& 4

Codes and Lattices in Cryptography Mini-Course **FCT-UNL, Spring 2023**

Notes 3 & 4: Hardness of SIS, LWE, and lattice problems

Lecturer: João Ribeiro

Introduction

In the previous lecture we constructed cryptographic objects whose security is based on the hardness of solving the LWE and SIS problems.

In this lecture we first relate the hardness of solving these problems on average to the worst-case hardness of finding short vectors in a given (point) lattice. Then, we discuss the hardness of some versions of this lattice problem. Finding short vectors in a lattice is conjectured to be hard for quantum computers, with the best quantum algorithms matching the best classical algorithms modulo generic quantum speedups. This makes LWE and SIS a popular basis for post-quantum cryptography.

3.1 Basics of lattices

Before we discuss the hardness of the LWE and SIS problems, we introduce lattices.

Definition 3.1 (Lattice) *We say that a set $\mathcal{L} \subseteq \mathbb{R}^n$ is a* lattice *if there exist vectors $v_1, v_2, \dots, v_m \in \mathbb{R}^n$ such that*

$$\mathcal{L} = \left\{ \sum_{i=1}^{m} z_i v_i : z_1, \dots, z_m \in \mathbb{Z} \right\}.$$

In words, $\mathcal{L}$ is a lattice if it consists of integer *linear combinations of a collection of vectors. To each lattice $\mathcal{L}$ we may associate a (non-unique) basis $B \in \mathbb{R}^{n \times m}$ and write $\mathcal{L}$ as the integer column span of B, i.e.,*

$$\mathcal{L} = \mathcal{L}(B) = \{Bz : z \in \mathbb{Z}^m\}.$$

Given a lattice $\mathcal{L}$, a parameter of interest is the ℓ_p-norm of its shortest nonzero vector, a quantity which we denote by

$$\lambda_1^{(p)}(\mathcal{L}) = \min_{v \in \mathcal{L} \setminus \{0\}} \|v\|_p,$$

where $\|v\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{1/p}$ for $p \geq 1$. When $p = 2$, which is the setting we will focus on throughout this lecture, we abbreviate $\lambda_1^{(2)}(\mathcal{L}) = \lambda_1(\mathcal{L})$.

3.1.1 Fundamental lattice problems

We also introduce two important computational problems on lattices. The first problem is related to determining the distance between the lattice and a target vector t.

Definition 3.2 (Closest Vector Problem) *The* γ-approximate Closest Vector Problem (denoted γ-CVP) *is the decisional promise problem defined as follows. On input a generator matrix* $B \in \mathbb{Z}^{m \times n}$*, a target* $\boldsymbol{t} \in \mathbb{Z}^m$*, and a distance parameter* $k \in \mathbb{Z}^+$*, the goal is to decide between the following two cases when one is guaranteed to hold:*

- *(YES)* $\mathsf{dist}(\mathcal{L}(B), \boldsymbol{t}) \leq k$*,*
- *(NO)* $\mathsf{dist}(\mathcal{L}(B), \alpha\boldsymbol{t}) > \gamma k$ *for any* $\alpha \in \mathbb{Z} \setminus \{0\}$*.*

The second problem is related to determining the length of the shortest lattice vector.

Definition 3.3 (Shortest Vector Problem) *The* γ-approximate Shortest Vector Problem (denoted γ-SVP) *is the decisional promise problem defined as follows. On input a generator matrix* $B \in \mathbb{Z}^{m \times n}$ *and a distance parameter* $k \in \mathbb{Z}^+$*, the goal is to decide between the following two cases when one is guaranteed to hold:*

- *(YES)* $\lambda_1(\mathcal{L}(B)) \leq k$*,*
- *(NO)* $\lambda_1(\mathcal{L}(B)) > \gamma k$*.*

Note that SVP is a special case of CVP with target vector $t = 0$. Therefore, solving γ-CVP is at least as hard as solving γ-SVP. Moreover, γ-CVP and γ-SVP become easier as the approximation factor γ increases (and make no sense when $\gamma < 1$).

3.2 Worst-case to average-case reductions

Like most of the computational problems underlying the security of cryptographic protocols, the LWE and SIS problems are *average-case* problems. In both cases, the adversary receives as input a random instance. For example, in the SIS case they receive a uniformly random matrix $A \leftarrow \mathbb{Z}_q^{n \times m}$. Therefore, to "break" SIS, it suffices to design an algorithm that is able to find a short nonzero vector $z \in \ker(A)$ for a *small fraction* of matrices A.

In contrast, complexity theory, which studies the hardness of computational problems, focuses mostly on *worst-case* problems. Examples of worst-case problems includes the γ-CVP and γ-SVP problems defined above. In this case, the goal is to design an algorithm that solves *every* instance of the problem with good probability. The complexity of worst-case problems is much better understood than that of average-case problems. Therefore, a natural approach towards understanding the complexity of a given average-case problem is to come up with an efficient *reduction* from a well-studied worst-case problem, transforming every instance of the worst-case problem into an average instance of the average-case problem. If such a reduction exists, then we know that solving the given average-case problem is at least as hard as the worst-case problem, and hardness results that apply to the worst-case problem also apply to the average-case problem.

For a long time, an important open problem in cryptography was to base the security of a cryptographic protocol on the *worst-case* hardness of a well-studied computational problem. This was finally resolved by already mentioned work of Ajtai [Ajt04], who constructed one-way functions based on the hardness of the SIS problem and also proved a version of the following result connecting the average-case hardness of SIS to the worst-case hardness of γ-SVP.

Theorem 3.1 ([Ajt04] and follow-up work, informal, as stated in [Pei15]) *For any $m = \mathsf{poly}(n)$, any $\beta > 0$, and any sufficiently large $q \geq \beta \cdot \mathsf{poly}(n)$, solving $\mathsf{SIS}_{n,m,q,\beta}$ is at least as hard as solving γ-SVP on worst-case n-dimensional lattices with high probability for some approximation factor $\gamma = \beta \cdot \mathsf{poly}(n)$.*

Regev [Reg09] showed a similar connection between average-case Decision-LWE and worst-case γ-SVP. More precisely, Regev holds in the *quantum setting* – if we can solve Decision-LWE efficiently, then there is an efficient quantum algorithm for solving γ-SVP. This result was later shown to hold in the classical setting as well by Peikert [Pei09]. We state this version below.

Theorem 3.2 ([Reg09, Pei09], informal, as stated in [Pei15]) *For any $m = \mathsf{poly}(n)$, any $q \leq 2^{\mathsf{poly}(n)}$, and any discretized gaussian distribution χ with variance $\alpha q \geq 2\sqrt{n}$, solving the $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ problem is at least as hard as solving γ-SVP on worst-case n-dimensional lattices with high probability for some approximation factor $\gamma \approx n/\alpha$.*

Given the above connections between SIS and LWE and γ-SVP, we would like to have some understanding of the worst-case hardness of the latter problem. The remainder of these notes will focus on this.

3.3 Hardness of lattice problems

Both the γ-CVP and γ-SVP problems described above are known to be difficult to solve under standard complexity-theoretic assumptions for even large approximation factors γ (although not as large as those needed for Theorems 3.1 and 3.2).

Perhaps the most well-known notion of computational hardness is NP-*hardness*. We discuss this notion informally. A problem Π is NP-hard if it is at least as hard as any problem in the complexity class NP, in the sense that if any problem in this class is shown to be hard to solve by polynomial-time algorithms, then so is Π. This corresponds to the major (and widely believed) conjecture in complexity theory that $\mathsf{P} \neq \mathsf{NP}$.

The standard way of showing that a new problem Π is NP-hard is to select another problem Π' which is already known to be NP-hard and then design a polynomial-time *reduction* from Π' to Π. A reduction from Π' to Π is an algorithm that takes as input an instance of Π' and outputs an instance of Π with the property that YES (resp. NO) instances of Π' are always mapped to YES (resp. NO) instances of Π.

Sometimes, as will be the case later on in these notes, these reductions use randomness and only guarantee that YES/NO instances of Π' are mapped to YES/NO instances of Π with high probability. This corresponds to the notion of NP-*hardness under randomized reductions*. Establishing such a reduction from Π' to Π implies that if there is a PPT algorithm that solves Π with high probability, then there is also a PPT algorithm that solves Π', which is NP-hard, with high probability. We believe that the latter cannot happen.[1] We refer the interested reader to [AB09] for a much more extensive and careful discussion of these concepts.

The NP-hardness of γ-CVP for arbitrary constant approximation factor γ follows without much effort via a reduction from the *Exact Set Cover* problem, as discussed in [ABSS97]. With more effort it is possible to extend NP-hardness of γ-CVP to much larger approximation factors γ, as captured in the following result due to Dinur, Kindler, Raz, and Safra [DKRS03].

Theorem 3.3 γ-CVP *is* NP-*hard for approximation factor* $\gamma = n^{c/\log\log n}$ *for some constant* $c > 0$.

We skip the proof of Theorem 3.3 and use it to show the following hardness result for γ-SVP.

Theorem 3.4 γ-SVP *is* NP-*hard for any approximation factor* $\gamma \in [1, \sqrt{2})$ *(under randomized reductions).*

Theorem 3.4 implies that there is no efficient algorithm for solving γ-SVP with $\gamma < \sqrt{2}$ provided that $\mathsf{P} \neq \mathsf{NP}$. The approximation factor in the hardness statement of Theorem 3.4 can, for example, be improved to an arbitrarily large constant γ under the same complexity-theoretic assumption, or to $2^{(\log n)^{1-\varepsilon}}$ for any small constant $\varepsilon > 0$ if we upgrade the underlying assumption $\mathsf{P} \neq \mathsf{NP}$ to stronger, but still sensible, complexity-theoretic assumptions [HR12].

The problem of establishing the NP-hardness of γ-SVP has a long history. Van Emde Boas [vEB81] first showed NP-hardness of the (exact) SVP problem in the ℓ_∞ norm and conjectured that SVP in the ℓ_2 norm is NP-hard. This remained open until work of Ajtai [Ajt98], and a line of work has since then obtained significantly stronger hardness results for γ-SVP (see [Ben23] for an excellent account of this progress). We prove Theorem 3.4 by mixing elegant arguments of Khot [Kho05] and Bennett and Peikert [BP22].

A first attempt. In order to prove Theorem 3.4, we must describe a polynomial-time algorithm that takes as input a γ-CVP instance (B, t, k) with $B \in \mathbb{Z}^{n\times m}$, $t \in \mathbb{Z}^n$, and $k \in \mathbb{Z}^+$, and outputs a γ'-SVP instance (B', k'), for an appropriate γ', such that (B, t, k) is a YES instance of γ-CVP if and only if (B', k') is a YES instance of γ'-SVP.

A natural first approach is to consider the augmented basis $B' = [B \mid -t]$ and $k' = k$. Vectors in $\mathcal{L}(B')$ are of the form $Bz - \beta t$ for some $z \in \mathbb{Z}^n$ and integer β. If (B, t, k) is a YES instance of γ-CVP, then (B', k') is also a YES instance of γ'-SVP. In fact, if there is $v = Bz \in \mathcal{L}(B)$ such that $\|v - t\|_2 \leq k$, then $v' = v - t \in \mathcal{L}(B')$, and so $\lambda_1(\mathcal{L}(B')) \leq \|v'\|_2 \leq k$.

[1] In complexity-theoretic language, this corresponds to the widely believed conjecture that $\mathsf{BPP} \neq \mathsf{NP}$.

Unfortunately, we run into issues when (B, t, k) is a NO instance of γ-CVP. In this case we have that $\|v - \beta t\|_2 > \gamma k$ for all $v \in \mathcal{L}(B)$ and all $\beta \in \mathbb{Z} \setminus \{0\}$. Consequently, vectors $v' \in \mathcal{L}(B')$ of the form $v - \beta t$ for $v \in \mathcal{L}(B)$ and $\beta \neq 0$ have large norm. However, we have no guarantees when $\beta = 0$, as it could happen that $\mathcal{L}(B)$ has short vectors, i.e., $\lambda_1(\mathcal{L}(B)) \ll \gamma k$. Therefore, (B', k') may not be a NO instance of γ'-SVP. We need a different approach.

An alternative approach. Khot [Kho05] gets around the problem we mentioned above via an alternative two-step approach. We begin by considering the "intermediate" basis

$$B_{\text{int}} = \begin{pmatrix} B & 0_{n\times m'} & -t \\ 0_{n'\times m} & A & -s \\ 0_m & 0_{m'} & 1 \end{pmatrix} ,$$

where $m', n' = \mathsf{poly}(n)$ and the basis-target vector pair (A, s) satisfies some special properties.

First, there should be many lattice vectors $w \in \mathcal{L}(A)$ close to s. This ensures that if (B, t, k) is a YES instance of γ-CVP, then $\mathcal{L}_{\text{int}} = \mathcal{L}(B_{\text{int}})$ contains many short vectors (call this quantity N_{good}). On the other hand, $\lambda_1(\mathcal{L}(A))$ should be large. This guarantees that if (B, t, k) is a NO instance of γ-CVP, then the short vectors of $\mathcal{L}_{\text{int}}$ are of the form $(v, 0^{n'}, 0)$ with v a short vector of $\mathcal{L}(B)$. Let N_{bad} denote the number of such vectors. If we set parameters correctly, then we can ensure that $N_{\text{good}} \gg N_{\text{bad}}$.

Of course, we are still not done, because there may be several short vectors in $\mathcal{L}_{\text{int}}$ when (B, t, k) is a NO instance of γ-CVP (at most N_{bad}). To get our final SVP instance, we randomly sparsify $\mathcal{L}_{\text{int}}$, intuitively keeping each lattice vector $v \in \mathcal{L}_{\text{int}}$ with probability $1/\rho$ with $N_{\text{bad}} \ll \rho \ll N_{\text{good}}$ (care must be taken to ensure the lattice structure is preserved). With high probability, all N_{bad} short vectors are removed in the NO case, but at least one short vector survives in the YES case.

The object (A, s) we need to carry this argument through is a *locally dense lattice*. We discuss this approach more carefully below.

3.3.1 Locally dense lattices

We begin by defining locally dense lattices formally.

Definition 3.4 (Locally dense lattice) *Fix a real number $\alpha \in (0, 1)$ and positive integers d, N, m, n. An (α, ℓ, N, n, m)-*locally dense lattice *is specified by a basis $A \in \mathbb{Z}^{n\times m}$ and a target vector $s \in \mathbb{Z}^n$ with the following properties:*

- ***(Large minimum distance):*** *$\lambda_1(\mathcal{L}(A)) \geq \ell$.*
- ***(Many lattice vectors close to s):*** *$|\{w \in \mathcal{L}(A) : \|w - s\|_2 \leq \alpha\ell\}| \geq N$.*

3.3.1.1 Locally dense lattices from error-correcting codes

We construct locally dense lattices with appropriate parameters via a connection to linear codes.

Definition 3.5 (Linear code) *An $[n, r, d]_q$-code $\mathcal{C}$ is a vector subspace of $\mathbb{F}_q^n$ of dimension r which also satisfies*

$$\min_{c \in \mathcal{C} \setminus \{0\}} \|c\|_0 \geq d,$$

where $\|c\|_0 = |\{i : c_i \neq 0\}|$ is the 0-norm (or Hamming weight) of c. We call d the minimum distance *of $\mathcal{C}$.*[2]

We may represent $\mathcal{C}$ as the column span of a generator matrix $G \in \mathbb{F}_q^{n \times r}$*, i.e., we can write*

$$\mathcal{C} = \mathcal{C}(G) = \{Gz : z \in \mathbb{F}_q^r\}.$$

Reed-Solomon codes are some of the most versatile linear codes. They offer an optimal tradeoff between dimension and minimum distance, but require alphabet size $q \geq n$.

Definition 3.6 (Reed-Solomon codes) *Fix a prime power q. The* length-q Reed-Solomon code with codimension h*, denoted by $\mathsf{RS}_{q,h}$, is defined as*

$$\mathsf{RS}_{q,r} = \{(f(\eta))_{\eta \in \mathbb{F}_q} : f \in \mathbb{F}_q[x], \deg f < q - h\}.$$

In words, a Reed-Solomon code consists of the evaluation vectors of all polynomials over $\mathbb{F}_q$ of bounded degree.[3] The following lemma captures basic properties of Reed-Solomon codes.

Lemma 3.1 *$\mathsf{RS}_{q,h}$ is a $[q, q-h, d]_q$-code with minimum distance $d = h + 1$.*

Proof: We discuss the minimum distance only. To see that $d = h + 1$, note that if some polynomial $f \in \mathbb{F}_q[x]$ of degree less than $q - h$ satisfies $f(\eta) = 0$ for at least $q - h + 1$ choices of $\eta \in \mathbb{F}_q$, then f must be the zero polynomial. ■

The Singleton bound states that any $[n, r, d]_q$-code must have $d \leq n - r + 1$. Therefore, Reed-Solomon codes have optimal minimum distance for any given length q and dimension r – codes with this property are called *Maximum Distance Separable* (MDS). We will see further applications of this property to cryptography later on in this mini-course. Also importantly, note that the generator matrix of the Reed-Solomon code $\mathsf{RS}_{q,h}$ can be constructed in time polynomial in q (it is a Vandermonde matrix). For much more about Reed-Solomon codes and coding theory in general, see the excellent book of Guruswami, Rudra, and Sudan [GRS22].

The following construction of locally dense lattices based on Reed-Solomon codes is due to Bennett and Peikert [BP22]. Fix a prime q, codimension h, and a real number $\alpha \in (0, 1)$.

- Take A to be the basis of the lattice

$$\mathcal{L} = \mathsf{RS}_{q,h} + q\mathbb{Z}^q = \{w \in \mathbb{Z}^q : w \pmod{q} \in \mathsf{RS}_{q,h}\}.$$

[2] If $\mathcal{C}$ has minimum distance at least d, then $\|c - c'\|_0 \geq d$ for any two distinct codewords $c, c' \in \mathcal{C}$. This captures worst-case error-correction properties of $\mathcal{C}$.

[3] Sometimes it is useful to restrict the evaluation set to be a particular subset of $\mathbb{F}_q$.

- Sample the target $s \in \{0,1\}^q$ uniformly at random from $\{v \in \{0,1\}^q : \|v\|_0 \leq \alpha^2 \cdot 2h\}$.

The following two lemmas capture the properties of (A, s) as a locally dense lattice. The first result lower bounds the minimum distance of $\mathcal{L}(A)$.

Lemma 3.2 *When $h \leq q/2$, the lattice $\mathcal{L}(A)$ satisfies $\lambda_1(\mathcal{L}(A)) \geq \sqrt{2h} = \sqrt{2(d-1)}$.*

It is instructive to compare Lemma 3.2 with the minimum distance of the Reed-Solomon code $\mathsf{RS}_{q,h}$, which satisfies $d = h + 1$. The latter easily implies the weaker result that

$$\lambda_1(\mathcal{L}(A)) \geq \sqrt{d} = \sqrt{h+1}.$$

However, for our application it is crucial that $\lambda_1(\mathcal{L}(A))$ is significantly larger than this naive bound.

The second lemma states that our sampling process for the target s successfully chooses with high probability a target with many close lattice vectors.

Lemma 3.3 *Fix $\alpha \in (0, 1)$. Then, with probability at least 0.99 the choice of (A, s) above satisfies*

$$\left|\left\{w \in \mathcal{L}(A) : \|w - s\|_2 \leq \alpha\sqrt{2h}\right\}\right| \geq \frac{1}{100}\binom{q}{\alpha^2 \cdot 2h} q^{-h}.$$

Proof: Call a coset[4] V of $\mathcal{L}(A)$ *bad* if

$$|V \cap \mathcal{B}_q(\alpha^2 \cdot 2h)| < \frac{1}{100}\binom{q}{\alpha^2 \cdot 2h} \cdot q^{-h}.$$

We show that s lands on a bad coset with probability at most 1/100. In fact, we have that

$$\begin{aligned}
\Pr[s \text{ lands on a bad coset}] &= \sum_{V : V \text{ is a bad coset}} \frac{|V \cap \mathcal{B}_q(\alpha^2 \cdot 2h)|}{|\mathcal{B}_q(\alpha^2 \cdot 2h)|} \\
&< \sum_{V : V \text{ is a bad coset}} \frac{1}{100 q^h} \\
&\leq \frac{1}{100}.
\end{aligned}$$

The first inequalities holds by the definition of bad coset, and the final inequality uses the fact that there are at most q^h cosets. ■

Combining Lemmas 3.2 and 3.3 immediately yields the following locally dense lattice construction.

Corollary 3.1 *There exists a PPT algorithm which on input a prime q, a positive integer $h \leq q/2$, and a real number $\alpha \in (0, 1)$ outputs with probability at least 0.99 an (α, ℓ, N, q, m')-locally dense lattice (A, s) with*

$$\ell = \sqrt{2h}, \quad N = \frac{1}{100}\binom{q}{\alpha^2 \cdot 2h} q^{-h}, \quad \textit{and} \quad m' = \mathsf{poly}(q).$$

Remark 3.1 This way of turning linear codes into lattices is a version of *Construction A*. For many more connections between lattices and codes, see the great book of Conway and Sloane [CS13].

[4] A coset of a lattice $\mathcal{L}$ is any set of the form $x + \mathcal{L}$ for some vector x.

3.3.2 An intermediate lattice construction

Suppose that we are given a γ-CVP instance (B, t, k) with $B \in \mathbb{Z}^{n \times m}$, $t \in \mathbb{Z}^n$, and $k \in \mathbb{Z}^+$. Consider the "intermediate lattice" $\mathcal{L}_{\text{int}}$ to be the lattice generated by the basis

$$B_{\text{int}} = \begin{pmatrix} B & 0_{n \times m'} & -t \\ 0_{n' \times m} & A & -s \\ 0_m & 0_{m'} & 1 \end{pmatrix} \in \mathbb{Z}^{(n+n'+1)\times(m+m'+1)} ,$$

where (A, s) is the $(\alpha, \ell = \sqrt{2h}, N, q, m')$-locally dense lattice described in Corollary 3.1 with

$$\alpha \in \left(\frac{1}{\sqrt{2}}, 1\right), \quad h = \left\lceil \frac{1 + (\gamma k)^2}{2} \right\rceil ,$$

and q the smallest prime larger than $10^{10}(2n(1+\gamma k))^{\frac{10}{2\alpha^2 - 1}} = \mathsf{poly}(n)$.[5] The parameters are chosen in this way to guarantee that $\lambda_1(\mathcal{L}(A)) \geq \ell = \sqrt{2h} > \gamma k$ (since $h \leq q/2$), and that

$$N = \frac{1}{100}\binom{q}{\alpha^2 \cdot 2h} q^{-h} \geq 10^5(2n(1+\gamma k))^{(\gamma k)^2}.$$

We will use these properties in our reduction. The bottom row guarantees that B_{int} is full-rank whenever A and B are full-rank.

Lemma 3.4 *Fix $\gamma > 1$. Consider any γ' such that*[6]

$$1 \leq \gamma' \leq \frac{\gamma}{\sqrt{3 + (\alpha\gamma)^2}}.$$

Then, the following two properties hold:

- *If (B, t, k) is a NO instance of γ-*CVP*, then there are at most*

$$N_{\text{bad}} = (2n(1+\gamma k))^{(\gamma k)^2}$$

vectors $v \in \mathcal{L}(B_{\text{int}})$ such that $\|v\|_2 \leq \gamma' k'$.

- *If (B, t, k) is a YES instance of γ-*CVP*, then there are at least*

$$N_{\text{good}} = N \geq 10^5(2n(1+\gamma k))^{(\gamma k)^2} = 10^5 N_{\text{bad}}$$

vectors $v \in \mathcal{L}(B_{\text{int}})$ such that $\|v\|_2 \leq k' = \gamma k/\gamma'$.

[5] By Bertrand's postulate, $q = \mathsf{poly}(n)$, and can be found in time $\mathsf{poly}(n)$ as well.

[6] As we take $\gamma \to \infty$, the upper bound on γ' approaches $1/\alpha$ from below. Since we can take α to be arbitrarily close to $\sqrt{2}$, we may set γ' arbitrarily close to $\sqrt{2}$ provided that γ is chosen to be sufficiently large.

Proof: Choose γ' as above and recall that $k' = \gamma k/\gamma'$. We proceed by cases.

If (B, t, k) is a NO instance of γ-CVP, then for every $x \in \mathbb{Z}^m$ and $\beta \in \mathbb{Z} \setminus \{0\}$ it holds that

$$\|Bx - \beta t\|_2 > \gamma k = \gamma' k'. \tag{3.1}$$

Consider an arbitrary vector $z = (x, y, \beta) \in \mathbb{Z}^{m+m'+1}$. We wish to upper bound the number of bad lattice vectors, i.e., vectors $B_{\text{int}} z$ such that $\|B_{\text{int}} z\|_2 \leq \gamma k$. Since $\|B_{\text{int}} z\|_2 \geq \|Bx - \beta t\|_2$ and recalling Equation (3.1), z must have $\beta = 0$. In turn, this implies that $y = 0^{m'}$, since then $\|B_{\text{int}} z\|_2 \geq \|Ay\|_2 \geq \lambda_1(\mathcal{L}(A)) > \gamma k$. We conclude that z must be of the form $z = (x, 0^{m'}, 0)$ for some $x \in \mathbb{Z}^m$, and so the associated bad lattice vectors v in $\mathcal{L}_{\text{int}}$ are of the form $v = (Bx, 0^{n'}, 0)$ for some $x \in \mathbb{Z}^m$. Therefore, we can bound the number of such vectors, N_{bad}, by the number of integer vectors in $\mathbb{Z}^n$ with ℓ_2 norm at most γk, and so

$$N_{\text{bad}} \leq (1 + 2\gamma k)^{(\gamma k)^2} \binom{n}{(\gamma k)^2} \leq (2n(1 + \gamma k))^{(\gamma k)^2},$$

where we have used the fact that $\binom{a}{b} \leq a^b$.

On the other hand, if (B, t, k) is a YES instance of γ-CVP, then there exists $x \in \mathbb{Z}^m$ such that $\|Bx - t\|_2 \leq k$. For each $y \in \mathbb{Z}^{m'}$ such that $\|Ay - s\|_2 \leq \alpha\ell$, consider the vector $z_y = (x, y, 1)$. Then, our choice of parameters ensures that

$$\|B_{\text{int}} z_y\|_2^2 \leq \|Bx - t\|_2^2 + \|Ay - s\|_2^2 + 1 \leq k^2 + (\alpha\ell)^2 + 1 \leq (\gamma k/\gamma')^2,$$

and so $\|B_{\text{int}} z_y\|_2 \leq \gamma k/\gamma' = k'$. It now suffices to observe that there are at least $N_{\text{good}} = N$ vectors $u = Ay$ such that $\|u - s\|_2 \leq \alpha\ell$, and hence at least N_{good} good lattice vectors $B_{\text{int}} z_y$. ■

3.3.3 Obtaining the final SVP instance

According to Lemma 3.4, our intermediate lattice $\mathcal{L}_{\text{int}}$ has (with high probability) the property that in the YES case it has at least N_{good} short vectors, and in the NO case it has at most N_{bad} short vectors. Moreover, our choice of parameters ensures that $N_{\text{good}} \geq 10^5 N_{\text{bad}}$. We would like to transform $\mathcal{L}_{\text{int}}$ into a final sublattice $\mathcal{L}_{\text{final}}$ in a way that at least one short vector of $\mathcal{L}_{\text{int}}$ is preserved in the YES case, but all short vectors are removed in the NO case. This would ensure that $\mathcal{L}_{\text{final}}$ is an appropriate instance of γ'-SVP in each case.

We accomplish this by "randomly sparsifying" the intermediate lattice $\mathcal{L}_{\text{int}}$, so that the properties mentioned above are satisfied with high probability. This is done by adding an appropriate random constraint to $\mathcal{L}_{\text{int}}$. More precisely, let ρ be the smallest prime larger than $100N_{\text{bad}}$. Sample a vector $w \in \mathbb{Z}^{n+n'}$ by sampling each entry independently and uniformly at random from $\{0, \ldots, \rho - 1\}$. Then, we define B_{final} to be the (integral) basis of the sublattice $\mathcal{L}_{\text{final}} = \mathcal{L}(B_{\text{final}}) \subseteq \mathcal{L}_{\text{int}}$ defined as

$$\mathcal{L}_{\text{final}} = \{v \in \mathcal{L}_{\text{int}} : \langle v, w \rangle = 0 \pmod{\rho}\}.$$

Theorem 3.5 *With probability at least* 0.9 *over the choice of random vector* w *above it holds that if* (B, t, k) *is a YES (resp. NO) instance of* γ-CVP, *then* (B_{final}, k') *is a YES (resp. NO) instance of* γ'-SVP.

Proof: Note that the probability that each nonzero vector $v \in \mathcal{L}_{\text{int}}$ is also in $\mathcal{L}_{\text{final}}$ is exactly $1/\rho$. Let X_v denote the indicator random variable of the event "$v \in \mathcal{L}_{\text{final}}$".

Suppose that (B, t, k) is a NO instance of γ-CVP. Then, Lemma 3.4 guarantees that with probability 0.99 there exist at most N_{bad} nonzero vectors $v \in \mathcal{L}_{\text{int}}$ such that $\|v\|_2 \leq \gamma' k'$. By the union bound[7], the probability that $X_v = 1$ for at least one of these vectors is at most $N_{\text{bad}}/\rho \leq 0.01$, where the inequality follows from our choice of ρ. Combining both events, we conclude that (B_{final}, k') is a NO instance of γ'-SVP with probability at least 0.9.

Now, suppose that (B, t, k) is a YES instance of γ-CVP. Then, Lemma 3.4 guarantees that with probability 0.99 there exist at least $N_{\text{good}} \geq 10^5 N_{\text{bad}}$ vectors $v \in \mathcal{L}_{\text{int}}$ such that $\|v\|_2 \leq k'$. We claim that the probability that $X_v = 0$ holds simultaneously for all such v is at most $\rho/N_{\text{good}} \leq 0.01$. To see this, first note that any two good nonzero vectors $v, v' \in \mathcal{L}_{\text{int}}$ are linearly independent over $\mathbb{R}$, and so X_v and $X_{v'}$ are independent random variables. Let $X = \sum_{\text{good } v} X_v$. Then, we have that $\mathrm{E}[X] = N_{\text{good}}/\rho$ by linearity of expectation and $\mathrm{Var}[X] = \sum_{\text{good } v} \mathrm{Var}[X_v]$ by the pairwise independence of the X_v's, and so

$$\Pr[X = 0] \leq \Pr[|X - \mathrm{E}[X]| \geq \mathrm{E}[X]] \leq \frac{\mathrm{Var}[X]}{\mathrm{E}[X]^2} = \frac{\rho^2 \sum_{\text{good } v} \mathrm{Var}[X_v]}{N_{\text{good}}^2} \leq \frac{\rho^2 \cdot N_{\text{good}}/\rho}{N_{\text{good}}^2} = \rho/N_{\text{good}}.$$

The second inequality is an application of Chebyshev's inequality. Combining both events, we conclude that (B_{final}, k') is a YES instance of γ'-SVP with probability at least 0.9. ■

Amplifying the approximation factor. Our argument above establishes NP-hardness of γ-SVP for approximation factor $\gamma \approx \sqrt{2}$. How can we obtain hardness for larger approximation factors? A natural approach is to repeatedly tensor the final lattice, i.e., take $\mathcal{L} = \mathcal{L}(B \otimes B)$, and hope that the length of the shortest vector squares with each tensoring operation. Although this is not true in general, it is known that Khot's SVP instances (with some fairly minor departures from our presentation here) do have this property [HR12].

Other ℓ_p-norms. Our argument above focused on the ℓ_2 norm, but it actually works for every ℓ_p norm. In more generality, several results about γ-SVP in the ℓ_2 norm can be ported to arbitrary ℓ_p norms through norm embeddings [RR06].

3.4 Notes and additional reading

The recent survey by Bennett [Ben23] on the complexity of SVP is an excellent source of known results and open problems.

[7] The union bound states that $\Pr[E_1 \vee E_2] \leq \Pr[E_1] + \Pr[E_2]$ for any two events E_1 and E_2.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABSS97] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[Ajt98] Miklós Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *Thirtieth Annual ACM Symposium on the Theory of Computing, STOC 1998*, pages 10–19. ACM, 1998.

[Ajt04] Miklós Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 3:1–32, 2004. Preliminary version in STOC 1996.

[Ben23] Huck Bennett. The complexity of the shortest vector problem. *SIGACT News*, 54(1):37–61, mar 2023.

[BP22] Huck Bennett and Chris Peikert. Hardness of the (approximate) shortest vector problem: A simple proof via Reed-Solomon codes, 2022. https://arxiv.org/abs/2202.07736.

[CS13] John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290. Springer Science & Business Media, 2013.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is np-hard. *Comb.*, 23(2):205–243, 2003. Preliminary version in FOCS 1998.

[GRS22] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. 2022. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book.

[HR12] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory Comput.*, 8(1):513–531, 2012. Preliminary version in STOC 2007.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, sep 2005. Preliminary version in FOCS 2004.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 333–342, New York, NY, USA, 2009. Association for Computing Machinery.

[Pei15] Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Paper 2015/939, 2015. https://eprint.iacr.org/2015/939.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009. Preliminary version in STOC 2005.

[RR06] Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *38th Annual ACM Symposium on Theory of Computing, STOC 2006*, pages 447–456. ACM, 2006.

[vEB81] Peter van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report, 1981. Available at `https://staff.fnwi.uva.nl/p.vanemdeboas/vectors/mi8104c.html`.