Hardness Assumptions in Lattice-Based Cryptography

Contents

Introduction

Lattices are mathematical objects that play an important role in many different areas such as number theory, geometry and group theory, and they have been studied for more than 250 years. The use of Euclidean lattices in cryptography started with the LLL algorithm in 1982 and its applications to cryptanalysis. In 1996, Ajtai showed that it was possible to build a hash function whose security is based on worst-case instances of hard problems on lattices. The field then took off in the late 2000s, first with Regev's work that introduced the Learning With Errors problem and showed how to build a secure public key encryption scheme from this problem, and then with Gentry's work in 2009 that was the first to build a fully homomorphic encryption scheme. Cryptographic schemes based on lattices have gone in a few years from theoretical to practical constructions, which are now among the final candidates for a transition to post-quantum cryptography.

History

- Carl Friedrich Gauß (1777-1855): use of lattices in number theory
- Hermann Minkowski (1864-1909): study the geometry of lattices
- **1982** LLL (Lenstra, Lenstra, Lovász): reduction of a lattice basis; early applications to factorization of polynomials with rational coefficients and integer linear programming problems
- **1996** Miklós Ajtai's work [Ajt96]: birth of lattice-based cryptography; introduction of the Short Integer Solution problem (SIS); first worst-case to average-case reduction and construction of a one-way function
- **1998** NTRU scheme [HPS98]: defines an average-case problem on special lattices, (at the time) no (known) connection to worst-case lattice problems, but very efficient
- **2005** Oded Regev's work [Reg05]: introduction of the Learning With Erros problem (LWE) together with a quantum worst-case to average-case reduction; public key encryption based on LWE
- **Since 2005**: a plethora of cryptographic constructions based LWE and/or SIS; see this nice website
  - Gentry Peikert Vaikuntanathan [GPV08]: "hash-then-sign" signatures, trapdoor functions, identity-based encryption;
  - Gentry [Gen09]: first fully homomorphic encryption schemes based on ideal latitces (variants based on LWE starting from 2011);
  - Lyubashevsky [Lyu09; Lyu12]: "Fiat-Shamir with Aborts" signatures.
- **Since 2009**: structured variants of SIS and LWE
  - Ring-SIS [LM06; PR06]
  - Ring-LWE [Ste+09; LPR10]
  - Module-SIS and Module-LWE [LS15]
- **Since 2016**: NIST's post-quantum standardization process
  - standardize post-quantum signatures and encryption
  - 5 out of the 7 finalist candidates are based on lattices
  - 3 of them are based on variants of LWE (Dilithium, Kyber, Saber)

Avantages

- Simple: cryptographic constructions can often be expressed with rather simple linear algebra
- Most constructions are provable secure in a strong sense: they rely on worst-case instances of well-studied lattice problems
- Presumably quantum-resistant: up to today, no significant speed-up over classical algorithms when considering quantum computers
- Very versatile: we know how to build a large variety of cryptographic primitives on lattices

Disadvantages

- When adapting known constructions from the discrete log setting to lattices, one often gets issues with the *smallness* of a solution to SIS or the noise of LWE
- Most practical constructions are still less efficient than discrete log/factoring based problems
- Discrete Gaussian distributions can be very tedious to work with

General Principle

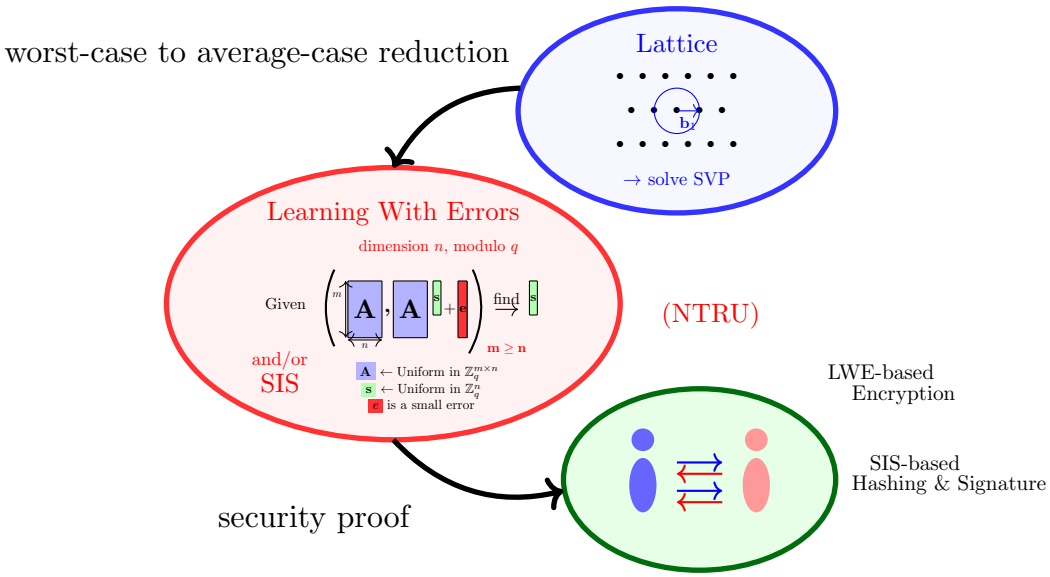


Figure 1: General principle of lattice-based cryptography

Acknowledgements

The present crash course lecture notes are a compressed and translated version of a course on lattice-based cryptography by Adeline Roux-Langlois given at the University of Rennes in 2021, from where most of the figures are taken. It further took inspiration from Peikert's survey [Pei16a] and my PhD thesis [Bou21]. Some examples are taken from the book of Hoffstein et al. [HPS08].

I further thank Xiaohui (Daisy) Ding and Marius Àrdal for pointing out erratas of previous versions of the handout.

Part I
Euclidean Lattices

In the first part of this crash course, we introduce the mathematical objects of Euclidean lattices, together with some computational problems that are relevant for cryptography.

1 Definitions

We start with some important definitions related to Euclidean lattices. First of all, *what is a lattice?*

Definition 1. *An n-dimensional Euclidean lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$.*

- Additive subgroup: $\mathbf{0} \in \Lambda$, and for all $\mathbf{x}, \mathbf{y} \in \Lambda$, $\mathbf{x}+\mathbf{y}, -\mathbf{x} \in \Lambda$;
- Discrete: exery $\mathbf{x} \in \Lambda$ has a neighborhood in which $\mathbf{x}$ is the only lattice point. That is, $\forall \mathbf{x} \in \Lambda, \exists \varepsilon > 0$ such that $\mathcal{B}(\mathbf{x}, \varepsilon) \cap \Lambda = \{\mathbf{x}\}$ (where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε around $\mathbf{x}$)[1].

Example 2. *The integer lattice $\mathbb{Z}^n$ is a lattice. We can also scale the integer lattice by any real number c, obtaining $c\mathbb{Z}^n$. Or, we can rotate the integer lattice by some orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ (that is $\boldsymbol{R}^T \cdot \boldsymbol{R} = \boldsymbol{I}_n$), obtaining $\boldsymbol{R}\mathbb{Z}^n$.*

Minima. For every lattice Λ, we define its (first) minimum as $\lambda_1(\Lambda) := \min_{\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$. We also consider its i-th minimum $\lambda_i(\Lambda)$ defined as the smallest r such that Λ contains i linearly independent vectors of norm at most r. More formally, $\lambda_i(\Lambda) := \min_{r \in \mathbb{N}} \{\dim(\mathrm{span}(\Lambda \cap \overline{\mathcal{B}}(\mathbf{0}, r))) \geq i\}$. Here $\overline{\mathcal{B}}(\mathbf{0}, r)$ denotes the closed ball of radius r around $\mathbf{0}$.

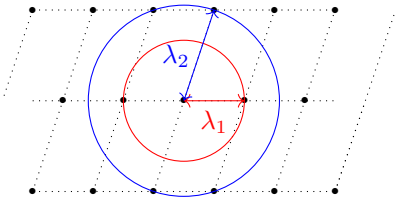


Figure 2: A lattice of $\mathbb{R}^2$ with first and second minimum

Bases. For their use in cryptography, we need an algorithmic way to describe a lattice Λ. An euclidean lattice can be defined with the help of a finite basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_k) \subset \mathbb{R}^n$, where $k \leq n$. More precisely, the lattice is given as all the integer linear combinations of the basis (column) vectors

$$\Lambda(\mathbf{B}) = \left\{ \sum_{i=1}^{k} z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\} = \{\mathbf{B}\mathbf{z} \colon \mathbf{z} \in \mathbb{Z}^n\}.$$

The *rank* of a lattice is the number k of basis vectors needed. If $k = n$, we call the lattice full-rank. We define the *span* of the lattice as the real vector space generated by its basis vectors. That is

$$\mathrm{span}(\Lambda(\mathbf{B})) := \mathrm{span}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{R}^n\}.$$

If the lattice has full rank n, then $\mathrm{span}(\mathbf{B}) = \mathbb{R}^n$.

[1] Throughout this course we implicitly mean the Euclidean ℓ_2-norm when we talk about norms and distances.

Note that a lattice basis $\mathbf{B}$ for a given lattice Λ is not unique. For any unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n\times n}$ (that is $\det(\mathbf{U}) = \pm 1$), the matrix $\mathbf{B} \cdot \mathbf{U}$ also defines a basis of Λ.[2]

Example 3. *The integer lattice $\mathbb{Z}^n$ has basis $\boldsymbol{I}_n$, the scaled lattice $c\mathbb{Z}^n$ has basis $c\boldsymbol{I}_n$ and the rotation lattice $\boldsymbol{R}\mathbb{Z}^n$ has basis $\boldsymbol{R}$.*

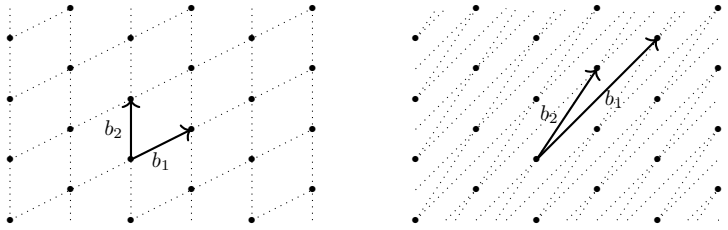


Figure 3: Two different bases for the same lattice in $\mathbb{R}^2$

Another useful mathematical notion is the (origin-centered) *fundamental parallelepiped* of a lattice Λ having basis $\mathbf{B}$. It is defined as $\mathcal{P}(\mathbf{B}) := \{\sum_{i=1}^{n} c_i \mathbf{b}_i : c_i \in [-1/2, 1/2)\}$. Note that every coset $\mathbf{x} + \Lambda$ with $\mathbf{x} \in \mathbb{R}^n$ has exactly one representative in $\mathcal{P}(\mathbf{B})$.

We further define the *volume* of a lattice as the absolute volume of the determinant of any of its basis. That is, $\det(\Lambda) := |\det(\mathbf{B})|$. It can be shown that this equals the volume of its fundamental parallelepiped.

Dual lattice. A last notion we need is the *dual* of a lattice $\Lambda \subset \mathbb{R}^n$. It is defined as

$$\Lambda^\vee := \{\mathbf{w} \in \mathrm{span}(\Lambda) \colon \langle \mathbf{w}, \mathbf{x} \rangle \in \mathbb{Z} \forall \mathbf{x} \in \Lambda\}.$$

If $\mathbf{B}$ is a basis for Λ, then is $(\mathbf{B}^T)^{-1}$ a basis for $\Lambda^\vee$. This directly implies that $\det(\Lambda^\vee) = \det(\Lambda)^{-1}$.

Example 4. *One can see that $\Lambda = \mathbb{Z}^n$ and their rotations $\boldsymbol{R}\mathbb{Z}^n$ are self-dual (that is $\Lambda^\vee = \Lambda$) and that $(c\mathbb{Z}^n)^\vee = 1/c \cdot \mathbb{Z}^n$.*

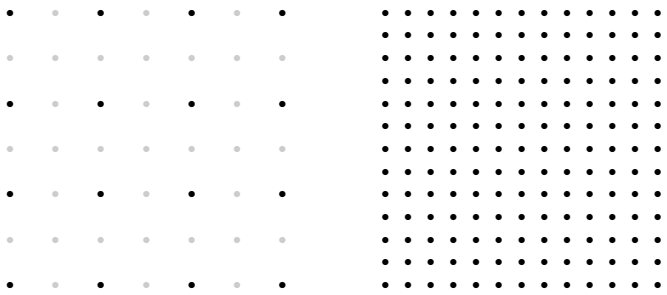


Figure 4: $2\mathbb{Z}^2$ and its dual $\frac{1}{2}\mathbb{Z}^2$

Minkowski. We defined the first minimum of a lattice. But how small is it for a given lattice? Minkowski provided an upper bound on the norm of a shortest non-zero vector in arbitrary lattices. More concretely, for a lattice Λ of dimension n and determinant $\det(\Lambda)$, it yields

$$\lambda_1(\Lambda) \leq \sqrt{n} \det(\Lambda)^{1/n}.$$

[2] This is easy to see, using $\mathbf{U} \cdot \mathbb{Z}^n = \mathbb{Z}^n$.

Example 5. *For the integer lattice, any unit vector is a shortest vector and thus* $\lambda_1(\mathbb{Z}^n) = 1$.

Using Minkowski's bound, we can show

Lemma 6. *Let* Λ *be an* n*-dimensional lattice. It yields*

- $\lambda_1(\Lambda) \cdot \lambda_1(\Lambda^\vee) \leq n$,
- $\lambda_1(\Lambda) \cdot \lambda_n(\Lambda^\vee) \geq 1$.

Banaszczyk showed a even stronger relation between the lattice's and its dual's minima, known as the transference theorem.

Theorem 7 ([Ban93, Lemma 1.5])**.** *Let* Λ *be an* n*-dimensional lattice. It yields*

$$1 \leq \lambda_1(\Lambda) \cdot \lambda_n(\Lambda^\vee) \leq n.$$

Heuristically, one can estimate the expected norm of a shortest non-zero vector in *randomly chosen* lattices by using the Gaussian heuristic, which slightly improves the Minkowski bound. It says that for an n-dimensional lattice Λ with determinant $\det(\Lambda)$, we expect

$$\lambda_1(\Lambda) \approx \sqrt{\frac{n}{2\pi e}} \cdot \det(\Lambda)^{1/n}.$$

q-ary lattices. There is a special class of lattices which plays an important role in lattice-based cryptography. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ for some integers $n, m, q \in \mathbb{N}$, we can define two lattices

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \colon \mathbf{y} = \mathbf{A}\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\} \quad \text{and}$$
$$\Lambda_q^\perp(\mathbf{A}^T) = \{\mathbf{y} \in \mathbb{Z}^m \colon \mathbf{A}^T\mathbf{y} = \mathbf{0} \bmod q\}.$$

Both lattices are of dimension m. The first is generated by the rows of $\mathbf{A}$ and has determinant q^{m-n}, whereas the second contains all vectors that are orthogonal to the rows of $\mathbf{A}$ and has determinant q^n. Furthermore, they are connected via lattice duality, i.e., $\Lambda_q^\perp(\mathbf{A}^T) = q \cdot \Lambda_q(\mathbf{A})^\vee$ and $\Lambda_q(\mathbf{A}) = q \cdot \Lambda_q^\perp(\mathbf{A}^T)^\vee$.

2 Computational Problems

We now define some important computational problems on Euclidean lattices. There are of course many more, and the study of their relations defines its own field of research. An overview of the existing dimension-preserving reductions among them can be found here [SD16].

2.1 Shortest Vector Problem

The Shortest Vector Problem (SVP) asks to find a shortest non-zero vector of a lattice.

Definition 8 (SVP)**.** *An input to the Shortest Vector Problem* SVP *is a basis* $\boldsymbol{B}$ *of an* n*-dimensional lattice* Λ*. The goal is to find a vector* $\boldsymbol{z} \neq \boldsymbol{0}$ *such that* $\|\boldsymbol{z}\| = \lambda_1(\Lambda)$.

As before, we implicitly assume the ℓ_2-norm, but all definitions can also be formulated with respect to other norms over $\mathbb{R}^n$. In cryptography, we use a relaxed version of this problem, which asks to find a shortest non-zero vector only up to some approximation factor γ.

Definition 9 (SVP_γ)**.** *Let* $\gamma = \gamma(n) \geq 1$ *be a function in the dimension* n*. An input to the approximate Shortest Vector Problem* SVP_γ *is a basis* $\boldsymbol{B}$ *of an* n*-dimensional lattice* Λ*. The goal is to find a vector* $\boldsymbol{z} \neq \boldsymbol{0}$ *such that* $\|\boldsymbol{z}\| \leq \gamma \cdot \lambda_1(\Lambda)$.

The larger γ, the easier the problem. For $\gamma = 1$, we recover the exact SVP problem from before.

The problem above requires to *find* a short vector, we call this the *search* variant of SVP_γ. But we only know how to build cryptographic schemes either on some (promise) *decision* variant of SVP_γ or on a more general search version of it.

The generalized search version asks to find not only one short vector of (approximate) norm $\lambda_1(\Lambda)$, but to find n linearly independent vectors of (approximate) norm at most $\lambda_n(\Lambda)$.

Definition 10 (SIVP_γ)**.** *Let $\gamma = \gamma(n) \geq 1$ be a function in the dimension n. An input to the approximate Shortest Independent Vector Problem SIVP_γ is a basis $\boldsymbol{B}$ of an n-dimensional lattice Λ. The goal is to find n linearly independent vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ such that $\|\boldsymbol{z}_i\| \leq \gamma \cdot \lambda_n(\Lambda)$ for all i.*

The next problem asks to distinguish between two cases (where we have the promise that the input lattice is in one of the two cases).

Definition 11 (GapSVP_γ)**.** *Let $\gamma = \gamma(n) \geq 1$ be a function in the dimension n. An input to the decision Shortest Vector Problem GapSVP_γ is a pair $(\boldsymbol{B}, r)$, where $\boldsymbol{B}$ is a basis of an n-dimensional lattice Λ and $r > 0$ is a real number. It is a YES instance if $\lambda_1(\Lambda) \leq r$, and it is a NO instance if $\lambda_1(\Lambda) > \gamma \cdot r$. The problem asks to distinguish whether a given instance is a YES or a NO instance.*

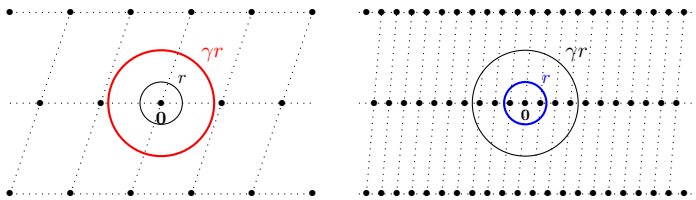


Figure 5: Examples of a NO and a YES instance of GapSVP_γ

2.2 Closest Vector Problem

The Closest Vector Problem (CVP) is another fundamental problem on Euclidean lattices. Given a point in the span of the lattice, it asks to find a lattice vector closest to it. This time, we directly define its approximate version.

Definition 12 (CVP_γ)**.** *Let $\boldsymbol{B}$ be a basis of an n-dimensional lattice $\Lambda(\boldsymbol{B})$ and $\gamma = \gamma(n) \geq 1$ be a function in the dimension n. An input to the Closest Vector Problem CVP_γ is a point $\boldsymbol{t} \in \mathrm{span}(\Lambda)$. The problem asks to find $\boldsymbol{x} \in \Lambda$ such that $\|\boldsymbol{t} - \boldsymbol{x}\| \leq \gamma \cdot \min_{\boldsymbol{y} \in \Lambda} \|\boldsymbol{t} - \boldsymbol{y}\|$.*

We can also define a decision version of CVP.

Definition 13 (GapCVP_γ)**.** *Let $\gamma = \gamma(n) \geq 1$ be a function in the dimension n. An input to the decision Closest Vector Problem GapCVP_γ is a triple $(\boldsymbol{B}, \boldsymbol{t}, r)$, where $\boldsymbol{B}$ is a basis of an n-dimensional lattice Λ, $\boldsymbol{t} \in \mathrm{span}(\Lambda)$ and $r > 0$ is a real number. It is a YES instance if $\mathsf{dist}(\boldsymbol{t}, \Lambda) \leq r$, and it is a NO instance if $\mathsf{dist}(\boldsymbol{t}, \Lambda) > \gamma \cdot r$. The problem asks to distinguish whether a given instance is a YES or a NO instance.*

For cryptography, we consider another promise version of CVP_γ.[3] That is, we are given a target point that is promised to be somewhat close to the lattice. The problem now asks to find the a lattice point closest to the target.

Definition 14 (BDD_δ)**.** *Let $\boldsymbol{B}$ be a basis of an n-dimensional lattice $\Lambda(\boldsymbol{B})$ and δ be a positive real. An input to the Bounded Distance Decoding problem BDD_δ is a point $\boldsymbol{t} \in \mathbb{R}^n$ of the form $\boldsymbol{t} = \boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{x} \in \Lambda(\boldsymbol{B})$ and $\|\boldsymbol{e}\| \leq \delta$. The problem asks to find $\boldsymbol{x}$ (or equivalently $\boldsymbol{e}$).*

For simplicity, we stated BDD with respect to the Euclidean norm. We remark, however, that it is often phrased with respect to the infinity norm in other works.

[3] Up to today, no cryptosystem has been proven secure directly based on CVP_γ.

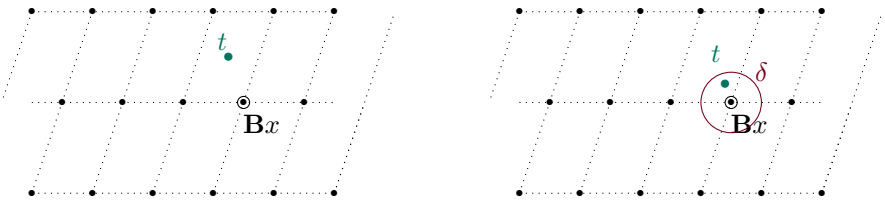


Figure 6: Examples of CVP and BDD_δ in dimension 2

Somehow, SVP looks like CVP with the target point (close to) $\mathbf{0}$. But CVP does allow the trivial solution $\mathbf{0}$, whereas SVP does not. Nonetheless, it is possible to give a reduction from SVP_γ to CVP_γ as we will see later. There is also a reduction in the other direction, but increasing the approximation factor from γ to $\sqrt{n}\gamma^2$ [SD16].

2.3 Easy Computational Problems

As we will see shortly, both SVP and CVP are difficult computational problems. However, there are also some problems on lattices that are easy to compute. For instance the following two problems.

Definition 15 (Membership)**.** *Given a basis $\boldsymbol{B}$ of an n-dimensional lattice $\Lambda(\boldsymbol{B})$ and a vector $\boldsymbol{v} \in \mathbb{R}^n$, decide if $\boldsymbol{v} \in \Lambda(\boldsymbol{B})$.*

Definition 16 (Equivalence)**.** *Given two bases $\boldsymbol{B}$ and $\boldsymbol{B}' \in \mathbb{R}^{n\times n}$. Decide if $\Lambda(\boldsymbol{B}) = \Lambda(\boldsymbol{B}')$.*

2.4 Reductions

There are many reductions between the problems we just defined, and proving all of them is definitively beyond this crash course. We refer the interested reader to Stephen-Davidowitz's overview of dimension-preserving reductions between lattice problems [SD16] for the relevant references.

For the solemn reason of mathematical curiosity, we now prove that GapSVP_γ reduces to GapCVP_γ, as shown by Goldwasser et al. [Gol+99].

Theorem 17. *There is a polynomial-time reduction from GapSVP_γ to GapCVP_γ for any input lattice $\boldsymbol{B}$ and any approximation factor γ.*

A naive approach to reduce the shortest vector problem to the closest vector problem would be, given an input basis $\mathbf{B}$ to the SVP problem, to forward the instance $(\mathbf{B}, \mathbf{0})$ to the CVP oracle. However, the oracle might simply return $\mathbf{0}$, which does not help to find a shortest vector. The strategy of the reduction is thus to take a target $\mathbf{w}$ that is not the zero vector and input a modified basis (defining a different lattice) that does not contain the target vector.

Proof. For every $j \in [n]$, we define the basis $\mathbf{B}^{(j)} := [\mathbf{b}_1, \dots, \mathbf{b}_{j-1}, 2\mathbf{b}_j, \mathbf{b}_{j+1}, \dots, \mathbf{b}_n]$. Note that this basis does not contain $\mathbf{b}_j$.

Claim 1. Let $\mathbf{v} = \sum_i c_i \mathbf{b}_i$ be a shortest vector. Then, there exists an index j such that $c_j = 1 \bmod 2$.

Claim 2. Let $\mathbf{v} = \sum_i c_i \mathbf{b}_i$ be a vector in $\Lambda(\mathbf{B})$ such that $c_j = 1 \bmod 2$ for some index j. Then, $\mathbf{u} := \frac{c_j+1}{2}(2\mathbf{b}_j) + \sum_{i\neq j} c_i \mathbf{b}_i \in \Lambda(\mathbf{B}^{(j)}$ and $\|\mathbf{u} - \mathbf{b}_j\| = \|\mathbf{v}\|$.

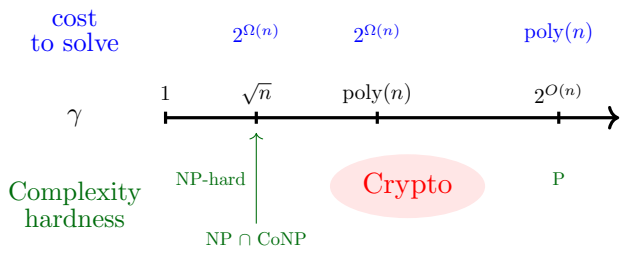


Figure 7: Schematic difficulty of GapSVP$_\gamma$

Claim 3. Let $\mathbf{u} = c'_j \cdot 2\mathbf{b}_j + \sum_{i \neq j} c_i \mathbf{b}_i \in \Lambda(\mathbf{B}^{(j)})$. Then, $\mathbf{v} := (2c'_j - 1)\mathbf{b}_j + \sum_{i \neq j} c_i \mathbf{b}_j$ is non-zero and lies in $\Lambda(\mathbf{B})$ and it yields $\|\mathbf{v}\| = \|\mathbf{u} - \mathbf{b}_j\|$.

The reduction now does the following: Given $(\mathbf{B}, r)$ as instance to GapSVP$_\gamma$, the reduction defines n instances to GapCVP$_\gamma$ by $(\mathbf{B}^{(j)}, \mathbf{b}_j, r)$ for $j = 1, \dots, n$. We will now show that if $(\mathbf{B}, r)$ is a YES instance, then there exists an index j such that $(\mathbf{B}^{(j)}, \mathbf{b}_j, r)$ is a YES instance. If $(\mathbf{B}, r)$ is a NO instance, then is $(\mathbf{B}^{(j)}, \mathbf{b}_j, r)$ a NO instance for every index j. Equivalently, we show that $(\mathbf{B}^{(j)}, \mathbf{b}_j, r)$ is not a NO instance, so there exists an index j such that $(\mathbf{B}, r)$ is not a NO instance.

□

Further, SVP$_\gamma$ is no easier than SIVP$_\gamma$ [SD16] which is itself no easier than GapSVP$_\gamma$ [Ban93].[4]

2.5 Complexity and Algorithms

Many lattice problems, and in particular the SVP and CVP problems defined above, are shown to be NP-hard for small (i.e., constant) approximation factors, e.g., [Ajt98]. However, for cryptography we need $\gamma = \mathsf{poly}(n)$.

In 1982, Lenstra, Lenstra and Lovász [LLL82] designed the now very popular LLL algorithm, that solves in polynomial time SVP$_\gamma$ for γ exponentially large in the lattice dimension. Later in 1987, Schnorr showed a trade-off between running time and approximation factor which can be achieved by an algorithm solving SVP$_\gamma$ [Sch87]. In practice, it is implemented by the BKZ algorithm by Schnorr and Euchner [SE94], which can be seen as a heuristic variant of Schnorr's algorithm. Following this trade-off, the best known algorithm to solve SVP$_\gamma$ with γ polynomial in the lattice dimension n has an exponential running time of $2^{\tilde{O}(n)}$ and, conversely, the best known algorithm to solve SVP$_\gamma$ with polynomial running-time can only achieve an exponential approximation factor γ of $2^{\tilde{O}(n)}$. Here, the term $\tilde{O}(n)$ designs the big O notation which hides logarithmic factors in n.

The above leads to the following conjecture which forms the starting point of lattice-based cryptography. Note that all asymptotic statements are with respect to the lattice dimension n, unless we state it otherwise.

Conjecture 18. *There is no polynomial-time classical or quantum algorithm that approximates the lattice problems* SVP$_\gamma$, GapSVP$_\gamma$ *or* SIVP$_\gamma$ *to within polynomial factors γ (for all possible input lattices).*

[4] Note that the approximation factors are not preserved through the reductions. In the reduction from GapSVP$_\gamma$ to SIVP$_\gamma$ it is mapped from γ to $n\gamma$, where n is the lattice dimension. And in the reduction from SIVP$_\gamma$ to SVP$_\gamma$ it is mapped from γ to $\sqrt{n}\gamma$.

3 Crypto Dilemma

> The way lattices can by used in cryptography is by no means obvious.
>
> *D. Micciancio & O. Regev*

Despite their assumed quantum-resistance, the computational lattice problems we have seen before seem unlikely to directly serve for the construction of (provably secure) cryptographic primitives. This is because their definition relies on arbitrary lattices, what we commonly call *worst-case* problems, as they are in general not hard to solve for any lattice, but assumed to be hard to solve in the worst-case. When designing cryptographic schemes, however, we usually need the hardness of random instances of some problem, what we call *average-case* problems. This challenge was solved with the help of intermediate lattice problems, namely the Short Integer Solution (SIS) [Ajt96] and Learning With Errors (LWE) [Reg05] problems, which are formulated as average-case problems, making them suitable for cryptography. Astonishingly, these intermediate lattice problems have been shown to be at least as hard to solve as some worst-case lattice problems, such as GapSVP_γ or SIVP_γ, for suitable parameter choices. Thus, Conjecture 18 which states that SIVP_γ and GapSVP_γ are classically and quantumly intractable implies that SIS and LWE are also classically and quantumly intractable. More details in the next lecture!

There have been trials to build public key encryption directly on worst-case problems, but none of them could be proven secure and at the same time withstood cryptanalysis for practical parameter settings. To name one concrete cryptosystem, Goldreich et al. [GGH97] introduced the GGH cryptosystem, which can be seen as a lattice analogue of the McEliece cryptosystem. From a high level, the secret key is given by a "good" (i.e., short) basis, where the public key is given by a "bad" (e.g., large) basis. For encryption, the message is encoded in a short vector, which is than added to a vector of the lattice (with respect to the bad basis). Knowing the good basis helps to recover the corresponding lattice point and hence the "shift" vector which encodes the message.

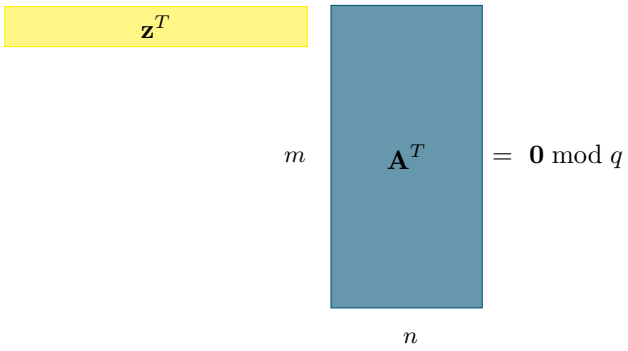


Figure 8: The Short Integer Solution (SIS) problem.

Part II
Average-Case Lattice Problems

In this part of the course, we introduce two problems on a specific class of lattices, that are very important for lattice-based cryptography.

4 Short Integer Solution

The Short Integer Solution problem, abbreviated by SIS, was introduced by Ajtai [Ajt96].

4.1 Definitions

Definition 19 (SIS)**.** *Let n, m and q be positive integers and β be a positive real. Given m independent vectors $\boldsymbol{a}_j$ sampled uniformly at random over $\mathbb{Z}_q^n$, forming the columns of a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, the problem $\mathrm{SIS}_{n,q,\beta,m}$ asks to find a nonzero vector $\boldsymbol{z} \in \mathbb{Z}^m$ of norm $\|\boldsymbol{z}\| \leq \beta$ such that*

$$\boldsymbol{A}\boldsymbol{z} = \sum z_i \boldsymbol{a}_i = \boldsymbol{0} \bmod q.$$

On the parameters. Clearly, the problem becomes harder if for fixed m, n and q we decrease the norm bound β. However, the norm bound β and the number m of vectors must be taken large enough to guarantee a solution. This is the case when we take $m \geq n \log(q)$ and $\beta \geq \sqrt{m}$.[5] Then, there are $2^m \geq q^n$ vectors $\mathbf{x} \in \{0,1\}^m$, so there must be two distinct $\mathbf{x} \neq \mathbf{x}' \in \{0,1\}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}'$ and hence $\mathbf{z} = \mathbf{x} - \mathbf{x} \in \{-1, 0, 1\}^m$ is a solution of norm at most β. In particular, parameters are mostly taken such that there are multiple solution to it. We say that SIS is a surjective problem. Furthermore, the SIS problem becomes easier if we increase the number m (we can simply ignore some columns) and harder if we increase the dimension n.

Hidden Lattice Problem. You may now wonder why this part is called *average-case lattice problems*. Where is the lattice? We can interpret SIS as a problem over *random* q-ary lattices. More precisely, SIS defines an instance of SVP_γ (Def. 9) in the random lattice

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \colon \mathbf{A}\mathbf{y} = \mathbf{0} \bmod q\},$$

where the approximation factor γ depends on the norm β.

[5] All logarithms are with respect to the base 2, unless made explicit.

Inhomogeneous version. To make things more confusing, sometimes the name SIS is used for a slightly different problem: instead of asking to find a short nonzero vector $\mathbf{z}$ that lies in the kernel of $\mathbf{A}$ modulo q, the *inhomogeneous* SIS problem (that we abbreviate in the following ISIS) asks for a given (random) target vector $\mathbf{t} \in \mathbb{Z}_q^n$ (and a uniform matrix $\mathbf{A}$) to find a short nonzero vector $\mathbf{z}$ such that $\mathbf{Az} = \mathbf{t} \bmod q$. In other words, SIS is the special case of ISIS, where $\mathbf{t} = \mathbf{0}$. But from a lattice perspective, SIS is much more convenient to work with. Another problem consists in sampling some short vector $\mathbf{z}$, then computing $\mathbf{t} = \mathbf{Az} \bmod q$ and asking to recover the unique $\mathbf{z}$ (or distinguish $\mathbf{t}$ from a random vector). Sometimes this problem is also called ISIS, but personally, I prefer the name Knapsack problem to avoid confusion. It is quite a different problem, for instance it is injective, not surjective.

Hermite Normal Form. When reading recent works on (practical) cryptographic primitives based on SIS, the problem is often presented in a slightly different way. Without loss of generality, we can assume that the n leftmost columns of the public matrix form an invertible matrix. That is, $\mathbf{A} = [\mathbf{A}_1 | \mathbf{A}_2]$ with $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times n}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{n \times (m-n)}$. In this case, we can equivalently work with the matrix $\mathbf{A}_1^{-1} \cdot \mathbf{A} = [\mathbf{I}_n | \overline{\mathbf{A}}]$, where $\overline{\mathbf{A}} = \mathbf{A}_1^{-1}\mathbf{A}_2$. So, it suffices to sample $\overline{\mathbf{A}}$ uniformly at random over $\mathbb{Z}_q^{n \times m-n}$ and treat $\mathbf{I}_n$ implicitly.

4.2 Hardness

As we have seen, SIS defines an instance of SVP over a (1) random (2) q-ary lattice. Both properties together make the lattice rather special. Whereas most experts on lattices agree that there exist hard instances of SVP, the situation is not intuitively clear for such special lattices. Fortunately, and maybe also a bit surprisingly, Ajtai [Ajt96] (and subsequent works [MR07; GPV08; MP13]) showed that SIS (under specific parameter choices) is at least as hard as solving approximate SIVP and approximate GapSVP on *any* lattice. Such reductions are often referred to as worst-case to average-case reductions.

Theorem 20. *For any $m = \mathsf{poly}(n)$, any $\beta > 0$ and any $q \geq \beta \cdot \mathsf{poly}(n)$, solving $\mathrm{SIS}_{n,q,\beta,m}$ with non-negligible probability is at least as hard as solving the problem GapSVP_γ and the problem SIVP_γ on arbitrary n-dimensional lattices with overwhelming probability, for some $\gamma = \beta \cdot \mathsf{poly}(n)$.*

For instance, [GPV08, Theorem 9.2] requires $q \geq \beta \cdot \sqrt{n} \cdot \omega(\log n)$ and $\gamma \geq \beta \cdot \sqrt{n} \cdot \omega(\sqrt{\log n})$, where $\omega(f(n))$ denotes a function that grows asymptotically faster than $f(n)$. Often, the notation $\gamma = \widetilde{O}(\sqrt{n}) \cdot \beta$ is used, where the $\widetilde{O}$ notation hides logarithmic factors. This result also applies to ISIS with a randomly chosen target $\mathbf{t}$. We skip the proof (and even its idea) here, but refer to [Pei16a, Sec. 4.1] for an accessible high level sketch.

4.3 Interlude: Ajtai's Hash Function

Even though this PhD course mainly focuses on the hardness assumptions made in lattice-based cryptography, we can do a very short detour and see how SIS serves as a hardness assumption to build collision-resistant hash functions.

Consider the hash function $f_{\mathbf{A}} \colon \{0,1\}^m \to \mathbb{Z}_q^n$ which is described by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ which is sampled uniformly at random. The function is defined as $f_{\mathbf{A}}(\mathbf{z}) = \mathbf{Az} \bmod q$. Note that $|\{0,1\}^m| = 2^m$ and $|\mathbb{Z}_q^n| = q^n$. Thus, we require $m > n \log q$ to ensure that the function is a compression. Typically, one chooses $m \approx 2n \log q$ to obtain a compression factor of 2. Note that any collision $\mathbf{z} \neq \mathbf{z}'$ immediately provides a solution to SIS with $\|\mathbf{z} - \mathbf{z}'\| \leq \sqrt{m} =: \beta$. The reasoning easily generalizes to the domain $\{0, \dots, d-1\}^m$ for any positive integer $d \geq 2$.

On the positive aspects, we can see that this hash function is very simple and involves only addition and multiplication modulo q. On the negative side, as it is presented here, it is rather inefficient. For concreteness, take $q = n^2$ and $m = 2n \log q = 4n \log n$. The description of the hash function $f_{\mathbf{A}}$ is of size $nm \log q = 8n^2 (\log n)^2$, similarly the computation of $\mathbf{Az}$. Efficient constructions rely on *structured* lattices, as we will see later in this course. In practice, one uses the SWIFFT hash function introduced by Lyubashevsky et al. [Lyu+08].

Leftover Hash Lemma. Quite useful is the following observation: If we sample $\mathbf{A}$ uniformly at random over $\mathbb{Z}_q^{n\times m}$ as well as $\mathbf{z}$ uniformly at random over $\{0,1\}^m$, we can use the leftover hash lemma to argue that $\mathbf{Az} \bmod q$ is statistically close to a uniform random vector over $\mathbb{Z}_q^n$ as long as m is large enough (compared to n and q). In other words, the decision variant of the Knapsack problem mentioned earlier becomes vacuously hard for large m. More precisely:

Lemma 21 (LHL)**.** *For some positive integers m, n and some prime q, the family of hash functions $\mathcal{H} = \{f_{\boldsymbol{A}} \colon \{0,1\}^m \to \mathbb{Z}_q^n\}_{\boldsymbol{A}\in\mathbb{Z}_q^{n\times m}}$ is universal. If furthermore, $m \geq n\log q - 2 + 2\log(1/\varepsilon)$ for some positive real ε, then it yields*

$$\Delta((\boldsymbol{A}, \boldsymbol{Az}), (\boldsymbol{A}, \boldsymbol{u})) \leq \varepsilon,$$

where $\boldsymbol{A} \leftarrow U(\mathbb{Z}_q^{n\times m})$, $\boldsymbol{z} \leftarrow U(\{0,1\}^m)$, and $\boldsymbol{u} \leftarrow U(\mathbb{Z}_q^n)$.

Here Δ denotes the statistical distance.

Proof. Recall that the family of hash functions $\mathcal{H}$ is universal if $\mathbb{P}_{\mathbf{A}\leftarrow U(\mathbb{Z}_q^{n\times m})}[f_{\mathbf{A}}(\mathbf{z}_1) = f_{\mathbf{A}}(\mathbf{z}_2)] = 1/|\mathbb{Z}_q^n| = q^{-n}$ for any $\mathbf{z}_1 \neq \mathbf{z}_2 \in \{0,1\}^m$. Let $\mathbf{z}_1 \neq \mathbf{z}_2$ such that $\mathbf{Az}_1 = \mathbf{Az}_2$. Let us start by focusing on the first row of $\mathbf{A}$. As $\mathbf{z}_1 = (\mathbf{z}_{1j})_j \neq \mathbf{z}_2 = (\mathbf{z}_{2j})_j$, there is at least one coefficient $k \in \{1,\dots,m\}$ such that $\mathbf{z}_{1k} \neq \mathbf{z}_{2k}$. Without loss of generality, we assume $k = 1$. For the first row of $\mathbf{A} = (a_{ij})_{ij}$, it yields $\sum_{j=1}^m a_{1j}\cdot(\mathbf{z}_{1j} - \mathbf{z}_{2j}) = 0 \bmod q$. This is equivalent to $a_{11} = (\mathbf{z}_{21} - \mathbf{z}_{11})^{-1}\sum_{j=2}^m a_j\cdot(\mathbf{z}_{1j} - \mathbf{z}_{2j})$. Here we used that q is prime and thus $\mathbb{Z}_q$ is a field and hence every nonzero element is invertible. In other words, the first coefficient of the first row of $\mathbf{A}$ is uniquely defined by the rest of the coefficients of the first row (and the fixed vector $\mathbf{z}_1$ and $\mathbf{z}_2$). Thus, this event happens with probability $1/q$. As every row of $\mathbf{A}$ is sampled independently from the others, we obtain by the union bound the universal property of the hash function family.

Now we simply use the general LHL as stated for instance in Lemma 2.1 in [Dod+08]. By using that the min-entropy of $U(\{0,1\}^m)$ is m and the size of the set $\mathbb{Z}_q^n$ is q^n we get the requirement $m \geq n\log q - 2 + 2\log(1/\varepsilon)$. □

For example, if $\varepsilon = 2^{-n}$, the condition $m \geq 3n\log q \geq n\log q - 2 + 2n$ is sufficient.

5 Learning With Errors

The Learning With Errors problem, abbreviated by LWE, was introduced by Regev [Reg05; Reg09].

5.1 Definitions

We start by defining the LWE distribution, which provides noisy linear equations.

Definition 22 (LWE distribution)**.** *Let n and q be positive integers and let χ be a distribution over $\mathbb{Z}$. For a fixed secret $\boldsymbol{s} \in \mathbb{Z}_q^n$, the* LWE *distribution $A_{\boldsymbol{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is obtained by choosing $\boldsymbol{a} \leftarrow U(\mathbb{Z}_q^n)$, $e \leftarrow \chi$ and outputting $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e \bmod q)$.*

We call $\mathbf{s}$ the *secret* and e the *noise* or *error* of the distribution. The LWE problem comes in two different variants. The first is a search problem and asks to find the secret $\mathbf{s}$ (or equivalently $\mathbf{e}$) and the decision problem asks to distinguish between the LWE distribution and the uniform distribution.

Definition 23 (Search LWE)**.** *Let m be a positive integer. Given m independent samples $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from $A_{\boldsymbol{s},\chi}$ for a uniformly random $\boldsymbol{s} \in \mathbb{Z}_q^n$. The problem search-*$\mathrm{LWE}_{n,q,\chi,m}$ *asks to find $\boldsymbol{s}$.*

We usually write $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ as the matrix whose rows are given by the LWE samples $\mathbf{a}_i$. Without the noise (i.e., $\chi = \{0\}$), the problem becomes easy, as we can simply solve the system of linear equations with respect to the secret $\mathbf{s}$ and recover it by linear algebra. It is thus the noise that makes the problem hard!

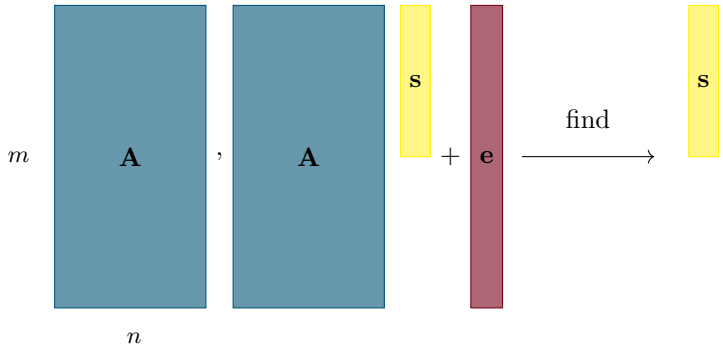


Figure 9: The LWE problem in its search variant. The number of rows m of $\mathbf{A}$ can be seen as the number of LWE samples and the number of columns n of $\mathbf{A}$ defines the dimension of the LWE problem.

Definition 24 (Decision LWE)**.** *Let m be a positive integer. Given m independent samples $(\boldsymbol{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ that are either drawn from $A_{\boldsymbol{s},\chi}$ for a uniformly random $\boldsymbol{s} \in \mathbb{Z}_q^n$ or drawn from the uniform distribution $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. The problem dec-$\mathrm{LWE}_{n,q,\chi,m}$ asks to distinguish both cases with non-negligible advantage.*

On the parameters. Let B denote the bound on the norm of noise vectors sampled from χ. Then one can see that the LWE problem gets harder if the ratio B/q or the dimension n increases. The parameter m usually doesn't has a large impact on the security, as long as it is not too big.[6] Whereas SIS is mostly used in the setting where multiple solutions exist, LWE parameters are usually chosen such that the solution is unique. We say that LWE is injective.

Noise distribution. One can see that the choice of the noise distribution χ has a direct impact on the hardness of the problem. If we would set it to the uniform distribution $\chi = U(\mathbb{Z}_q)$, then $\mathbf{e}$ completely hides the hidden $\mathbf{s}$ and thus it is vacuously hard to distinguish LWE instances from uniform ones. On the other extreme, if the noise distribution is too small, say a Bernoulli distribution $\chi = \mathrm{Ber}(p)$ with a very small probability p of sampling 1, then the problem becomes easy to solve. In this case $\mathbf{e}$ is a very sparse binary polynomial and one can simply hope to have enough equations without noise. So what are typical choices for χ? For cryptography to work, we need χ to provide error vectors of *small* norms but with enough entropy. In theoretical results, χ is often a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},\sigma}$.[7] In practical constructions, χ is often the uniform distribution over a set $\{-\beta, \dots, +\beta\}$ for some small integer β (say $\beta = 3$) or the binomial distribution of small parameters (say 3).

Secret distribution. To add even more variation, one can also consider different distributions for the *secret*. In the "standard" version, the secret is sampled uniformly at random over $\mathbb{Z}_q^n$. As for the noise, we can define variants where the secret follows a Gaussian distribution (cf. Hermite normal form), or the uniform distribution over a small set. Small secrets and noises not only improve efficiency of cryptographic schemes, they are also important in the setting of fully homomorphic encryption. The performance of such schemes crucially depends on the size of the secret.

[6] Agora and Ge [AG11] showed that if one has roughly n^{2B+1} samples, then one can solve LWE in time roughly n^{2B}.

[7] More on discrete Gaussians in Section 5.2.

Hidden Lattice Problem. As for SIS, we can interpret LWE as a problem over *random* q-ary lattices. More precisely, LWE defines an instance of BDD (Def. 14) in the random lattice

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \colon \mathbf{y} = \mathbf{A}\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\},$$

where $\mathbf{b}$ is the target point whose distance to the lattice is given by the noise $\mathbf{e} \leftarrow \chi^m$.

(Re-)randomization. Sometimes, it can be convenient to (re-)randomize the secret. In particular, this is used when going from LWE with a fixed secret $\mathbf{s}$ (called worst-case in [Reg05]) to LWE with a random secret $\mathbf{s}'$ (called average-case). To do so, we transform a sample from $A_{\mathbf{s},\chi}$ to $A_{\mathbf{s}+\mathbf{t},\chi}$, by using the linearity of matrix-vector multiplications. Given an instance $(\mathbf{A}, \mathbf{b})$, we compute $(\mathbf{A}, \mathbf{b} + \mathbf{A}\mathbf{t}) = (\mathbf{A}, \mathbf{A}(\mathbf{s} + \mathbf{t}) + \mathbf{e})$. We can also re-randomize the noise by adding $\mathbf{e}'$ to $\mathbf{b}$, that is $(\mathbf{A}, \mathbf{b} + \mathbf{e}') = (\mathbf{A}, \mathbf{A}\mathbf{s} + (\mathbf{e} + \mathbf{e}'))$. Note, however, that this increases the amount of noise.

Search-to-decision. As often the case, regarding their use in cryptography, the decision problem is much more convenient than the search problem. Luckily, both problems are equivalent (up to some polynomial loss in the advantage), as we see in the following.

Lemma 25 ([Reg05, Lem. 4.2])**.** *The problems search-*LWE *and dec-*LWE *are computationally equivalent.*

Note that the original proof was restricted to *prime* moduli that are polynomial in n, but subsequent works generalized it to essentially any modulus, see references in [Pei16a, Sec. 4.2.2].

Hermite Normal Form. As for SIS, when reading recent works on cryptographic constructions based on LWE, the LWE problems are often presented in a slightly different way. Instead of sampling $\mathbf{s}$ uniformly at random over $\mathbb{Z}_q^n$, it is drawn from the same distribution as the noise e, i.e., $\mathbf{s} \leftarrow \chi^n$. We then say that LWE is in its *Hermite normal form* and write HNF-LWE. Applebaum et al. [App+09] gave a very simple proof why HNF-LWE and LWE are equivalent, for both the search and the decision versions.

Lemma 26. *The problems* HNF$-$LWE *and* LWE *are computationally equivalent, where the modulus q and the dimension n are preserved.*

Learning Parity With Noise. When Regev introduced LWE, he actually presented it as a generalization of the classical coding problem Learning Parity With Noise (LPN). When specifying $q = 2$ and χ a Bernoulli distribution over $\{0, 1\}$, LWE coincides with LPN. However, both problems seem to behave quite differently. Whereas LWE is a geometric problem, that is connected to lattices and where we require the noise to have small Euclidean norm (or more generally a small ℓ_p-norm), LPN is a decoding problem, where the distance between code words is measured via the Hamming distance. It is still a great research direction to better understand the relationship between both problems.

Learning With Rounding. In 2012, Banerjee et al. [BPR12] introduced a deterministic variant of LWE, namely the Learning With Rounding (LWR) problem. One advantage of using LWR instead of LWE is that one doesn't need to sample an error from some specific distribution. In particular for (discrete) Gaussian error distributions, this task can be tedious. Before we formally present this deterministic variant of LWE, we need a method to round elements from $\mathbb{Z}_q$ to $\mathbb{Z}_p$, where $p \leq q$. The modular rounding function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ is defined as $\lfloor x \rceil_p = \left\lfloor \left(\frac{p}{q}\right) \cdot x \right\rceil \bmod p$, where $\lfloor \cdot \rceil$ is the standard rounding function, mapping every $y \in \mathbb{R}$ to its closest integer $\lfloor y \rceil \in \mathbb{Z}$. The modular rounding function extends component-wise to vectors over $\mathbb{Z}_q$ and coefficient-wise to polynomials in $\mathbb{Z}_q[x]$. One can deterministically lift any element from $\mathbb{Z}_p$ back to $\mathbb{Z}_q$ by mapping $y \in \mathbb{Z}_p$ to $\lfloor q/p \cdot y \rceil \in \mathbb{Z}_q$. This induces a rounding error whose absolute value is bounded above by q/p, i.e., $\left\lfloor q/p \cdot \lfloor y \rceil_p \right\rceil = y + e$, where $|e| \leq q/p$.

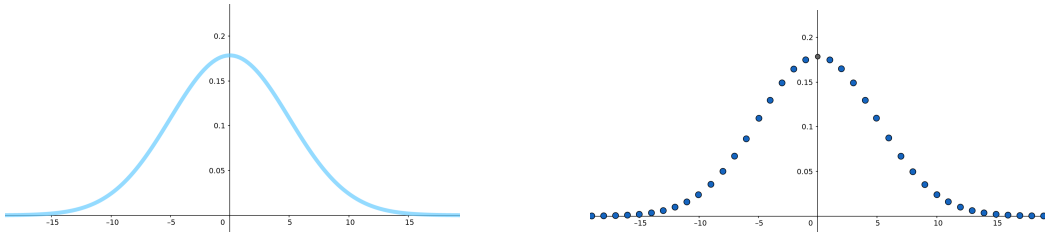


Figure 10: Graph of the probability density function of a continuous Gaussian in dimension $n = 1$ and of the probability mass function of a discrete Gaussian over the lattice $\Lambda = \mathbb{Z}$.

Now, an LWR sample is of the form $(\mathbf{a}, b = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ and we can define the search and decision LWR problems in an analogue manner to LWE. The hardness of LWR is based on the hardness of LWE. More precisely, there was a sequence of works that proved reductions (under specific parameter conditions) from LWE to LWR. There are also structured variants of LWR, that follow the same design principles as the one for LWE (cf. later lectures of this course).

5.2 Discrete Gaussian Distributions

Theoretical works on LWE, in particular those that show its worst-case hardness, use a Gaussian noise distribution. Furthermore, all lattice trapdoor functions rely on Gaussian distributions as well. More precisely, we often need *discrete* Gaussian distributions, which are much more trickier to work with than their continuous counterparts. Let us properly define all the notions in the following.

Let $s > 0$, $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^n$, we define the (spherical) Gaussian function $\rho_{s,\mathbf{c}}$ and the Gaussian distribution $\mathcal{D}_{s,\mathbf{c}}$ of width s and center $\mathbf{c}$ as

$$\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / s^2) \quad \text{and} \quad \mathcal{D}_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n.$$

If it is origin-centered, we omit the subscript $\mathbf{c} = \mathbf{0}$. Gaussian distributions have two important properties:

Tail bound: an element sampled from a Gaussian distribution has (with high probability) small norm.

Sum: the sum of two Gaussian variables is again a Gaussian variable, i.e., $\mathcal{D}_s + \mathcal{D}_t = \mathcal{D}_{\sqrt{s^2+t^2}}$.

Lemma 27. *Let $s > 0$ and $\boldsymbol{x} \in \mathbb{R}^n$, then* $\Pr_{\boldsymbol{x} \hookleftarrow \mathcal{D}_s}[\|\boldsymbol{x}\| \geq \sqrt{n}s] \leq 2^{-n}$.

Discrete Gaussian. For any lattice $\Lambda \subset \mathbb{R}^n$, width $s > 0$ and center $\mathbf{c} \in \mathbb{R}^n$, we define the discrete Gaussian distribution $D_{\Lambda,s,\mathbf{c}}$ obtained by conditioning $\mathcal{D}_{s,\mathbf{c}}$ to the event of $\mathbf{x} \in \Lambda$. For $\mathbf{x} \in \Lambda$ we get:

$$D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\mathcal{D}_{s,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{y} \in \Lambda} D_{s,\mathbf{c}}(\mathbf{y})}.$$

Smoothing parameter. Unfortunately, the discrete Gaussian distribution doesn't behave automatically like a continuous one. For example, in the continuous case, it yields $\mathcal{D}_{s,\mathbf{c}}(\mathbf{x}) = \mathcal{D}_{s,\mathbf{0}}(\mathbf{x}-\mathbf{c})$. However, in the discrete case, it yields $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = D_{\Lambda+\mathbf{c},s,\mathbf{0}}(\mathbf{x}-\mathbf{c}) \neq D_{\Lambda,s,\mathbf{0}}(\mathbf{x}-\mathbf{c})$. Furthermore, sums of Gaussians are in general not Gaussian anymore. The smoothing parameter of a lattice Λ gives a threshold above which a discrete Gaussian very much behaves like a continuous one. It is denoted by $\eta_\varepsilon(\Lambda)$ for some $\varepsilon > 0$ and was introduced by Micciancio and Regev [MR07]. It is formally defined as the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^\vee \setminus \{\mathbf{0}\}) \leq \varepsilon$. As $\rho_{1/s}$ is continuous and strictly decreasing, the same holds for η_ε.

Luckily, we can bound it from above and below with the help of the first minimum of its dual lattice.

Lemma 28 ([Ban93, Lem. 1.5] and [Reg05, Claim 2.13]). *Let Λ be an n-dimensional lattice and $\varepsilon = \exp(-n)$, it holds*

$$\frac{\sqrt{n}}{\sqrt{\pi}\lambda_1(\Lambda^\vee)} \leq \eta_\varepsilon(\Lambda) \leq \frac{\sqrt{n}}{\lambda_1(\Lambda^\vee)}.$$

Example 29. *For the integer lattice $\Lambda = \mathbb{Z}^n$, we know that $\lambda_1(\Lambda^\vee) = 1$ and thus $\sqrt{n/\pi} \leq \eta_\varepsilon(\mathbb{Z}^n) \leq \sqrt{n}$ for $\varepsilon = \exp(-n)$.*

Once we are above the smoothing parameter, the total Gaussian measure of any translation of the lattice is essentially the same.

Lemma 30 ([MR07, Lem. 4.4]). *Let Λ be an n-dimensional lattice. Then, for any $\varepsilon \in (0,1)$, $s \geq \eta_\varepsilon(\Lambda)$ and $\boldsymbol{c} \in \mathbb{R}^n$, we have $\rho_{s,\boldsymbol{c}}(\Lambda) \in \left[\frac{1+\varepsilon}{1-\varepsilon}, 1\right] \cdot \rho_s(\Lambda)$.*

A very useful property is that, once the Gaussian width s exceeds the smoothing parameter, sampling a continuous Gaussian on $\mathbb{R}^n$ (or a discrete Gaussian of some larger lattice) and reducing it modulo the lattice provides a distribution that is statistically close to the uniform distribution over the lattice cosets.

Lemma 31 ([MR07, Lem. 4.1]). *Let Λ be an n-dimensional lattice, $\varepsilon > 0$, and $s > \eta_\varepsilon(\Lambda)$. Then the distribution of the coset $\boldsymbol{e} + \Lambda$, where $\boldsymbol{e} \leftarrow D_s$, is within statistical distance $\varepsilon/2$ of the uniform distribution over the cosets of Λ.*

Lemma 32 ([GPV08, Cor. 2.8]). *Let Λ, Λ' be n-dimensional lattices with $\Lambda' \subseteq \Lambda$. Then, for any $\varepsilon \in (0, 1/2)$ and $s \geq \eta_\varepsilon(\Lambda')$, and any $\boldsymbol{uc} \in \mathbb{R}^n$, the distribution of $(D_{\Lambda,s,\boldsymbol{c}} \bmod \Lambda')$ is within statistical distance 2ε of the uniform distribution over $(\Lambda \bmod \Lambda')$.*

It now holds, if $s, t > \eta_\varepsilon(\Lambda)$, the sum of discrete Gaussians on the *same* lattice defines again a discrete Gaussian distribution:

$$D_{\Lambda,s} + D_{\Lambda,t} = D_{\Lambda,\sqrt{s^2+t^2}}.$$

5.3 Hardness

As we have seen, LWE defines an instance of BDD over a (1) random (2) q-ary lattice. As for SIS, there is a rather surprising result due to Regev [Reg05] that shows that LWE (for uniform secret and Gaussian noise) is at least as hard as solving approximate SIVP and approximate GapSVP on *any* lattice.

Theorem 33. *For any $m = \mathsf{poly}(n)$, any modulus $q \leq 2^{\mathsf{poly}(n)}$ and any discrete Gaussian error distribution χ of size $\alpha q \geq 2\sqrt{n}$ (where $0 < \alpha < 1$), solving (search/decision) $\mathrm{LWE}_{n,q,\chi,m}$ with non-negligible probability is at least as hard as solving quantumly the problem GapSVP_γ and the problem SIVP_γ on arbitrary n-dimensional lattices with overwhelming probability, for some $\gamma = \widetilde{O}(n/\alpha)$.*

Clearly, the approximation factor γ decreases for larger α. The word *quantumly* here means that any efficient solvers for LWE (either quantum or classic) only leads to quantum solver for SIVP and GapSVP. Later works dequantized this reduction, but only work for GapSVP [Pei09; Bra+13]. We skip the proof (and even its idea) here, but refer to [Pei16a, Sec. 4.2] for a high level sketch.

Note that the theorem above is shown for *uniform* secrets over $\mathbb{Z}_q^n$ and *discrete Gaussian* noise distributions. Regarding other choices of secret/noise distributions, there are various results. All of them show that (under some increasing parameters) the LWE variant with different secret/noise distribution is at least as hard as 'standard' LWE with uniform secret and Gaussian noise. Such reductions exist for secrets sampled over the uniform distribution over the set $\{-\beta, \dots, \beta\}^n$ for small values of β [Gol+10; Bra+13] as well as for the noise sampled over the uniform distribution

over the same set [MP13; DM13]. However, in the latter case, we have some bound on the number of samples m that we can provide. If m gets too big, there are polynomial-time attacks [AG11; MP13].

More recently, Brakerski and Döttling introduced the notion of entropic hardness, and showed that LWE remains hard to solve for any secret distribution that has enough min-entropy [BD20].

In practical schemes, we can also find secret/noise distributions for which no hardness reductions exist. For instance, the finalist Kyber of the NIST post-quantum project[8] uses centered binomial distributions. The alternate finalist Frodo uses approximations of rounded Gaussian distributions. An earlier candidate, LAC, uses the distribution over ternary vectors with fixed Hamming weight. The hardness of the corresponding LWE variant is argued by the fact that no lattice-attack exploits the concrete structure of the distributions, only the size of the resulting coefficients.

5.4 Interlude: Regev's Public Key Encryption

5.4.1 The original description

Let m, n and q be positive integers and χ a distribution over $\mathbb{Z}$. For simplicity, assume that the LWE secret is sampled uniformly at random over $\mathbb{Z}_q^n$. The message space is $\{0,1\}$, that is, we are going to encrypt a single bit.

KGen: Sample $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{e} \leftarrow \chi^m$. Return $\mathsf{sk} = \mathbf{s}$ and $\mathsf{pk} = (\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$.

Enc: For $\nu \in \{0,1\}$, sample $\mathbf{r} \leftarrow U(\{0,1\}^m)$ and return the ciphertext $(\mathbf{u}, v)$, where $\mathbf{u}^T = \mathbf{r}^T\mathbf{A}$ and $v = \mathbf{r}^T\mathbf{b} + \lfloor q/2 \rfloor \cdot \nu$.

Dec: Compute $v - \mathbf{u}^T\mathbf{s}$. If the result is closer to 0 than to $\lfloor q/2 \rfloor$, then output $\nu' = 0$. Else output $\nu' = 1$.

Correctness. We require χ^m to provide *short* noise elements of Euclidean norm at most $\frac{q}{8\sqrt{m}}$. It yields

$$\begin{aligned} v - \mathbf{u}^T\mathbf{s} &= \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) + \lfloor q/2 \rfloor \cdot \nu - \mathbf{r}^T\mathbf{A}\mathbf{s} \\ &= \mathbf{r}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot \nu. \end{aligned}$$

Now, $\mathbf{r}$ is a binary vector and $\mathbf{e}$ sampled from χ, where χ provides elements of short norms. More precisely

$$|\mathbf{r}^T\mathbf{e}| \leq \|\mathbf{r}\| \cdot \|\mathbf{e}\| \leq \sqrt{m} \cdot \frac{q}{8\sqrt{m}} = q/8.$$

For $\nu = 0$ the value of $v - \mathbf{u}^T\mathbf{s}$ will be close to 0 and for $\nu = 1$ the value will be close to $\lfloor q/2 \rfloor$.

Security. For security, we require $m \geq 3(n+1)\log q$. IND-CPA security is proven in a game-based model. Game 0 is the IND-CPA game using the encryption scheme described above. In Game 1, the second part $\mathbf{b}$ of the public key isn't sampled honestly (as an LWE instance), but sampled uniformly at random over $\mathbb{Z}_q^m$. Assuming the hardness of decision LWE, Game 0 and Game 1 are computationally indistinguishable. In Game 2, the ciphertext $(\mathbf{u}, v)$ is now replaced by a random element sampled over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Note that $(\mathbf{r}^T\mathbf{A}, \mathbf{r}^T\mathbf{b}) = \mathbf{r}^T\mathbf{A}'$, where $\mathbf{A}' = (\mathbf{A}|\mathbf{b})$. As both $\mathbf{A}$ and $\mathbf{b}$ are random, so is $\mathbf{A}'$. Using the leftover hash lemma (Lemma 21), the ciphertext is statistically close to uniform and so the adversary can't distinguish Game 1 and Game 2. Now, in Game 2, the ciphertext contains no information on the encrypted message, so the adversary can only guess.

[8] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/

Dual Regev. As detailed out by Gentry et al. [GPV08, Sec. 7.1], it is possible to define a dual version of the Regev encryption scheme, where the roles of the LHL and the LWE problem are switched. More precisely, it uses the LHL to argue for the statistical closeness to uniform of the public key and uses then the computational hardness of decision LWE to argue for the IND-CPA security.

5.4.2 A better template

In the security proof we argued statistical closeness using the LHL. As we already need to use computational assumptions, a natural question is whether we can avoid the LHL and only use computational arguments instead. Indeed, we can do so, however, we need the LWE secret to be short. For simplicity, we assume $m = n$ and thus that $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ is quadratic.

KGen: Sample $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{n \times n})$ and $\mathbf{s}, \mathbf{e} \leftarrow \chi^n$. Return $\mathsf{sk} = \mathbf{s}$ and $\mathsf{pk} = (\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$.

Enc: For $\nu \in \{0, 1\}$, sample $\mathbf{r}, \mathbf{f} \leftarrow \chi^n$ and $f' \leftarrow \chi$ and return the ciphertext $(\mathbf{u}, v)$, where $\mathbf{u}^T = \mathbf{r}^T\mathbf{A} + \mathbf{f}^T$ and $v = \mathbf{r}^T\mathbf{b} + f' + \lfloor q/2 \rfloor \cdot \nu$.

Dec: Compute $v - \mathbf{u}^T\mathbf{s}$. If the result is closer to 0 than to $\lfloor q/2 \rfloor$, then output $\nu' = 0$. Else output $\nu' = 1$'

Correctness. We require χ to provide *short* elements of absolute value at most B such that $2mB^2 + B < q/8$. It yields

$$\begin{aligned} v - \mathbf{u}^T\mathbf{s} &= \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) + f' + \lfloor q/2 \rfloor \cdot \nu - (\mathbf{r}^T\mathbf{A} + \mathbf{f}^T)\mathbf{s} \\ &= \mathbf{r}^T\mathbf{e} - \mathbf{f}^T\mathbf{s} + f' + \lfloor q/2 \rfloor \cdot \nu. \end{aligned}$$

Now, $\mathbf{e}, \mathbf{s}, \mathbf{f}$ and f' are all sampled from χ, where χ provides elements of short norms. More precisely

$$|\mathbf{r}^T\mathbf{e} - \mathbf{f}^T\mathbf{s} + f'| \leq \|\mathbf{r}\| \cdot \|\mathbf{e}\| + \|\mathbf{f}\| \cdot \|\mathbf{s}\| + |f'| \leq 2(\sqrt{m}B \cdot \sqrt{m}B) + B < q/8.$$

For $\nu = 0$ the value of $v - \mathbf{u}^T\mathbf{s}$ will be close to 0 and for $\nu = 1$ the value will be close to $\lfloor q/2 \rfloor$.

More message bits. In order to encrypt messages of $k = \mathsf{poly}(n)$ bits, one can replace the vectors $\mathbf{r}$ and $\mathbf{f}$ by matrices $\mathbf{R}, \mathbf{F} \leftarrow \chi^{n \times k}$, defining $\mathbf{U}^T = \mathbf{R}^T\mathbf{A} + \mathbf{F}^T$ and $\mathbf{v} = \mathbf{R}^T\mathbf{b} + \mathbf{f}' + \lfloor q/2 \rfloor \cdot \mathbf{m}$.

Security. Now, to argue security, we argue several times with HNF-LWE. First, we replace (as before) the public key by some uniform vector $\mathbf{b}$. Second, we see that $(\mathbf{r}^T\mathbf{A} + \mathbf{f}^T, \mathbf{r}^T\mathbf{b} + f')$ defines another instance of LWE (in HNF) with secret $\mathbf{r}$, public matrix $(\mathbf{A}, \mathbf{b})^T$ and noise $(\mathbf{f}, f')$. So, we can replace the ciphertext by a uniform random element and again the attacker can now only guess within the IND-CPA game.

Improvements. One improvement that is used in the literature, is to use a more compact random seed together with a pseudo-random function to compute $\mathbf{A} = \mathsf{PRF}(seed_{\mathbf{A}})$. This saves storage, as one doesn't need to send the full matrix in the public key, while still guaranteeing fresh randomness for every key pair. This becomes even more important when considering PKE schemes as key-exchange mechanisms (KEM). Frodo [Bos+16] is a KEM which starts from this textbook construction together with the random seed idea (and some other improvements).

6 Connections between SIS and LWE

Decision LWE to Search SIS. This reduction is often used for attacks on LWE (called the dual attack, for more details see [APS15]). Given as input an LWE challenge $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. Forward $\mathbf{A}$ to an SIS-oracle which will respond with a short non-zero vector $\mathbf{z} \in \mathbb{Z}^m$ of norm $\|\mathbf{z}\| \leq \beta$ such that $\mathbf{z}^T\mathbf{A} = \mathbf{0} \bmod q$. Now, one can computer $\mathbf{z}^T\mathbf{b}$. If the resulting element is short, one guesses that $(\mathbf{A}, \mathbf{b})$ is a sample of LWE. Else, output that it is a uniform sample. If $(\mathbf{A}, \mathbf{b})$ is indeed an instance of LWE it yields $\mathbf{z}^T\mathbf{b} = \mathbf{z}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \mathbf{z}^T\mathbf{e}$, where $|\mathbf{z}^T\mathbf{e}| \leq \|\mathbf{z}\| \cdot \|\mathbf{e}\|$. As both $\mathbf{e}$ and $\mathbf{z}$ are short, their inner product is small as well. If $\mathbf{b}$ is uniformly random, so is $\mathbf{z}^T\mathbf{b}$, and hence with reasonable probability not short.

Duality between Knapsack and LWE. There is a reduction from (search/decision) Knapsack to (search/decision) LWE and vice versa. This equivalence is sometimes also referred to as syndrome decoding. For more details see [MM11, Sec. 4]. The high level idea is the following: Sampling a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ for sufficiently large m (i.e. $m \geq n + \omega(\log n)$), provides with overwhelming probability a non-singular matrix, that is, the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^n$. By linear algebra, we can find a matrix $\mathbf{G} \in \mathbb{Z}_q^{(m-n) \times m}$ such that $\mathbf{G}\mathbf{A} = \mathbf{0} \in \mathbb{Z}_q^{(m-n) \times n}$ and such that all columns of $\mathbf{G}$ generate $\mathbb{Z}_q^{m-n}$. We can further randomize $\mathbf{G}$ by left-multiplying it by a random unimodular matrix (recall: this doesn't change the underlying lattice). For simplicity, let's call the randomized matrix $\mathbf{G}$ as well. Let $(\mathbf{A}, \mathbf{b})$ be an input to the LWE problem. We can transform this sample to an input sample $(\mathbf{G}, \mathbf{t})$ of the knapsack by computing $\mathbf{t} = \mathbf{G}\mathbf{b}$. If $\mathbf{b}$ was uniform, so is $\mathbf{t}$. If $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, it yields $\mathbf{t} = \mathbf{G}\mathbf{e}$. In other words, the noise distribution of LWE becomes the secret distribution of the Knapsack problem. The reduction in the other direction is similar.

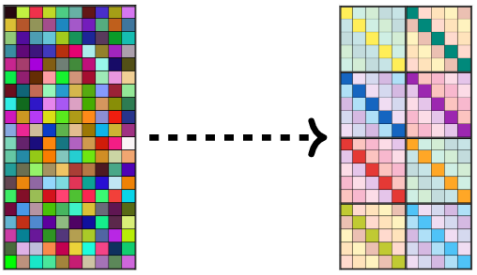
Quantumly SIS to LWE. Another connection between SIS and LWE was given in [Ste+09, Sec. 4]. More precisely, they use the duality connection between the q-ary lattice $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ to show a quantum reduction from SIS to search LWE. Using the reduction from decision LWE to SIS and search-to-decision for LWE, we can interpret this as both problems being quantumly equivalent. It is still an open problem to show the equivalence for purely classical reductions.

Part III
Structured Lattice Problems

As eluded vaguely before, the few cryptographic constructions we have seen so far are not very practical. Why is this and what is the problem with using the plain LWE and SIS problems? We can see that the public key sizes and computations are quadratic in the security parameter λ (assuming $m, n \sim \lambda$), this is quite big and slow.

Let us for instance look at the hash function $f_{\mathbf{A}}$ from Section 4.3. Simply reading the public key $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ takes time $nm \log q > n^2$ (as $m > n$ for $f_{\mathbf{A}}$ to be compressing). However, we can find pre-images or collisions in time $2^{O(m)}$ by testing all possible values of $\{0,1\}^m$. Our hope is to reduce the size of the public key to something roughly $m \approx n$. The idea:

Reduce needed storage of the public key
and speed-up the computations
by adding **structure**!



In this course, we focus on the so-called *module* variants of SIS and LWE. Those are the ones that are most used in practice as they offer a good security-efficiency balance and are better for fine-tuning concrete parameters. For simplicity, we focus on the so-called power-of-two cyclotomic rings as most practical schemes are initiated in this setting. This also helps to simplify the mathematical framework.

7 Mathematical Setting

Let us first fix the mathematical setting we are using in the following.

7.1 Ring of Polynomials

Let $\mathbb{Z}[x]$ denote the ring of polynomials with integer coefficients. Further, let $n = 2^k$ for $k \in \mathbb{N}$ be a power of two. We define the quotient ring $R := \mathbb{Z}[x]/(x^n + 1)$. Informally, this ring now contains "polynomials *modulo* $x^n + 1$". This means that x^n is identified with -1 and thus every element in R can be uniquely represented by a polynomial of degree less than n. By identifying a polynomial $f(x) = f_0 + f_1 x + \cdots f_{n-1} x^{n-1}$ with its coefficient vector $\tau(f) := (f_0, \ldots, f_{n-1})^T$ we obtain an isomorphism τ between R and $\mathbb{Z}^n$. This is sometimes called the *coefficient embedding*.

Example 34. *Let $n = 4$ and $f'(x) = -x^5 + x^4 + x^3 - 3x^2 + x + 2$ be an integer polynomial in $\mathbb{Z}[x]$. Its unique representation in $\mathbb{Z}[x]/(x^4 + 1)$ is $f(x) = -x(-1) + (-1) + x^3 - 3x^2 + x + 2 = x^3 - 3x^2 + 2x + 1$. Let $g(x) = -2x^3 + 5$ be another polynomial in $\mathbb{Z}[x]/(x^n + 1)$. We can see that the sum $f(x) + g(x) = -x^3 - 3x^2 + 2x + 6$ is again in the quotient ring. Further, we can multiply both polynomials and reduce them modulo $x^n + 1$ to obtain a well-defined multiplication operation*

in this same ring. More precisely,

$$\begin{aligned} f(x) \cdot g(x) &= -2x^6 + 6x^5 - 4x^4 + 3x^3 - 15x^2 + 10x + 1 \\ &= -2x^2(-1) + 6x(-1) - 4(-1) + 3x^3 - 15x^2 + 10x + 1 \\ &= 3x^3 - 13x^2 + 4x + 5. \end{aligned}$$

Interestingly, $f(x)g(x)$ can also be written as a matrix-vector product $\mathrm{Rot}(f) \cdot \tau(g)$, where the matrix $\mathrm{Rot}(f) \in \mathbb{Z}^{n \times n}$ only depends on $f(x)$ and $\tau(g) := (g_0, \dots, g_{n-1})^T \in \mathbb{Z}^n$ is the coefficient vector of the polynomial $g(x) = g_0 + g_1 x + \cdots + g_{n-1}x^{n-1}$.

More precisely, for a polynomial $f(x) = f_0 + f_1 x + \cdots f_{n-1}x^{n-1}$, we define the matrix $\mathrm{Rot}(f)$[9] as

$$\mathrm{Rot}(f) = \begin{bmatrix} f_0 & -f_{n-1} & \cdots & -f_1 \\ f_1 & f_0 & \cdots & -f_2 \\ \vdots & \vdots & \ddots & \vdots \\ f_{n-1} & f_{n-2} & \cdots & f_0 \end{bmatrix} \in \mathbb{Z}^{n \times n}.$$

We can now define *vectors* over R, where every coefficient of such a vector is given by one polynomial. Let $\mathbf{f} = (f_j)_{j \in [d]}$ and $\mathbf{g} = (g_j)_{j \in [d]}$ both be vectors in R^d with $d \in \mathbb{N}$. Then their inner product $\langle \mathbf{f}, \mathbf{g} \rangle$ can be written as

$$[\mathrm{Rot}(f_1) | \cdots | \mathrm{Rot}(f_d)] \cdot \tau(g_j)_{j \in [d]} \in \mathbb{Z}^n.$$

In the same manner, we can define *matrices* over R, where every entry of such a matrix is given by one polynomial. Let $\mathbf{F} = (f_{kj})_{k \in [m], j \in [d]} \in R^{m \times d}$ and $\mathbf{g} = (g_j)_{j \in [d]} \in R^d$. Then their matrix-vector product $\mathbf{F} \cdot \mathbf{g}$ can be written as

$$\begin{bmatrix} \mathrm{Rot}(f_{11}) & \cdots & \mathrm{Rot}(f_{1d}) \\ \mathrm{Rot}(f_{21}) & \cdots & \mathrm{Rot}(f_{2d}) \\ \vdots & & \vdots \\ \mathrm{Rot}(f_{m1}) & \cdots & \mathrm{Rot}(f_{md}) \end{bmatrix} \cdot \tau(g_j)_{j \in [d]} \in \mathbb{Z}^{nm}.$$

Overall, we now see how we can replace a fully random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ by some structured matrix using properties of R, see Figure 7.1.

Geometry. The coefficient embedding τ also allows us to define a geometry on R. We can measure lengths and distances by setting $\|f\| := \|\tau(f)\|$, where the latter is the standard Euclidean norm over $\mathbb{Z}$. We extend this definition component-wise to vectors in the natural way. Furthermore, we can sample elements over R by sampling the polynomial's coefficients independently over $\mathbb{Z}$. In particular, we have different ways of sampling from a Gaussian distribution over R. Either, we can sample every coefficient from some discrete Gaussian distribution over $\mathbb{Z}$ or directly sample from a discrete Gaussian distribution over $\mathbb{Z}^n$.

Choice of n. The reason why n is taken as a power of two is that in this case $x^n + 1$ is irreducible, guaranteeing that R is an integral domain, which in turn prevents some standard attacks. It also benefits from a very rich algebraic structure as it is the $2n$-th cyclotomic polynomial and hence R is the ring of integers of a cyclotomic field. Those rings are well understood and have very nice algebraic properties. For this crash course however, we won't dig deeper into the mathematical background :)

7.2 Module Lattices

We now explain how this mathematical setting gives rise to Euclidean lattices with additional structure.

[9] Because of its special structure this matrix is often called *nega-cyclic* or *anti-cyclic* matrix.

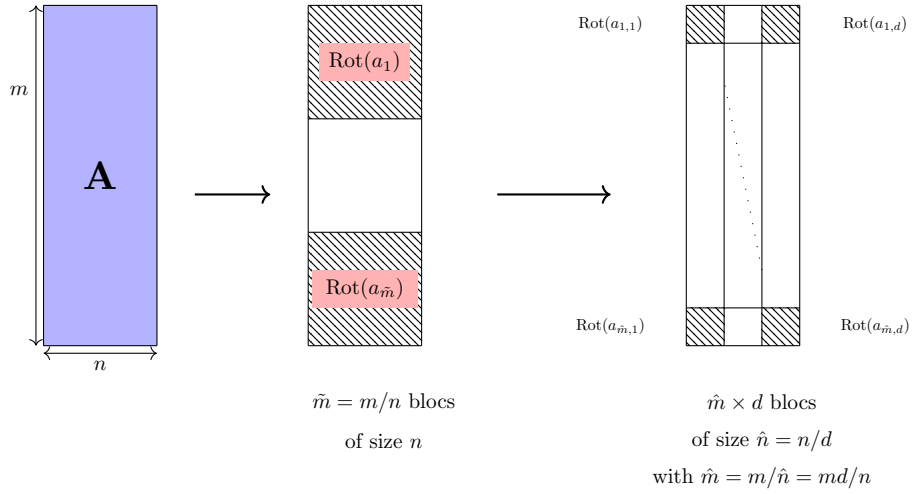


Figure 11: Main idea of structured variants. Whereas we need m random rows to describe the fully random matrix $\mathbf{A}$, we only need m/n random rows in the second and md/n in the last case.

Ideal lattices. Let us recall some definitions of our algebra courses. An ideal $\mathcal{I} \subseteq R$ is an additive subgroup of a ring R that is closed under multiplication by any ring element. That is, $\mathcal{I}$ is closed under addition (and substraction) *and* for any $y \in \mathcal{I}$ and $r \in R$, it yields $ry \in \mathcal{I}$. Using the coefficient embedding τ that we defined above, we can embed the ideal into $\mathbb{Z}^n$. The subgroup property makes sure that the image $\tau(\mathcal{I})$ is a lattice in $\mathbb{Z}^n$.[10] And the additional ideal property makes this lattice a special lattice, that we call *ideal lattice*. In particular, if we take a monomial x^j for some $j \in [n-1]$ (which is an element of our ring $R = \mathbb{Z}[x]/(x^n+1)$) and multiply it with any element of the ideal $y \in \mathcal{I}$, we have $\|x^j y\| = \|y\|$. The key insight here is that multiplying by a monomial only shifts the coefficient vector and possibly changes the sign, but doesn't change the norm of the coefficient vector. Hence, any element $y \in \mathcal{I}$ gives rise to n linearly independent vector $y, yx, yx^2, \ldots, yx^{n-1}$. This means, once we found *one* shortest vector, we actually found n linearly independent shortest vector. That is, $\lambda_1(\tau(\mathcal{I})) = \cdots = \lambda_n(\tau(\mathcal{I}))$ (see Section 1). Recall that for any lattice, there is a reduction from SIVP_γ to $\mathrm{SVP}_{\gamma\sqrt{n}}$. In the ideal lattice setting for power-of-two cyclotomics, the reduction is improved by a factor of $\sqrt{n}$ in the approximation factor.[11]

Module lattices. For what follows, we are not interested in ideals over R, but in modules over R. Informally, modules can be thought of a generalization of the real vector space $\mathbb{R}^d$ for some positive integer d. In $\mathbb{R}^n$, the scalar multiplication is defined by elements over $\mathbb{R}$. Now, we are looking at subsets $\mathcal{M} \subseteq R^d$, where the scalar multiplication is defined by elements of R.[12] Note that ideals are modules for $d = 1$. By applying the coefficient embedding τ to every coefficient of vectors in $\mathcal{M}$, we can define the set $\tau(\mathcal{M}) \subset (\mathbb{Z}^n)^d = \mathbb{Z}^{nd}$. As for the ideal case, this defines a lattice of dimension nd with special properties, as it is closed under scalar multiplication. As before, we can multiply any element $\mathbf{y}$ of $\mathcal{M}$ by a monomial x^j for some $j \in [n-1]$ and this won't change the norm of $\mathbf{y}$. However, $x^j\mathbf{y}$ only gives us n linearly independent vectors, not $n \cdot d$. So, an oracle for Mod-SVP isn't enough to solve Mod-SIVP.

Canonical embedding. Throughout this crash course we use the coefficient embedding, because it is much more intuitive and requires less mathematical background knowledge. However, in many

[10] Most of the time, $\mathcal{I}$ is used to design the ideal *and* the corresponding ideal lattice at the same time.
[11] With respect to the so-called canonical embedding, this is true for all cyclotomics.
[12] Here, we simplify things a bit and talk about modules as subsets of R^d, but actually they are subsets of K^d, where K is the corresponding cyclotomic field to R.

of the quoted work, another embedding is used which may be seen as more appropriate from an algebraic point of view. It is called canonical embedding and has the nice property that not only addition, but also multiplication of two ring elements is component-wise with respect to it. That means that the corresponding "rotation" matrix becomes a diagonal matrix. For more details we refer to the discussion by Lyubashevsky et al. [LPR13] in Section 1.2.

Hardness of ideal lattice problems. An important question that arises when considering structured lattices, is how hard are standard problems, such as searching for a shortest vector, applied to this class of lattices? Informally, we don't know (yet?) of any algorithm exploiting the module structure, but we do know algorithms that exploit the *ideal* structure. In the following we focus on the hardness of Id-SVP (i.e., SVP restricted to ideal lattices).

On the one hand, some works have proven worst-case to average-case reductions for problems in ideal lattices [Gen09; Boe+20]. They proved that there exist distributions over the set of ideal lattices such that an ideal chosen from this distribution is "as hard as possible". More formally, if one can solve Id-SVP for such random lattices with non-negligible probability, then one can solve Id-SVP in any ideal lattice.

On the other hand, several works have shown weaknesses of Id-SVP for specific choices of ideals or parameters. Cramer et al. [Cra+16] showed that Id-SVP can be solved in quantum polynomial time for *principal* ideals (i.e., ideals generated by a single ring element) of cyclotomic fields, when the generator is sampled from a Gaussian distribution. It is also known that the relaxed variant of Id-SVP with a large approximation factor $\approx 2^{\sqrt{d}}$ can be solved in quantum polynomial time in cyclotomic fields of degree d [CDW21]. In 2021, Pan et al. [Pan+21] showed that, for some prime ideals with a lot of symmetries (in Galois number fields), the Id-SVP problem could be solved classically in polynomial time. The result has been generalized to any ideal (whose prime factors are not ramified) over any number fields by Boudgoust et al. [BGP22]. Finally, there is also a line of work targeting Id-SVP for all ideals of all number fields [PHS19; BR20; Ber+21]. However, those algorithms require some exponential-time pre-processing, and are at the moment no better than lattice reduction algorithms that work on unstructured lattices (such as BKZ).

8 Module Variants

A nice talk that summarizes all module lattice problems that are relevant for lattice based cryptography - and that we are going to define just below - was given by Damien Stehlé as an invited talk at PQCrypto 2021, still available online [here].

8.1 Module Short Integer Solution

Throughout the section, we assume that computations are done over the ring $R = \mathbb{Z}[x]/(x^n + 1)$. For an integer q we set $R_q = R/(qR) = \mathbb{Z}_q[x]/(x^n + 1)$. We now define the Module Short Integer Solution problem, abbreviated M-SIS, which was first introduced by Langlois and Stehlé [LS15].

Definition 35 (Module-SIS)**.** *Let m, d and q be positive integers and β be a positive real. Given m independent vectors $\boldsymbol{a}_j$ sampled uniformly at random over R_q^d, forming the columns of a matrix $\boldsymbol{A} \in R_q^{d\times m}$, the problem* M-SIS$_{d,q,\beta,m}$ *asks to find a nonzero vector $\boldsymbol{z} \in R^m$ of norm $0 < \|\boldsymbol{z}\| \leq \beta$ such that*

$$\boldsymbol{A}\boldsymbol{z} = \sum_{i=1}^{m} z_i \boldsymbol{a}_i = \boldsymbol{0} \bmod q.$$

Actually, M-SIS generalizes plain SIS, simply take $n = 1$ and thus $R = \mathbb{Z}$. Another special case of M-SIS is given for $d = 1$, which is called R-SIS. In this case the matrix $\mathbf{A}$ is composed of one single row and thus we are considering the inner-product of this single row with $\mathbf{z}$. Note that, historically, R-SIS was introduced before M-LWE in two concurrent works by Peikert and Rosen [PR06] and Lyubashevsky and Micciancio [LM06]. One can define an inhomogeneous version and Hermite normal form as for SIS.

Gain in Efficiency. Recall from the discussion above that every entry a_{ik} in the matrix $\mathbf{A}$ defines a matrix of multiplication $\mathrm{Rot}(a_{ik}) \in \mathbb{Z}_q^{n\times n}$. Hence, the matrix $\mathbf{A}$ defines the matrix $\mathrm{Rot}(\mathbf{A}) \in \mathbb{Z}_q^{nd\times nm}$. Wheres in the plain setting, we had to store $(nd)(nm)\log q$ bits, we now only have to store $(nd)m\log q$ bits (as one row of $\mathrm{Rot}(\mathbf{A})$ defines the $n-1$ next rows). Hence, we gain a factor of n. Regarding computation, we can make use of FFT-like techniques to compute $z_i a_{ij}$ in quasi-linear time $O(n\log n)$. Hence, computing $\mathbf{Az}$ can be done in $O(dmn\log n)$, where we again save a factor n.

Hidden Structured Lattice. We can interpret M-SIS as a problem over *random* q-ary module lattices. More precisely, M-SIS defines an instance of Mod-SVP$_\gamma$ in the random module lattice

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{y} \in R^m \colon \mathbf{Ay} = \mathbf{0} \bmod q\},$$

where the approximation factor γ depends on the norm β.

Hardness. Similar to its unstructured counterpart, M-SIS also enjoys worst-case to average-case connections for suitable parameter choices from lattice problems such as SIVP$_\gamma$. Whereas the hardness results for SIS start from lattice problems in the class of general Euclidean lattices, the set has to be restricted to module lattices in the case of M-SIS. Recall, these module lattices correspond to modules in the ring R and we refer to the related lattice problem as Mod-SIVP$_\gamma$. The hardness theorem for M-SIS strongly resembles Theorem 20 on the hardness of plain SIS, where one replaces n simply by nd and one restricts the corresponding lattice problem to module lattices.

Theorem 36. *For any $m = \mathsf{poly}(n)$ and $d \in \mathbb{N}$, any $\beta > 0$ and any $q \geq \beta \cdot \mathsf{poly}(nd)$, solving* M-SIS$_{d,q,\beta,m}$ *with non-negligible probability is at least as hard as solving the problem* Mod-SIVP$_\gamma$ *on arbitrary nd-dimensional module lattices with overwhelming probability, for some $\gamma = \beta \cdot \mathsf{poly}(nd)$.*

For instance, [LS15, Thm. 3.6] requires $q \geq \beta \cdot \sqrt{nd} \cdot \omega(\log nd)$ and $\gamma \geq \beta \cdot \sqrt{nd} \cdot \omega(\sqrt{\log(nd)})$.

8.2 Module Learning With Errors

The module variant of LWE was first defined by Brakerski et al. [BGV12] and thoroughly studied by Langlois and Stehlé [LS15]. All following three definitions can be obtained by replacing $\mathbb{Z}$ by R and n by d in the corresponding definitions of LWE.

Definition 37 (Module-LWE distribution)**.** *Let d and q be positive integers and let χ be a distribution over R. For a fixed secret $\boldsymbol{s} \in R_q^d$, the* M-LWE *distribution $A_{\boldsymbol{s},\chi}$ over $R_q^d \times R_q$ is obtained by choosing $\boldsymbol{a} \leftarrow U(R_q^d)$, $e \leftarrow \chi$ and outputting $(\boldsymbol{a}, b = \langle \boldsymbol{s}, \boldsymbol{a}\rangle + e \bmod q)$.*

Again, there is a search and a decision variant.

Definition 38 (Search Module-LWE)**.** *Let m be a positive integer. Given m independent samples $(\boldsymbol{a}_i, b_i) \in R_q^d \times R_q$ from $A_{\boldsymbol{s},\chi}$ for a uniformly random $\boldsymbol{s} \in R_q^d$. The problem search-*M-LWE$_{d,q,\chi,m}$ *asks to find $\boldsymbol{s}$.*

We usually write $\mathbf{A} \in R_q^{m\times d}$ as the matrix whose rows are given by the M-LWE samples $\mathbf{a}_i$.

Definition 39 (Decision Module-LWE)**.** *Let m be a positive integer. Given m independent samples $(\boldsymbol{a}_i, b_i) \in R_q^d \times R_q$ that are either drawn from $A_{\boldsymbol{s},\chi}$ for a uniformly random $\boldsymbol{s} \in R_q^d$ or drawn from the uniform distribution $U(R_q^d \times R_q)$. The problem dec-*M-LWE$_{d,q,\chi,m}$ *asks to distinguish both cases with non-negligible advantage.*

Again, we can define variants of this problem, where we modify the secret and/or the noise distribution. Further, we can define the Hermite Normal Form and the Learning With Rounding variant.

Hidden Structured Lattice. As for M-SIS, we can interpret M-LWE as a problem over *random* q-ary module lattices. More precisely, M-LWE defines an instance of BDD (restricted to module lattices) in the random lattice

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in R^m : \mathbf{y} = \mathbf{A}\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in R^d\},$$

where $\mathbf{b}$ is the target point which distance from the lattice is given by the noise $\mathbf{e} \leftarrow \chi^m$.

Hardness. Similarly to M-SIS, the module version of LWE benefits from worst-case to average-case reductions from problems over module lattices. The hardness theorem for M-LWE strongly resembles Theorem 33 on the hardness of plain LWE, where the lattice problem is now on nd-dimensional module lattice. It was proven by Langlois and Stehlé [LS15, Thm. 4.7].

Theorem 40. *For any $m = \mathsf{poly}(n)$, any modulus $q \leq 2^{\mathsf{poly}(n)}$ and any discrete Gaussian error distribution χ of size $\alpha q \geq 2\sqrt{d}\cdot\omega(\sqrt{\log n})$ (where $0 < \alpha < 1$), solving (search/decision)* M-LWE$_{d,q,\chi,m}$ *with non-negligible probability is at least as hard as solving quantumly the problem* Mod-GapSVP$_\gamma$ *and the problem* Mod-SIVP$_\gamma$ *on arbitrary nd-dimensional module lattices with overwhelming probability, for some $\gamma = \widetilde{O}(n\sqrt{d}/\alpha)$.*

Later works have dequantized this reduction, but, similarly to the plain setting, only work for Mod-GapSVP$_\gamma$ [Bou+20] and large enough rank d. Again, we can show that M-LWE doesn't become significantly easier to solve if we set the noise and/or the secret distribution to be the uniform distribution of polynomials with coefficients in $\{-\beta, \dots, \beta\}^n$ [**BoudgoustJRW22b**]. Further, the entropic hardness has been shown for M-LWE via two incomparable techniques by Lin et al. [LWW20] and Boudgoust et al. [**BoudgoustJRW22**]. As an additional feature, a converse reduction from M-LWE to Mod-SIVP$_\gamma$ is proven for the special case of power-of-2 cyclotomics [LS15] and improved by Wang and Wang [WW19] for all cyclotomic fields.

Other ring structures. We remark that it is possible to define M-LWE with respect to rings of the form $\mathbb{Z}[x]/f(x)$, for an irreducible polynomial $f(x)$ other than $x^n + 1$. However, not for all choices of $f(x)$ this is a good idea. More information can be found in Peikert's paper on how *not* to instantiate R-LWE/M-LWE [Pei16b].

8.3 Special Role of Ring-LWE and Ring-SIS

In Part II we have introduced the plain SIS and LWE problems together with the hardness reduction from worst-case SIVP to the average-case SIS or LWE. We can observe that we actually have an equivalence. If we have an oracle at hand that solves SIVP$_\gamma$ on an arbitrary lattice Λ, then we can solve SIS for the bound $\beta = \gamma \cdot \lambda_n(\Lambda)$. For LWE, we note that it defines an instance of BDD for some special lattice and there is a reduction from BDD to SIVP (via a problem that is called unique-SVP and via GapSVP). Hence, if we can solve SIVP on arbitrary lattices, we can also solve LWE (for some parameters).

How does the situation looks for structured variants for LWE and SIS? Interestingly, there seems to be a difference between modules of rank exactly one (i.e., ideals) and modules of rank at least 2. First, note that the rank of the module lattice defined by M-SIS (i.e., $\Lambda_q^\perp(\mathbf{A})$) and M-LWE (i.e., $\Lambda_q(\mathbf{A})$) depends on m. Further, if we take $d = 1$ (the ring case), then we have to choose $m > 1$. If we would take $m = 1$, then R-SIS wouldn't possess a solution and decision R-LWE would become vacuously hard (for $a \in R_q^\times$).

The worst-case to average-case reductions we have seen before applied for rank 1 modules says that there is a reduction from Id-SIVP to R-SIS (Theorem 36) and to R-LWE (Theorem 40). Using the observation done just above, we obtain a reduction from R-LWE (or R-SIS) to Mod-SIVP, not to Id-SIVP. So Id-SIVP and R-LWE are not equivalent. In particular, all algorithms that improve the best known attacks against SIVP over ideal lattices, don't change the state of the art of attacks against R-LWE (and even less for M-LWE).

8.4 Subtleties over Number Fields

Even though, it seems simple to obtain M-SIS and M-LWE simply by replacing the integers $\mathbb{Z}$ by the ring of integers R of a number field K, it isn't so straightforward for everything.

One very instructive example is the Leftover Hash Lemma that we have proven in Section 4.3. To prove Lemma 21, we required q to be prime so that $\mathbb{Z}_q$ becomes a field, where every non-zero element is a unit. However, for q prime R_q may not be a field, and in particular, not every non-zero element is a unit.

There have been several solutions to this problem. One solution, as detailed out by Lin et al. [LWW20], is to restrict ourselves to number fields and modulus such that R_q is indeed a field. This is a very strong restriction, as there are number fields where no such q exists. For example, in the case of the m-th cyclotomic number field, we need to require (at least) that $(\mathbb{Z}/m\mathbb{Z})^\times$ is cyclic. For power-of-two cyclotomics, this is not the case.

Another solution is to not use the way via universal families of hash functions, but to prove the LHL "by hand". This has first been done by Micciancio [Mic07] for the case of $R = \mathbb{Z}[x]/(x^n - 1)$ and later generalized by different works. The most general version can be found in my thesis [Bou21].

An orthogonal way is to make use of discrete Gaussians of width above the smoothing parameter of R (seen as a lattice). This has been shown for all number fields by Roşca et al. [RSW18].

8.5 Interlude: Fiat-Shamir with Aborts Signatures

In the following we describe a blueprint to construct a signature scheme whose hardness is based on module lattice problems. It follows the so-called Fiat-Shamir with Aborts paradigm, first introduced by Lyubashevsky [Lyu12]. It can be seen as a lattice analogue of the Schnorr signature, but with some important caveats. The following scheme is essentially the signature scheme from Güneysu et al. [GLP12] (adapted from the ring to the module setting). This design paradigm builds also the starting point of Dilithium [Duc+18], a finalist in the ongoing standardization process run by NIST.[13] The second strategy (not covered in the course) to construct lattice-based signatures is to follow the so-called GPV-approach [GPV08]. It can be seen as a hash-then-sign analogue, but again, some important caveats need to be taken into account.

Setting. Let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, with n a power of two and q a prime such that $q = 1 \bmod 2n$. For $k, \ell \in \mathbb{N}$, let $\mathbf{A} \in R_q^{k\times\ell}$ follow the uniform distribution and be a public shared parameter of the system. The number of columns ℓ and the number of rows k should be adapted to the required security level, but usually they are small constants. Let $H_c\colon \{0,1\}^* \to C = \{c \in R\colon \|c\|_1 = d, \|c\|_\infty = 1\}$ be a random oracle with d such that $|C| > 2^{2\lambda}$, where λ denotes the required security level. Let $s, \beta \in \mathbb{Z}$ and the message space $\mathcal{M} = \{0,1\}^*$. We rely on the key set $S_\beta = \{a \in R\colon \|a\|_\infty \leq \beta\}$ with $\beta \in \mathbb{N}$. Finally, let $\mathcal{D}$ denote a distribution over $R^{\ell+k}$ providing (with overwhelming probability) vectors of norm at most B and to which we associate a rejection probability $\Pr_{rej}$.

The signature scheme $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$ from [GLP12] is illustrated in Figure 12.

Description. The algorithm KGen samples a secret key vector $\mathbf{s}$, composed of elements of R with coefficients of size at most β, and sets the verification key to $\mathbf{t} = [\mathbf{A}|\mathbf{I}_k] \cdot \mathbf{s} \in R_q^k$. At the beginning of the signing procedure, a masking vector $\mathbf{y}$ following the distribution D is sampled. The signing party then computes $\mathbf{u} = [\mathbf{A}|\mathbf{I}_k] \cdot \mathbf{y} \in R_q^k$, which serves together with the message m as input to the random oracle H_c. The output c of H_c is a polynomial in R with exactly d coefficients that are ± 1 and the remaining coefficients are 0. The second part of a potential signature is defined as $\mathbf{z} = \mathbf{s} \cdot c + \mathbf{y}$. In order to make the distribution of the signature independent of the secret key, the algorithm only outputs the potential signature with probability $\Pr_{rej}$. This step is called rejection sampling. In order to verify σ, the verifier first re-constructs the hash value $c = H_c(\mathbf{u}, m)$ and then checks if the norm of $\mathbf{z}$ is smaller than B and that $[\mathbf{A}|\mathbf{I}_k] \cdot \mathbf{z} = \mathbf{t} \cdot c + \mathbf{u}$.

[13] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/

$\mathsf{KGen}(1^\lambda)$:	sample $\mathbf{s} \leftarrow U(S_\beta^{\ell+k})$
	set $\mathsf{sk} = \mathbf{s}$ and $\mathsf{vk} = \mathbf{t} = [\mathbf{A}\|\mathbf{I}_k] \cdot \mathbf{s} \in R_q^k$
	return $(\mathsf{sk}, \mathsf{vk})$
$\mathsf{Sig}(\mathsf{sk}, m)$:	set $\mathbf{z} = \perp$
	while $\mathbf{z} = \perp$ *do*:
	sample $\mathbf{y} \leftarrow D$
	set $\mathbf{u} = [\mathbf{A}\|\mathbf{I}_k] \cdot \mathbf{y} \in R_q^k$
	compute $c = H_c(\mathbf{u}, m) \in C$
	set $\mathbf{z} = \mathbf{s} \cdot c + \mathbf{y}$
	with probability $1 - \Pr_{rej}$
	set $\mathbf{z} = \perp$
	return $\sigma = (\mathbf{u}, \mathbf{z})$
$\mathsf{Vf}(\mathsf{vk}, \sigma, m)$:	re-construct $c = H_c(\mathbf{u}, m)$
	if $\|\mathbf{z}\|_2 < B$ and $[\mathbf{A}\|\mathbf{I}_k] \cdot \mathbf{z} = \mathbf{t} \cdot c + \mathbf{u}$,
	then return 1
	else return 0

Figure 12: The signature scheme from [GLP12] with minor modifications.

The parameters β, d, B and $\Pr_{rej}$ have to be set strategically such that the scheme is correct, efficient and secure, see [Lyu12; Duc+18].

Distribution $\mathcal{D}$. For simplicity, we leave the concrete definition of the distribution $\mathcal{D}$ open. In the literature, mainly two instantiations have been studied: the discrete Gaussian distribution over R and the uniform distribution over the set of elements of small norms [Lyu09; Lyu12]. The literature provides concrete formulas for the rejection probability $\Pr_{rej}$ and the bound B. For example, for $\mathcal{D} = D_s^{k+\ell}$ the discrete Gaussian distribution of width s, the bound B comes from the Gaussian tail bound (Lemma 27) and the rejection probability can be computed as $\min(1, D_s^{\ell+k}(\mathbf{z})/M \cdot D_{c\cdot\mathbf{s},s}^{\ell+k}(\mathbf{z}))$, where M is a constant that depends on β (the Euclidean norm of the secret $\mathbf{s}$) and d (the ℓ_1-norm of the challenge c).

Lattice Peculiarities. We have already seen one caveat in the lattice setting. As $\mathbf{s}$, c and $\mathbf{y}$ are all elements of short norm, $\mathbf{z}$ would leak information on $\mathbf{s}$ if we wouldn't apply the rejection sampling. In order to save in the size of the signature, we can send c instead of $\mathbf{u}$ in the signature. By verifying if $c = H_c([\mathbf{A}|\mathbf{I}_k] \cdot \mathbf{z} - \mathbf{t} \cdot c, m)$, and that the norm of $\mathbf{z}$ is small enough, we can equivalently verify the signature. This does not only reduce the dimension of the vector from k to 1, but also the total bit-length from $nk \log_2 q$ to $n \log_2 3$, as $\mathbf{u}$ can be any vector in R_q^k but c is a polynomial with ternary coefficients.

Security. We now discuss the high level idea how to prove the security of $\Pi = (\mathsf{KGen}, \mathsf{Sig}, \mathsf{Vf})$ as specified in Figure 12. It follows the original work of Lyubashevsky and proves the security based on the hardness of M-LWE and M-SIS in two steps.

Step 1: Modify the signing algorithm so that it doesn't depend on the secret key anymore. As, after the rejecting sampling, the distribution of the response $\mathbf{z}$ is independent of the secret key $\mathbf{s}$, we can change the order of computing the response $\mathbf{z}$ and the commitment $\mathbf{u}$. More precisely, the signing procedure now first samples $\mathbf{z} \leftarrow \mathcal{D}$ and the challenge $c \leftarrow U(C)$ and only then computes $\mathbf{u} = [\mathbf{A}|\mathbf{I}_k]\mathbf{z} - \mathbf{t}c$ and programs $H_c(\mathbf{u}, m) = c$. One can show that with only very small probability the random oracle has been queried on $(\mathbf{u}, m)$ before.

Step 2: Modify the key generation algorithm. The key generation algorithm is modified so that the secret key $\mathbf{s}$ comes from the set $S_{\beta'}^{\ell+k}$ with larger coefficients ($\beta' > \beta$). The bound β' is chosen in such a way that with high probability multiple solutions for a given $\mathbf{t}$ exist. Assuming the hardness of M-LWE both games are computationally close.

Now, once the adversary outputs a forgery, the General Forking Lemma [BN06] is applied to obtain two signatures $(\mathbf{z}_1, c_1)$ and $(\mathbf{z}_2, c_2)$ for the same message and the same commitment $\mathbf{u}$. As both signatures are valid, it yields $[\mathbf{A}|\mathbf{I}_k]\,(\mathbf{z}_1 - c_1\mathbf{s} - \mathbf{z}_2 + c_2\mathbf{s}) = 0$. This solves M-SIS.

9 NTRU

9.1 NTRU Problem

The NTRU problem was introduced in the 90's by Hoffstein et al. [HPS98] and is the underlying hardness assumption of the now well-known NTRU encryption/signature scheme. The acronym stands for *Number Theory Research Unit*. On the positive side, it gives rise to quite efficient schemes that have withstood roughly 30 years of cryptanalysis (when correctly parametrized). Furthermore, it can be rephrased as a problem over random q-ary module lattices, making it a presumably quantum-resistant hardness source. However, on the negative site, it doesn't benefit (yet?) from worst-case to average-case reductions (in contrast to M-SIS and M-LWE from before). More on that later. Let us first introduce the problem.

Originally, the problem was defined over the ring $\mathbb{Z}[x]/(x^n - 1)$ for any prime n. For the sake of simplicity (and coherence) let us stick to the ring $R = \mathbb{Z}[x]/(x^n + 1)$, where n is a power-of-two. We denote by $R_q^\times$ the elements of R_q that are invertible, i.e., for $f \in R_q^\times$ it exists a f_q^{-1} such that $f \cdot f_q^{-1} = 1 \bmod (x^n + 1, q)$.

Definition 41 (NTRU distribution)**.** *Let q be a positive integer and χ be a distribution over R_q. For a fixed $f \in R_q^\times$, the* NTRU *distribution $N_{f,\chi}$ over R_q is obtained by sampling $g \leftarrow \chi$, and outputting $h = g/f \in R_q$.*

Note that, both f and g will be chosen to have very small coefficients, but the inverse of f in general won't have small coefficients anymore and hence h neither. It has a search and decision variant, similarly to LWE.

Definition 42 (Search NTRU)**.** *Let β be a positive real. Given a sample $h \in R_q$ from $N_{f,\chi}$, find $(z_1, z_2) \in R^2$ such that $z_1 + h \cdot z_2 = 0 \bmod q$ and $0 < \|(z_1, z_2)\| \leq \beta$.*

Here, we denote by $\|(z_1, z_2)\|$ the norm of the vector obtained by concatenating the coefficient vector of z_1 and of z_2. Informally, one can interpret NTRU as some kind of *homogeneous* version of R-LWE in HNF, where the term $z_1 + h \cdot z_2$ needs to be 0 modulo q.

The decision variant has also been called 'Decision Small Polynomial Ratio' (DSPR) problem [LTV12], and 'NTRU Decisional Key Cracking' problem [Ste14].

Definition 43 (Decision NTRU)**.** *Given a sample $h \in R_q$, the problem decision-$\mathrm{NTRU}_{q,\chi}$ asks to distinguish between the real case where $h \leftarrow N_{f,\chi}$ for some randomly chosen f in $R_q^\times$, and the random case where $h \leftarrow U(R_q)$.*

Module variant. It is also possible to formulate a module variant of NTRU in the natural way, as for instance studied by Chuengsatiansup et al. [Chu+20]. Instead of sampling single ring elements g, f, one can sample matrices $\mathbf{G}, \mathbf{F}$ (where $\mathbf{F}$ is required to be invertible over R_q and set the output of the module NTRU distribution as $\mathbf{H} = \mathbf{G} \cdot \mathbf{F}_q^{-1}$.

Hidden Structured Lattice. As for M-SIS and M-LWE, we can interpret NTRU as problem over *random* q-ary module lattices. More precisely, search NTRU defines an instance of SVP (restricted to module lattices of rank 2) in the random lattice

$$\Lambda_q^\perp(h) = \{(z_1, z_2) \in R^2 \colon h \cdot z_1 + z_2 = 0 \bmod q\}.$$

We won't go into details here, but it is actually a promise variant of the shortest vector problem, as we have a guarantee that the norm of a shortest vector is much smaller than what we would expect from a random lattice. This promise variant is called *unique* SVP.

Multiple samples. One can also generalize the definitions above to multiple sample of the NTRU distribution. In this case, the input to (search/decision) NTRU is of the form $h_1, \dots, h_t$ for t samples. The hidden lattice becomes

$$\Lambda_q^\perp(h_1, \dots, h_t) = \{(z_0, z_1, \cdots, z_t) \in R^{t+1} : h_j \cdot z_0 + z_j = 0 \bmod q \forall j \in [t]\}.$$

According to the current state-of-the-art, handing out multiple samples for the same private f is not insecure, as long as the parameters are instantiated appropriately. There is some small loss in concrete security, as the promise gap in the unique SVP becomes slightly larger.

Hardness. When f and g are sampled from a (discrete) Gaussian distribution with standard deviation larger[14] than $\sqrt{q}$, decision NTRU becomes vacuously hard, as the NTRU distribution gets statistically close to uniform [SS11, Theorem 3.2].

For *overstretched* parameter choices, where the modulus q is subexponentially large in the ring degree, the NTRU problems have been shown to be solvable in polynomial time, e.g. [DW21]; however, for cryptographic applications we are interested in much smaller values of q similar to that used in the original NTRU cryptosystem [HPS98], for which no efficient attacks are known against either search or decision NTRU problems.

Recently, a variant of the search variant of NTRU has been reduced from SVP over ideal lattices [PS21]. But there is neither a search to decision reduction nor a direct worst-case to average-case reduction to decision NTRU.

NTRU to Ring-LWE. As pointed out by Peikert [Pei16a], there is a rather simple reduction from decision NTRU to search R-LWE. Given independent samples $h_i \in R_q$, the reduction samples a fixed secret s and independent noise elements e_i for R-LWE from the corresponding distribution. It then inputs (h_i, b_i) to the R-LWE oracle, where $b_i = h_i \cdot s + e_i$. The reduction then outputs 'NTRU' if the R-LWE oracle's output equals s. Else, it outputs 'Uniform'. If h_i was sampled from the uniform distribution, so was (h_i, b_i) a correctly distributed R-LWE instance and hence the oracle's output was correct. If, however, $h_i = g_i/f$, it yields $b_i = (g_i \cdot s)/f + e_i$, where the information of s is theoretically hidden if the error distribution of R-LWE is sufficiently wider than the secret distribution of NTRU.

Considering this reduction an the reduction from SVP over ideal lattices to search NTRU, it seems that the hardness of NTRU lies between ideal and module SVP (for rank at least 2). It is an open problem and interesting research direction to better understand its precise position with respect to those structured lattice problems.

9.2 Interlude: NTRU Encrypt

We now have a look at the textbook NTRU Encrypt scheme, as originally introduced by Hoffstein et al. [HPS98]. The same blueprint is still used in recent and highly efficient schemes, such as the finalist NTRU and alternate finalist NTRU Prime of NIST's standardization process.

Let n, p, q be positive integers such that p, q are prime and co-prime with each other. Further, let $L_f, L_g, L_\phi, L_m \subseteq \mathbb{Z}[x]/(x^n + 1) = R$. Most commonly, those sets are given by ternary polynomials (i.e., coefficients in $\{-1, 0, 1\}$) of fixed Hamming weight. Additionally, we require L_f to only contain polynomials that are invertible modulo q and modulo p. In other words, there exist polynomials F_q and F_p such that $f \cdot F_q = 1 \in R_q$ and $f \cdot F_p = 1 \in R_p$.

KGen: Sample $f \leftarrow U(L_f)$ and $g \leftarrow U(L_g)$. Return $\mathsf{sk} = (f, F_p)$ and $\mathsf{pk} = h = F_q \cdot g \in R_q$.

Enc: For $m \in L_m$, sample $\phi \leftarrow U(L_\phi)$ and return the ciphertext c, where $c = p\phi h + m \in R_q$.

Dec: Compute $m' = F_p \cdot (f \cdot c \bmod q) \bmod p$.

[14] How much larger depends, among other things, on the splitting behavior of the defining polynomial modulo q. Less splitting allows for smaller q.

Correctness. It yields

$$\begin{aligned} f \cdot c \bmod q &= f(p\phi h + m) \bmod q \\ &= fp\phi F_q g + fm \bmod q \\ &= p\phi g + fm \bmod q, \end{aligned}$$

where all ring elements have very small coefficients and thus there is no "wrap-around" modulo q and the equation is also true over the integers. Now, it follows

$$\begin{aligned} F_p(f \cdot c \bmod q) \bmod p &= F_p(p\phi g + fm) \bmod p \\ &= F_p fm \bmod p = m. \end{aligned}$$

Security. There is no security reduction for NTRU Encrypt. However, we can analyze its security with respect to known attacks. For example, to recover the secret key, one has to solve search NTRU, which is itself an instance of SVP of a rank-2 module lattice.

References

[AG11] Sanjeev Arora and Rong Ge. "New Algorithms for Learning in Presence of Errors". In: *ICALP (1)*. Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 403–415.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *STOC*. ACM, 1996, pp. 99–108.

[Ajt98] Miklós Ajtai. "The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions (Extended Abstract)". In: *STOC*. ACM, 1998, pp. 10–19.

[App+09] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO*. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 595–618.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On the concrete hardness of Learning with Errors". In: *J. Math. Cryptol.* 9.3 (2015), pp. 169–203.

[Ban93] Wojciech Banaszczyk. "New bounds in some transference theorems in the geometry of numbers". In: *Math. Ann.* 296.4 (1993), pp. 625–635. ISSN: 0025-5831. DOI: 10.1007/BF01445125. URL: https://doi.org/10.1007/BF01445125.

[BD20] Zvika Brakerski and Nico Döttling. "Hardness of LWE on General Entropic Distributions". In: *EUROCRYPT (2)*. Vol. 12106. Lecture Notes in Computer Science. Springer, 2020, pp. 551–575.

[Ber+21] Olivier Bernard et al. "Log-S-unit lattices using Explicit Stickelberger Generators to solve Approx Ideal-SVP". In: *IACR Cryptol. ePrint Arch.* (2021), p. 1384.

[BGP22] Katharina Boudgoust, Erell Gachon, and Alice Pellet-Mary. "Some Easy Instances of Ideal-SVP and Implications on the Partial Vandermonde Knapsack Problem". In: *CRYPTO (2)*. Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 480–509.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ITCS*. ACM, 2012, pp. 309–325.

[BN06] Mihir Bellare and Gregory Neven. "Multi-signatures in the plain public-Key model and a general forking lemma". In: *CCS*. ACM, 2006, pp. 390–399.

[Boe+20] Koen de Boer et al. "Random Self-reducibility of Ideal-SVP via Arakelov Random Walks". In: *CRYPTO (2)*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 243–273.

[Bos+16] Joppe W. Bos et al. "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE". In: *CCS*. ACM, 2016, pp. 1006–1018.

[Bou+20] Katharina Boudgoust et al. "Towards Classical Hardness of Module-LWE: The Linear Rank Case". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 289–317.

[Bou21] Katharina Boudgoust. *Theoretical Hardness of Algebraically Structured Learning With Errors*. 2021.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. "Pseudorandom Functions and Lattices". In: *EUROCRYPT*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 719–737.

[BR20] Olivier Bernard and Adeline Roux-Langlois. "Twisted-PHS: Using the Product Formula to Solve Approx-SVP in Ideal Lattices". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 349–380.

[Bra+13] Zvika Brakerski et al. "Classical hardness of learning with errors". In: *STOC*. ACM, 2013, pp. 575–584.

[CDW21] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. "Mildly Short Vectors in Cyclotomic Ideal Lattices in Quantum Polynomial Time". In: *J. ACM* 68.2 (2021), 8:1–8:26.

[Chu+20] Chitchanok Chuengsatiansup et al. "ModFalcon: Compact Signatures Based On Module-NTRU Lattices". In: *AsiaCCS*. ACM, 2020, pp. 853–866.

[Cra+16] Ronald Cramer et al. "Recovering Short Generators of Principal Ideals in Cyclotomic Rings". In: *EUROCRYPT (2)*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 559–585.

[DM13] Nico Döttling and Jörn Müller-Quade. "Lossy Codes and a New Variant of the Learning-With-Errors Problem". In: *EUROCRYPT*. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 18–34.

[Dod+08] Yevgeniy Dodis et al. "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data". In: *SIAM J. Comput.* 38.1 (2008), pp. 97–139.

[Duc+18] Léo Ducas et al. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.1 (2018), pp. 238–268.

[DW21] Léo Ducas and Wessel P. J. van Woerden. "NTRU Fatigue: How Stretched is Overstretched?" In: *ASIACRYPT (4)*. Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 3–32.

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *STOC*. ACM, 2009, pp. 169–178.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. "Public-Key Cryptosystems from Lattice Reduction Problems". In: *CRYPTO*. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 112–131.

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems". In: *CHES*. Vol. 7428. Lecture Notes in Computer Science. Springer, 2012, pp. 530–547.

[Gol+10] Shafi Goldwasser et al. "Robustness of the Learning with Errors Assumption". In: *ICS*. Tsinghua University Press, 2010, pp. 230–240.

[Gol+99] Oded Goldreich et al. "Approximating Shortest Lattice Vectors is not Harder than Approximating Closest Lattice Vectors". In: *Inf. Process. Lett.* 71.2 (1999), pp. 55–61.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *STOC*. ACM, 2008, pp. 197–206.

[HPS08] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. *An introduction to mathematical cryptography*. Vol. 1. Springer, 2008.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: *ANTS*. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 267–288.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra Jr., and László Lovász. "Factoring polynomials with rational coefficients". In: *Math. Ann.* 261.4 (1982), pp. 515–534. ISSN: 0025-5831. DOI: 10.1007/BF01457454. URL: https://doi.org/10.1007/BF01457454.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. "Generalized Compact Knapsacks Are Collision Resistant". In: *ICALP (2)*. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 144–155.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *J. ACM* 60.6 (2013), 43:1–43:35.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 75.3 (2015), pp. 565–599.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption". In: *STOC*. ACM, 2012, pp. 1219–1234.

[LWW20] Hao Lin, Yang Wang, and Mingqiang Wang. "Hardness of Module-LWE and Ring-LWE on General Entropic Distributions". In: *IACR Cryptol. ePrint Arch.* (2020), p. 1238.

[Lyu+08] Vadim Lyubashevsky et al. "SWIFFT: A Modest Proposal for FFT Hashing". In: *FSE*. Vol. 5086. Lecture Notes in Computer Science. Springer, 2008, pp. 54–72.

[Lyu09] Vadim Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures". In: *ASIACRYPT*. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 598–616.

[Lyu12] Vadim Lyubashevsky. "Lattice Signatures without Trapdoors". In: *EUROCRYPT*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 738–755.

[Mic07] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions". In: *Comput. Complex.* 16.4 (2007), pp. 365–411.

[MM11] Daniele Micciancio and Petros Mol. "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions". In: *CRYPTO*. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 465–484.

[MP13] Daniele Micciancio and Chris Peikert. "Hardness of SIS and LWE with Small Parameters". In: *CRYPTO (1)*. Vol. 8042. Lecture Notes in Computer Science. Springer, 2013, pp. 21–39.

[MR07] Daniele Micciancio and Oded Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". In: *SIAM J. Comput.* 37.1 (2007), pp. 267–302.

[Pan+21] Yanbin Pan et al. "On the Ideal Shortest Vector Problem over Random Rational Primes". In: *EUROCRYPT (1)*. Vol. 12696. Lecture Notes in Computer Science. Springer, 2021, pp. 559–583.

[Pei09] Chris Peikert. "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract". In: *STOC*. ACM, 2009, pp. 333–342.

[Pei16a] Chris Peikert. "A Decade of Lattice Cryptography". In: *Found. Trends Theor. Comput. Sci.* 10.4 (2016), pp. 283–424.

[Pei16b] Chris Peikert. "How (Not) to Instantiate Ring-LWE". In: *SCN*. Vol. 9841. Lecture Notes in Computer Science. Springer, 2016, pp. 411–430.

[PHS19] Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. "Approx-SVP in Ideal Lattices with Pre-processing". In: *EUROCRYPT (2)*. Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 685–716.

[PR06] Chris Peikert and Alon Rosen. "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices". In: *TCC*. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 145–166.

[PS21] Alice Pellet-Mary and Damien Stehlé. "On the Hardness of the NTRU Problem". In: *ASIACRYPT (1)*. Vol. 13090. Lecture Notes in Computer Science. Springer, 2021, pp. 3–35.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. ACM, 2005, pp. 84–93.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (2009), 34:1–34:40.

[RSW18] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. "On the Ring-LWE and Polynomial-LWE Problems". In: *EUROCRYPT (1)*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 146–173.

[Sch87] Claus-Peter Schnorr. "A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms". In: *Theor. Comput. Sci.* 53 (1987), pp. 201–224.

[SD16] Noah Stephens-Davidowitz. *Dimension-preserving reductions between lattice problems.* http://noahsd.com/latticeproblems.pdf, last accessed on 08.07.2021. 2016.

[SE94] Claus-Peter Schnorr and M. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Math. Program.* 66 (1994), pp. 181–199.

[SS11] Damien Stehlé and Ron Steinfeld. "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices". In: *EUROCRYPT*. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 27–47.

[Ste+09] Damien Stehlé et al. "Efficient Public Key Encryption Based on Ideal Lattices". In: *ASIACRYPT*. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 617–635.

[Ste14] Ron Steinfeld. "NTRU cryptosystem: Recent developments and emerging mathematical problems in finite polynomial rings". In: *Algebraic Curves and Finite Fields: Cryptography and Other Applications* 16 (2014), p. 179.

[WW19] Yang Wang and Mingqiang Wang. "Module-LWE versus Ring-LWE, Revisited". In: *IACR Cryptol. ePrint Arch.* (2019), p. 930.