Worst-Case Hardness

1 Introduction

In the previous course, we have introduced the Short Integer Solution (SIS) problem and showed that it give rises to some basic cryptographic primitives such as collision resistant Hash functions. We have also shown that the SIS can be viewed as a lattice problem, more specifically as the problem of finding a short vector for a certain distribution of random lattices.

From a security perspective, this doesn't necessarily means much: we can solve SIS by solving approx-SVP in a certain lattice; while we may be convinced that lattice problem are hard, this does not imply that SIS is hard. The reduction is in the wrong direction.

Consider the dull example for computing the GCD of x and y. One could devise the following algorithm: factor x and y into prime factors, and compute then take the product of the common factors. This doesn't mean that GCD is as hard as factoring, only that GCD is at most as hard as factoring. And indeed, GCD can be computed much faster than factoring.

What we would want to show is that solving SIS implies solving SVP. And by this, we shouldn't restrict SVP to a specific class of lattices, because such a specific class might in fact hide structure that makes the problem easy. That is, we want to reduce *any* instance of SVP to SIS. In other term, we want to consider SVP as a *worst-case* problem (universal quantification over the instance), and reduce it to average-case instances of SIS.

Not only this dismisses the issue of SIS lattices being easy as a whole, it also dismisses the possibility that a significant fraction of SIS instances could be easy; it would be a serious issue for cryptography if random keys had a non negligible probability to be easy to break.

The plausibility of weak keys has been a serious issue in cryptography; for example the RSA cryptosystem based on factoring in fact requires "strong primes", rather than primes: if a prime factor p or q of the key $N = pq$ has the property that $p - 1$ has many small factors, the it is easier to break.

Parameters In the following, we leave the parameters β, q, m of $\mathrm{SIS}_{n,m,q,\beta}$ arbitrary until the very end of the proof. But we can already think of them as being polynomial in n, and this n will also be the dimension of the worst-case lattice we want to solve SVP in.

2 A Worst-case to Average-case reduction for SIS

We are going to achieve this iteratively: first we will show that given a long basis and an oracle for SIS, then we can find a vector in L that is shorter than the vector in the long basis; this is the core of the reduction. Then we will argue than this short vector can be integrated in the basis to make it a bit better, and therefore will iteratively obtain a good basis.

2.1 Finding a Shorter Vector

The core of Ajtai reduction is given as Algorithm 1. In brief, given a lattice it generates random points $\mathbf{v}_i$ in the lattice $\frac{1}{q}L$; for q large enough, these points are relatively short compared to the current basis $\mathbf{B}$ of L. Yet, those points $\mathbf{v}_i$ are not themselves in L, so we could instead try to find

a small linear combinations of them $\mathbf{s} = \sum x_i \mathbf{v}_i$ that is in L. Because $\frac{1}{q}L \simeq \mathbb{Z}_q^n$, we can rephrase the constraint $\mathbf{s} \in L$ as a system of linear equation modulo q, namely, an SIS instances.

Algorithm 1: Ajtai Worst-Case to average Reduction core

Input : A basis $\mathbf{B}$ of a full-rank lattice L of rank n.
An Oracle $\mathcal{A}$ for $\mathrm{SIS}_{n,m,q,\beta}$ where q is odd.
A sampling oracle for a distribution $\mathcal{D}$ over $\mathbb{R}^n$.
Output: A vector $\mathbf{v} \in L$

for $i \in \{1, \dots m\}$ **do**
  $\mathbf{t}_i \leftarrow \mathcal{D}$ // $\in \mathbb{R}^n$
  $\mathbf{v}_i \leftarrow \mathsf{NearestPlane}(\frac{1}{q}\mathbf{B}, \mathbf{t}_i)$ // $\in \frac{1}{q}L$
  $\mathbf{a}_i = (q\mathbf{B}^{-1} \cdot \mathbf{v}_i) \bmod q$ // $\in \mathbb{Z}_q^n$
end
$\mathbf{x} \leftarrow \mathcal{A}([\mathbf{a}_1, \dots, \mathbf{a}_m])$ // $\in \mathbb{Z}^m \setminus \{\mathbf{0}\}$
$\mathbf{y} \leftarrow 2^{-k} \cdot \mathbf{x}$ where $k = \max\{k \geq 0 : 2^k | \gcd(\mathbf{x})\}$ // $\in \mathbb{Z}^m \setminus \{\mathbf{0}\}$
$\mathbf{s} \leftarrow [\mathbf{v}_1, \dots, \mathbf{v}_m] \cdot \mathbf{x}$ // $\in L$
return s

Let us also assume that $k = 0$ for this discussion, that is $\mathbf{y} = \mathbf{x}$. Let's assume that the distribution $\mathcal{D}$ is supported by a ball of radius r, that is $\|\mathbf{t}_i\| \leq r$. Then $\|\mathbf{v}_i\| \leq r + \frac{1}{q}\mu(\mathcal{P}(\mathbf{B}^\star))$, and we can conclude that $\|\mathbf{s}\| \leq m\beta \cdot (r + \frac{1}{q}\mu(\mathcal{P}(\mathbf{B}^\star)))$. If q is large enough and β small enough, then we have found a vector significantly shorter than the ones of the current basis $\mathbf{B}$. There are two difficulty to work out to prove that we indeed get an good solution to approx-SVP:

1. While the matrix $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_m] \in \mathbb{Z}_q^{n \times m}$ has the correct type, the definition of the oracle $\mathcal{A}$ only guarentees its output if $\mathbf{A}$ has the uniform distribution.

2. If it does work, we have the guarentee that $\mathbf{x}$ is short and non-zero, and therefore that $\mathbf{s} = \mathbf{A}\mathbf{x}$ is short, but it might still be zero. Such an accident might seem unlikely, but the oracle $\mathcal{A}$ might be whimsical. In particular, if that adversary knows L and $\mathbf{v}_i$ entirely, he can certainly force such an accident. We should make sure that there remains enough uncertainty about the $\mathbf{v}_i$ even given the $\mathbf{a}_i$ to limit these accidents.

Both issues will be resolved by a notion called smoothness. The existence of smooth function that can be sampled from efficiently is deffered to the last section of this lecture.

Definition 1 (Smoothness) *A distribution $\mathcal{D}$ over $\mathbb{R}^n$ is called ε-smooth with respect to a full rank lattice $L \subset R^n$ if $\mathcal{D}$ mod L is at statistical distance at most ε from the uniform distribution over the torus $\mathbb{R}^n/L$:*

$$\Delta(\mathcal{D} \bmod L, \mathcal{U}(\mathbb{R}^n/L)) \leq \varepsilon.$$

We recall that the statistical distance between two distributions $\mathcal{D}, \mathcal{E}$ is given as:

$$\Delta(\mathcal{D}, \mathcal{E}) := \frac{1}{2}\int_{\mathbf{x}} |\mathcal{D}(\mathbf{x}) - \mathcal{E}(\mathbf{x})| d\mathbf{x}.$$

where $\mathcal{D}(\mathbf{x})$ denotes the probability density of $\mathcal{D}$ at $\mathbf{x}$. We say of two distribution that they are ε-close if their statistical distance is less than ε. A key property of the statistical distance is the so-called data-procesisng inequality: for any function f, applying f can only decrease the statistical distance: $\Delta(f(\mathcal{D}), f(\mathcal{E})) \leq \Delta(\mathcal{D}, \mathcal{E})$.

At last, it is also useful to relax the support the constraint on the support of $\mathcal{D}$ as follow.

DEFINITION 2 *A distribution $\mathcal{D}$ over $\mathbb{R}^n$ is called (ε, r)-bounded if $\mathbb{P}_{\mathbf{x}\leftarrow\mathcal{D}}[\|\mathbf{x}\| > r] \leq \varepsilon$.*

We are now ready to state the correctness of the algorithm. Note below that we make a stronger statement that the output being non-zero with high probability: we in fact claim it to be close uniform $\mod 2L$, which will help us later on construct better bases rather than just a short vectors. But for now, we note that it is therefore non-zero except with probability close to 2^{-n}.

THEOREM 3 *Let L be a full-rank lattice of $\mathbb{R}^n$. Let $\mathcal{A}$ be an $\mathrm{SIS}_{n,m,q,\beta}$ oracle succeeding with probability at least p. Let $\mathcal{D}$ with a sampling oracle, that is (r, ε_0)-bounded, ε_1-smooth with respect to $2L$.*

Then, with probability at least $p - m(\varepsilon_0 + \varepsilon_1)$ Algorithm 1 returns a vector $\mathbf{s}$ of L of length at most $m\beta \cdot (r + \frac{1}{q}\mu(\mathcal{P}(\mathbf{B}^\star)))$. Furthermore, $\mathbf{s} \bmod 2L$ is ε_1-close to uniform in $L/2L$.

PROOF: We first claim that the $\mathbf{v}_i$ are ε_1 close to uniform in $\frac{1}{q}L/2L$. Indeed, if $\mathbf{t}$ is uniform in $\mathbb{R}^n/L$, then $\mathbf{v} = \mathsf{NearestPlane}(\frac{1}{q}\mathbf{B}, \mathbf{t})$ is uniform in $\frac{1}{q}L/2L$ by a tiling argument similar to that of Exercise 1 Sheet 1. By the smoothness assumption, $\mathbf{t}_i$ are ε_1-close to uniform in $\mathbb{R}^n/2L$, and we conclude our claim by the data-processing inequality.

In particular, the $\mathbf{v}_i$ are ε_1 close to uniform in $\frac{1}{q}L/L$, which is equivalent to $\mathbf{a}_i$ being uniform in $\mathbb{Z}_q^n$.

Hence, the matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ fed to the oracle is $m\varepsilon$-close to uniform in $\mathbb{Z}_q^{n\times m}$. Therefore, once again by the data-processing inequality, the output $\mathbf{x}$ of $\mathcal{A}(\mathbf{A})$ is a non-zero solution to $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ of norm at most β with probability at least $p - m\varepsilon_1$. Finally, we note that 2 being co-prime to q, it also hold that $\mathbf{A}\mathbf{y} = \mathbf{0} \bmod q$, and that $\|\mathbf{y}\| \leq \|\mathbf{x}\| \leq \beta$. Finally, all the $\mathbf{t}_i$ have norm at most r with probability $m\varepsilon_0$ and we conclude on the first statement of the theorem.

For the second half of the statement, note that by construction $\mathbf{y} \neq 0 \bmod 2$, and let i be an index such that $y_i \neq 0 \bmod 2$. Recall that $\mathbf{v}_i$ was ε_1 close to uniform in $\frac{1}{q}L/2L$, yet we only revealed $\mathbf{v}_i \bmod L$ to the adversary: $(\mathbf{v}_i - (\mathbf{v}_i \bmod L)) \bmod 2L$ is ε_1-close to uniform even conditionned on the knowledge of $(\mathbf{v}_i \bmod L)$ by Chinese-Remainder Theorem. This suffice to make $\mathbf{V}\mathbf{y}$ ε_1-close to uniform in $L/2L$. □

2.2 Building a better basis

The reduction above allows to construct vectors that are shorter than the current basis. Furthermore, we have the guarentee that those vectors are uniform in $L/2L$, hence after say $2n$ calls, we can be almost sure to have $\mathbf{s}_i$ that generate $L/2L$. Indeed, $L/2L \simeq \mathbb{Z}_2^n$, and $2n$ uniform samples in $\mathbb{Z}_2^n$ generate it all with probability $1 - O(2^{-n})$. We can therefore extract a set of n such $\mathbf{s}_i$ that generates it, giving a matrix $\mathbf{S}$ of linearly independent vectors: if they were linearly dependent, they would also be dependent modulo $2L$ and hence not generating $L/2L$.

However, $\mathbf{S}$ may generate only a full-rank sublattice of L and not L entirely, this is problematic as we intend to repeat the process iteratively to improve the basis progressively. For this, we propose an algorithm to reconstruct a good basis $\mathbf{C}$ of L out of a bad basis and a set $\mathbf{S}$ of short linearly independant vectors of L. More specifically we can guarentee that $\mu(\mathcal{P}(\mathbf{C}^\star)) \leq \sqrt{n}/2 \cdot \max\|\mathbf{s}_i\|$.

Algorithm 2: ShortBasisReconstruction$(\mathbf{B}, \mathbf{S})$

Input : A basis $\mathbf{B}$ of a full-rank lattice L of rank n. A full-rank set $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ of vectors of L

Output: A basis $\mathbf{C}$ such that $\|\mathbf{c}_i^\star\| \leq \max\|\mathbf{s}_i\|$.

Remove all $\mathbf{0}$ vectors from $\mathbf{S}$
Let $k \in \mathbb{N}$ be such that $\frac{1}{k}\mathbf{s}_1$ is primitive in L
$\mathbf{c}_1 \leftarrow \frac{1}{k}\mathbf{s}_1$
$\mathbf{S}' = \{\pi_{\mathbf{c}_1}^\perp(\mathbf{s}) | \mathbf{s} \in \mathbf{S}\}$
$\mathbf{B}' =$ any basis of $\pi_{\mathbf{c}_1}^\perp(L)$
$\mathbf{C}' =$ ShortBasisReconstruction$(\mathbf{B}', \mathbf{S}')$
return $[\mathbf{c}_1, \pi_{\mathbf{c}_1}^{-1}(\mathbf{c}_1'), \pi_{\mathbf{c}_1}^{-1}(\mathbf{c}_2'), \dots]$

In the above algorithm, it is assume that we know how to realize various subtasks: finding the k making a vector primitive, constructing an (non-necessarily short) basis of $\pi^\perp(L)$, and finally, lifting vectors $\pi_{\mathbf{c}_1}^{-1}(\mathbf{c}_1')$. The last step can simply be done via Babai Nearest Plane lifting, as we have seen in previous lecture. The construction of a basis of $\pi^\perp(L)$ can be done by eliminating linear dependencies in $\pi^\perp(\mathbf{B})$ using the Hermite Normal Form algorithm. Finally, to find the k to make $\mathbf{c}_1$ primitive, one may simply take the gcd of $\mathbf{s}_1$ written in base $\mathbf{B}$: $k := \gcd(\mathbf{B}^{-1}\mathbf{s}_1)$.

Lemma 4 *Algorithm 2 is correct and runs in polynomial time.*

Proof: The key remark is that $k \geq 1$, therefore $\|\mathbf{c}_1\| \leq \|\mathbf{s}_1\|$. One also need to argue that $\mathbf{S}'$ is a full rank set of L', to finally conclude by induction. The details are left to the reader. □

3 L-Smooth Distributions

3.1 Naive Smoothness

In this lecture, we are going to repeatedly re-use a Lemma similar to one Exercise Sheet 1, counting the number of point in a shifted ball. We recall that $\mu(L)$ denotes the covering radius of L, i.e. the smallest radius of a ball that is L-covering: $\mu(L) := \min\{r > 0 \mid r\mathfrak{B} + L \supset \mathrm{Span}_{\mathbb{R}}(L)\}$.

This formulation requires proving the existence of an L-tiling T with outer radius $\mu(T) = \mu(L)$; such a tiling is given by the Voronoi cell (see Homework).

Lemma 5 *For any full rank lattice L of rank n and $r > 2\mu(L)$ and any shift $\mathbf{t} \in \mathbb{R}^n$, it holds that*

$$\frac{(r - 2\mu(L))^n}{\det(L)} \leq \frac{|L \cap (\mathbf{t} + r\mathcal{B})|}{\mathrm{vol}(\mathcal{B})} \leq \frac{(r + 2\mu(L))^n}{\det(L)}.$$

Furthermore, if $\mathbf{t} = \mathbf{0}$, the quantity $2\mu(L)$ can be replaced by $\mu(L)$.

Proof: Let T be a tiling set of L that is contained in $\mu(L)\mathcal{B}$. Without loss of generality, we can assume that $\mathbf{t} \in T$, as shifting $\mathbf{t}$ by a lattice point $\mathbf{v} \in L$ doesn't affect the size of $|L \cap (\mathbf{t} + r\mathcal{B})|$:

$$|L \cap (\mathbf{t} + \mathbf{v} + r\mathcal{B})| = |(L - \mathbf{v}) \cap (\mathbf{t} + r\mathcal{B})| = |L \cap (\mathbf{t} + r\mathcal{B})|.$$

In particular, $\|\mathbf{t}\| \leq \mu(L)$. Consider the set $(L \cap (r\mathcal{B} + \mathbf{t})) + T$, and note that it is included the ball of radius $r + 2\mu$. This gives $\mathrm{vol}((L \cap (r\mathcal{B} + \mathbf{t})) + T) \leq \mathrm{vol}((r + 2\mu(L))\mathcal{B}) = (r + 2\mu(L))^n \mathrm{vol}(\mathcal{B})$.

But since T is tiling, and any tiling set has volume $\det(L)$, the left hand side can be written as $\sum_{\mathbf{x}\in L\cap(r\mathcal{B}+\mathbf{t})} \mathrm{vol}(T) = |L\cap(r\mathcal{B}+\mathbf{t})|\det(L)$. Thus $|L\cap(r\mathcal{B}+\mathbf{t})|\det(L) \leq (r+2\mu(L))^n \mathrm{vol}(\mathcal{B})$ and the upper bound follows.

For the lower bound, one considers the inclusion $(r-2\mu(L))\mathcal{B} \subseteq (L\cap(r\mathcal{B}-\mathbf{t}))+T$ and concludes by a similar reasoning. And the improved bounds for $\mathbf{t}=0$ is also a straightforward adaptation of the above. □

Lemma 6 *If $\mathbf{t} \leftarrow \mathcal{U}(r\mathfrak{B})$, then distribution $\mathcal{D}$ of $\mathbf{t}$ mod L is $O(\varepsilon)$-close to the uniform distribution over the torus $\mathbb{R}^n/L$ for $\varepsilon = \frac{2n\cdot\mu(L)}{r}$.*

Proof: Let $\mathbf{t}\in\mathbb{R}^n/L$ be a point on the torus; the probability density function of $\mathcal{D}(\mathbf{t})$ is proportional to the number of points of the lattice coset $\mathbf{t}+L$ that fall in the ball $r\mathfrak{B}$, which we call $N(\mathbf{t}) = |(\mathbf{t}+L)\cap r\mathcal{B}|$. This rewrites as $N(\mathbf{t}) = |L\cap(-\mathbf{t}+r\mathcal{B})|$ and we can apply Lemma 5:

$$\begin{aligned} N(\mathbf{t}) &\in \frac{\mathrm{vol}(\mathfrak{B})}{\det(L)} r^n \cdot [(1-2\mu(L)/r)^n, (1+2\mu(L)/r)^n] \\ &\in \frac{\mathrm{vol}(\mathfrak{B})}{\det(L)} r^n \cdot [1-O(\varepsilon), 1+O(\varepsilon)]\,. \end{aligned}$$

Hence $\mathcal{D}(\mathbf{t}) = \frac{1}{\det L}\cdot\frac{N(\mathbf{t})}{\mathbb{E}_{\mathbf{t}'}[N(\mathbf{t}')]} \in \frac{1}{\det L}[\frac{1-O(\varepsilon)}{1+O(\varepsilon)}, \frac{1+O(\varepsilon)}{1-O(\varepsilon)}]$. We conclude noting that $\frac{1-O(\varepsilon)}{1+O(\varepsilon)} = 1-O(\varepsilon)$ and $\frac{1+O(\varepsilon)}{1-O(\varepsilon)} = 1+O(\varepsilon)$. □

3.2 Smoothness via Harmonic Analysis

The above lemma for smoothness is quite proof: the distance to uniform decreases as $O(\frac{2n\cdot\mu(L)}{r})$. There exists much stronger results of the sort, namely, that it decreases exponentially fast in the dimension:

$$\varepsilon = O\left(\frac{2n}{r\lambda_1(L^\vee)}\right)^n.$$

We briefly overview the general idea behind such smoothness bound to explain the sudden appearance of the minimal distance of the dual . Rather than uniform distribution over a ball, let us consider a gaussian distribution of parameter σ

$$\rho_\sigma(\mathbf{x}) := \frac{1}{(\sigma\sqrt{2\pi})^n}\exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right).$$

We want to understand how close to uniform is this distribution when taken modulo the lattice. This is done by periodizing this function, that is by considering the L-periodic function:

$$\rho_\sigma(\mathbf{x}+L) := \sum_{\mathbf{v}\in L}\rho_\sigma(\mathbf{x}+\mathbf{v}).$$

We can apply Fourier decomposition on this periodic function. Note that the Fourier transform of ρ_σ over $\mathbb{R}$ is $\rho_{1/2\pi\sigma}$; the Fourier transform of its periodization over $\mathbb{R}^n/L$ is therefore the restriction of $\rho_{1/2\pi\sigma}$ to the dual lattice $L^\vee$ (Poisson Summation Formula):

$$\rho_\sigma(\mathbf{x}+L) = \sum_{\mathbf{y}\in L^\vee}\rho_{1/2\pi\sigma}(\mathbf{y})\cdot\exp(-2\imath\pi\langle\mathbf{x},\mathbf{y}\rangle).$$

The point is that if $1/2\pi\sigma$ is somewhat smaller than $\lambda_1(L^\vee)$, then, all terms but $\mathbf{y} = 0$ are extremely small, while the term for $\mathbf{y} = \mathbf{0}$ correspond to the constant function: the periodization of the Gaussian is very close to constant.

A similar argument can be made for the uniform distribution over a ball, but requires to study the Bessel function, which is quite more technical.

4 Wrap-up

We now have all the ingredient for the full Worst-Case to Average Case Reduction.

We are given an arbitrary basis $\mathbf{B}$ of a lattice L, and an oracle $\mathcal{A}$ for $\mathrm{SIS}_{n,m,q,\beta}$, which for simplicity assume to be successful with probability $p \geq 1/2$. We set $\mathcal{D}$ to be the uniform distribution over a ball of radius $r > 4mn^2\mu(L)$, Which is $(r, 0)$-bounded and ε_1-smooth with respect to both L and $2L$ according to Lemma 6 for $\varepsilon_1 = O(1/nm)$.

We call Algorithm 1 repeatedly to successfully obtain $2n$ vectors, collected in a set $\mathbf{S}$, of length at most $\ell = m\beta(r + \frac{1}{q}\mu(\mathcal{P}(\mathbf{B}^\star)))$, which requires on average $2n/(p - m\varepsilon_1) \leq 4n + o(1)$ many calls by Theorem 3. By the same Theorem, these $2n$ vectors $\mathbf{S}$ generate L modulo $2L$, and therefore generate a full-rank sublattice of L.

We then construct a basis $\mathbf{C}$ of L such that $\mu(\mathcal{P}(\mathbf{C}^\star)) \leq \sqrt{n}\ell$ using algorithm 2, and note that

$$\frac{\mu(\mathcal{P}(\mathbf{C}^\star)))}{\mu(\mathcal{P}(\mathbf{B}^\star)))} \leq \frac{m\beta r\sqrt{n}}{\mu(\mathcal{P}(\mathbf{B}^\star)))} + \frac{m\beta\sqrt{n}}{q}.$$

The second term can be made very small by the very choice of the $\mathrm{SIS}_{n,m,q,\beta}$ parameters, while the first term is small as long as $\mu(\mathcal{P}(\mathbf{B}^\star))$ is large.

In other term, the new basis $\mathbf{C}$ has an an outer radius $\mu(\mathcal{P}(\mathbf{C}^\star))$ significantly shorter (say by a factor 2) than the old basis $\mathbf{B}$, unless the $\mu(\mathcal{P}(\mathbf{B}^\star))$ was already short, say shorter than $2m\beta r\sqrt{n} = 8m^2n^{5/2}\beta \cdot \mu(L)$.

That is, we can repeat the process until we obtain a basis that is only polynomially worse in length that the covering radius of L, namely, the final basis $\mathbf{F}$ satisfies

$$\mu(\mathcal{P}(\mathbf{F}^\star)) \leq 4m^2n^{5/2}\beta \cdot \mu(L).$$

The number of repetition to reach such a basis is also polynomial in the size of the input basis, as the process divides $\mu(\mathcal{P}(\mathbf{B}^\star))$ at each iteration. Instantiating the parameters, we conclude.

THEOREM 7 *If there exists a polynomial time algorithm solving* $\mathrm{SIS}_{n,m,q,\beta}$ *with probability* $p > 1/2$ *for any* $m, q, \beta = n^{O(1)}$ *such that* $m\beta\sqrt{n}/q = o(1)$*, then there exists a polynomial time algorithm that find a short non-zero vectors in any n-dimensional lattice L of length at most* $\mu(L) \cdot n^{O(1)}$.