduction to the Geometry of Numbers*. Classics in Mathematics, Springer, 1997 ed., 1996.
- [35] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. Grundlehren der mathematischen Wissenschaften 290, Springer-Verlag New York, 3 ed., 1999.
- [36] I. Stewart and D. Tall, *Algebraic number theory and Fermat’s last theorem*. AK Peters/CRC Press, 2001.

- [37] O. Regev, “Lecture notes for the Tel-Aviv University course Lattices in Computer Science.” https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/, 2004. Accessed: November 9, 2023.
- [38] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 197–206, 2008.
- [39] L. C. Washington, *Introduction to cyclotomic fields*, vol. 83. Springer Science & Business Media, 1997.
- [40] V. Lyubashevsky and D. Micciancio, “Generalized compact knapsacks are collision resistant,” in *International Colloquium on Automata, Languages, and Programming*, pp. 144–155, Springer, 2006.
- [41] V. Lyubashevsky, C. Peikert, and 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 32*, pp. 35–54, Springer, 2013.
- [42] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, “Classical hardness of learning with errors,” in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 575–584, 2013.
- [43] K. de Boer, L. Ducas, A. Pellet-Mary, and B. Wesolowski, “Random self-reducibility of ideal-SVP via Arakelov random walks,” in *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II*, pp. 243–273, Springer, 2020.
- [44] C.-P. Schnorr, “A hierarchy of polynomial time lattice basis reduction algorithms,” *Theoretical computer science*, vol. 53, no. 2-3, pp. 201–224, 1987.
- [45] D. Stehlé and R. Steinfeld, “Making NTRU as secure as worst-case problems over ideal lattices,” in *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, pp. 27–47, Springer, 2011.
- [46] T. Pornin and T. Prest, “More efficient algorithms for the NTRU key generation using the field norm,” in *Public-Key Cryptography–PKC 2019: 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography, Beijing, China, April 14-17, 2019, Proceedings, Part II*, pp. 504–533, Springer, 2019.
- [47] L. Ducas and T. Prest, “Fast fourier orthogonalization,” in *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pp. 191–198, 2016.

- [48] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, and K. Xagawa, “ModFalcon: compact signatures based on module-NTRU lattices,” in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pp. 853–866, 2020.
- [49] J. H. Cheon, D. Kim, T. Kim, and Y. Son, “A new trapdoor over module-NTRU lattice and its application to ID-based encryption,” *Cryptology ePrint Archive*, 2019.
- [50] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 9.7)*, 2022. <https://www.sagemath.org>.
- [51] L. Babai, “On Lovász’ lattice reduction and the nearest lattice point problem,” *Combinatorica*, vol. 6, pp. 1–13, 1986.
- [52] A. K. Lenstra, H. W. Lenstra, and L. Lovász, “Factoring polynomials with rational coefficients,” *Mathematische annalen*, vol. 261, no. ARTICLE, pp. 515–534, 1982.
- [53] D. Micciancio and C. Peikert, “Trapdoors for lattices: Simpler, tighter, faster, smaller,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 700–718, Springer, 2012.
- [54] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, “Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits,” in *Advances in Cryptology—EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings 33*, pp. 533–556, Springer, 2014.
- [55] C. Fieker and D. Stehlé, “Short bases of lattices over number fields,” in *International Algorithmic Number Theory Symposium*, pp. 157–173, Springer, 2010.
- [56] H. Cohen, “A Course in Computational Algebraic Number Theory,” *Graduate Texts in Mathematics*, 1993.

A Computing a somewhat short generator of an ideal

Apart from the integral basis and the two-element representation, there is a third, often useful way to represent an integral ideal; all integral ideals $I \subseteq \mathcal{R}$ can be written as the intersection of a principal fractional ideal and the ring of integers \mathcal{R} . Moreover, such a representation can be computed efficiently, as stated in Lemma 4.2 of [18]. In addition to that result, we are interested in bounding the norm of the generating element. Since the full proof is technical and mostly overlaps with the proof of Theorem 3 of [55], we only outline the proof here.

Lemma A.1 (Adapted from lemmas 2.6 and 4.2 of [18], as well as Theorem 3 of [55]). *Let K be a number field and \mathcal{R} be its ring of integers. Then, there exists a PPT algorithm (in $\text{size}(I)$ and $\log \Delta_K$) which, given a non-zero integral ideal I as input (represented by an arbitrary \mathbb{Z} -basis), computes $z \in K$ such that $\langle z \rangle \cap \mathcal{R} = I$.*

Moreover, the output satisfies

$$\left\| \sigma \left(z^{-1} \right) \right\| \leq 2^{2(d+1)} d^2 \Delta_K^{2/d} \delta_K^4 \mathcal{N}(I)^3$$

Proof (sketch). If $I = \mathcal{R}$ the algorithm returns $z = 1$, so let us consider what happens if that is not the case. Let $J = \mathcal{N}(I) \cdot I^{-1}$; this is integral by Lemma 2.63. Also notice that we can compute an integral basis for J for example using Algorithm 4.8.21 of [56].

Since $I \in \mathcal{R}$, $1 \in I^{-1}$ and $\mathcal{N}(I) \in J$. Therefore, we can input $(J, \mathcal{N}(I))$ to the algorithm of Lemma 2.6 of [18]. The algorithm returns $y \in J$ such that $J = \langle \mathcal{N}(I) \rangle + \langle y \rangle$. Moreover, the running time of the algorithm is polynomial in $\text{size}(\mathcal{N}(I))$, $\text{size}(I)$ and $\log(\Delta_K)$.

Since $J \neq \mathcal{R}$, $y \neq 0$ so let us define $z = \frac{\mathcal{N}(I)}{y}$. We have

$$I^{-1} = \frac{J}{\mathcal{N}(I)} = \frac{1}{\mathcal{N}(I)} (\langle \mathcal{N}(I) \rangle + \langle y \rangle) = \mathcal{R} + \langle z^{-1} \rangle.$$

Applying Lemma 2.62 yields $I = \mathcal{R} \cap \langle z \rangle$.

It remains to bound z^{-1} from above; for that purpose, we need to analyze the algorithm of Lemma 2.6 of [18] more carefully. Their algorithm is the same as the algorithm described in Fig. 1 of [55], except for defining x_1 to be x instead of the first element of the reduced basis. In our context, $x = \mathcal{N}(I)$. By following the algorithm step by step, one can verify that even after this change, it holds that

$$\left\| \sigma(x_2) \right\| \leq 4d^2 \gamma^4 \Delta_K^{\frac{2}{d}} \max_i \left\| \sigma(r_i) \right\|^4 \mathcal{N}(\mathfrak{a})^{\frac{4}{d}},$$

where γ is a lattice reduction parameter, $\{r_i\}_{i \in [d]}$ is a \mathbb{Z} -basis of K and $\mathfrak{a} = \langle x_1 \rangle$; we refer to [55] for details.

Using LLL as the lattice reduction algorithm, we can take $\gamma = 2^{d/2}$. Using this knowledge and translating the previous bound to our notations we get

$$\left\| \sigma(y) \right\| \leq 2^{2(d+1)} d^2 \Delta_K^{\frac{2}{d}} \delta_K^4 \mathcal{N}(\mathcal{N}(I))^{\frac{4}{d}} = 2^{2(d+1)} d^2 \Delta_K^{\frac{2}{d}} \delta_K^4 \mathcal{N}(I)^4.$$

Notice that even though $\delta_K = \max_i \|\sigma(r_i)\|_\infty$ and the former equation uses Euclidean norm, this does not pose a problem. This is because [55] uses the inequality $\max_i \|\sigma(br_i)\| \leq \max_i \|\sigma(b)\| \|\sigma(r_i)\|$ but, due to properties of the canonical embedding, we also have $\max_i \|\sigma(br_i)\| \leq \max_i \|\sigma(b)\| \|\sigma(r_i)\|_\infty$.

Hence,

$$\begin{aligned} \left\| \sigma(z^{-1}) \right\| &= \left\| \sigma\left(\frac{y}{\mathcal{N}(I)}\right) \right\| \leq \|\sigma(y)\| \left\| \sigma(\mathcal{N}(I)^{-1}) \right\|_\infty \\ &\leq \|\sigma(y)\| \mathcal{N}(I)^{-1} = 2^{2(d+1)} d^2 \Delta_K^{\frac{2}{d}} \delta_K^4 \mathcal{N}(I)^3. \end{aligned}$$

□