COS 533: Advanced Cryptography — Princeton University
Lecture 19 (November 27, 2017)
Lecturer: Mark Zhandry — Scribe: Fermi Ma

Notes for Lecture 19

1 Worst Case to Average Case Reduction

Why should we believe particular instances of problems are hard? For some lattices, the shortest vector problem is not hard. For building crypto, we want a justification for why particular instances of problems are hard.

Theorem: Suppose we have an oracle that solves the short integer solution problem (on average) (which we denote as $SIS_{n,m,q,\beta}$). Then we can solve a modification of the $SIVP_\gamma$ problem (in the worst case) if $q = O(n^2), B = O(\sqrt{m}), m = \Omega(n \log q)$.

What this says is that SIS is the right distribution to consider. If I believe that SIVP is hard in the worst-case, then I can't produce an SIS oracle without getting a contradiction.

1.1 Definitions

Successive minima of lattices. $\lambda_1(L)$ is the length of the shortest vector in $L \setminus \{0\}$. We define

$$\lambda_i(L) = \min_\ell \{\ell : L \text{ contains } i \text{ linearly independent vectors of length } \leq \ell\}$$

The shortest vector problem is

$$SVP_\gamma : \text{ find vector v of length } \leq \lambda_i(L)$$

The shortest independent vectors problem is

$$SIVP_\gamma : \text{ find basis } B = \{b_1, \ldots, b_n\} \text{ such that } |b|_i \leq \gamma \lambda_n(L) \forall i$$

You can relate these two problems but they're somewhat incomparable.

The shortest integer solution problem $SIS_{n,m,q,\beta}$ is: given $A \leftarrow \mathbb{Z}_q^{n \times m}$ randomly sampled, find $x \in \mathbb{Z}^m$ such that $A \cdot x = 0 \mod q$ and $|x| \leq \beta$.

1.2 Reduction

Proof of theorem. Our algorithm works as follows. Given B for lattice L, choose $x_1, \ldots, x_m \in \mathbb{Z}^n$ according to discrete Gaussian of width σ. Let $y_i = x_i \mod L(B)$. Essentially, y_i is the result of subtracting out a lattice point from x_i so that we end up in the fundamental parallelpiped. It's not hard to figure out that $y_i = B(B^{-1}x_i \mod 1)$, since $B^{-1}x \mod 1$ drops the integer component, and then multiplying by B gets us inside the parallelpiped. It's easy to check that when $y_i = B(B^{-1}x_i \mod 1)$, the difference between y_i and x_i is a lattice point.

We claim that the y_i's are basically uniformly distributed in the fundamental parallelpiped. This is intuitively clear if we pick the Discrete Gaussian to have high enough width (with respect to the size of the parallelpiped). We won't prove this, but we have a lemma that formalizes this. If $\sigma \geq \lambda_n(L)\sqrt{\log n + \log(1/\epsilon)}$, then y_i is ϵ close (distributions are ϵ close) to random.

Note that the fundamental parallelpiped has the same volume regardless of the choice of basis vectors. So even though the above lemma doesn't depend on the length of the basis vectors, it's still correct since there is a one to one correspondence between points in any fundamental parallelpiped.

Now consider the bigger lattice $\frac{1}{q}L(B)$. We're going to let $a_i = \lceil (\frac{1}{q}B)^{-1}y \rfloor$, which is the rounding of y so that it is uniform in $\mathbb{Z}_q^n$. Then we compute $z_i = (\frac{1}{q}B)a_i$ which is uniform among the lattice points in $\frac{1}{q}L(B)$ within the parallelpiped of $L(B)$. Then we let $A = (a_1, \ldots, a_m)$, and we give this to the SIS oracle to get $e = e_1, \ldots, e_n$.

We know that since e is a SIS solution, we have $\sum_i e_i a_i = 0 \mod q$ and that $|e| \leq \beta$. We'll turn this into a short vector.

For now, we say that we output $U = \sum_i e_i(x_i - y_i + z_i)$. This won't be entirely right, but we'll show how to fix it.

First, note that $U \in L(B)$. To see this, we notice that $\sum_i e_i(x_i - y_i) \in L(B)$. This is because we chose y_i so the $x_i - y_i$ differences would be in the lattice, and then we take integer combinations.

Now we just need to show that $\sum_i e_i z_i \in L(B)$. We have $\sum_i e_i a_i = 0 \mod q$, and if we scale down a_i by q, we have $\sum_i e_i(a_i/q) = 0 \mod 1$. So we conclude that $\sum_i e_i(\frac{1}{q}Ba_i) = 0 \mod B$ from multiplying both sides by the basis B. This means that I can subtract a lattice point from $\sum_i e_i z_i$ to get to the origin, meaning that $\sum_i e_i z_i$ itself is a lattice point.

Next claim is that U is short. We have

$$|U| \leq |\sum_i e_i x_i| + |\sum_i p_i(z_i - y_i)|$$

This quantity is approximately

$$\approx \beta\sigma\sqrt{n} + \beta n \max|b_i|/q.$$

The first term is from the Cauchy-Schwarz inequality, and the second term comes from the fact that z_i is a point in the larger lattice, and y_i is just a random point, so the distance is at most the length of a basis vector divided by q.

If we set q to be around n^2, we want to claim that this is actually a short vector. But note that $\max|b_i|$ could be exponentially big, so we don't immediately have our result. Instead, we'll use an iterative approach based on the fact that if I choose q to be bigger than βn, then $\frac{\beta n \max|b_i|}{q}$ is already smaller than the largest vector in the basis. We can pick it so that it's smaller than $1/2$ times the largest vector in the basis, and then swap out the largest vector in the basis for U. Repeating this a polynomial number of times eventually gives a small enough basis, and then we get a short U.

More precisely, we get $\max|b_i| \leq O(n\lambda_n)$, which solves $SIVP_\gamma$. And also $|U| = O(Bn\sqrt{m})$ gives us an SVP_γ solution.

Two issues we may have are 1) U may be in the subspace of all the shorter vectors (i.e. swapping in U for the longest basis vector no longer gives a basis), and 2) U may itself be 0.

It turns out that the randomness in the x_i's will solve both of these issues with the analysis. Note U is composed of e_i, x_i, y_i, z_i terms, but that even with the e_i, y_i, z_i terms fixed, the difference $x_i - y_i$ can vary as a discrete Gaussian. This randomness is enough to give us U vectors that do not have these problems.

2 Cryptanalysis

It turns out that lattices are useful for solving presumably hard problems. We consider the problem of computing a short basis for a lattice given a basis for the lattice.

This is the Lenstra-Lenstra-Lovasz (LLL) Algorithm.

As an illustrating example, imagine two basis vectors that are very close to each other and form a very skinny parallelpiped. We can try subtracting the two vectors. But maybe $b_2 - 2b_1$ is a better short vector than $b_2 - b_1$. Roughly what we try to do is find the optimal integer k such that $b_2 - kb_1$ is as short as possible. To do this, we'll use Gram-Schmidt.

Recall that in Gram-Schimdt, given a basis $B = (b_1, \ldots, b_n)$, we compute a bunch of orthonormal vectors that span the same space (actually we'll just care about orthogonality and not orthonormality). So $\tilde{b_1} = b_1$. Then $\tilde{b_2} = b_2 - \mu_{2,1}\tilde{b_1}$ where $\mu_{2,1} = \frac{\langle b_2, \tilde{b_1}\rangle}{\langle \tilde{b_2}, \tilde{b_1}\rangle}$. Then we get $\tilde{b_3} = b_3 - \mu_{3,1}\tilde{b_1} - \mu_{3,2}\tilde{b_2}$.

However, we can't just do Gram-Schmidt, since we're dividing by stuff and therefore my orthogonal basis will be rational (not lattice vectors). So we'll just do rounding instead of subtracting off exactly $\mu_{2,1}$. So the algorithm is $b_1' = b_1$ and $b_2' = b_2 - \lceil \mu_{2,1} \rfloor b_1'$. But what if b_1 was already quite large? Then at the end of this procedure we still have at least one large vector in the lattice. The fix is that if $|b_2'| \leq |b_1'|$, we swap and repeat. This is basically the Extended Euclidean Algorithm but applied to integer vectors. At some point this iterative process will end.

The result is that I get some b_1', b_2' such that $|b_1'| \leq |b_2'|$ and $|\mu_{2,1}| \leq \frac{1}{2}$.

The LLL algorithm essentially generalizes this to make sure that it runs in polynomial time. We define a $\sigma-$LLL reduced basis, where $\delta \in (\frac{1}{4}, 1)$. This basis is reduced if $|\mu_{i,j}| \leq \frac{1}{2}$ $\forall i > j$ (if this isn't the case, we can do more steps). Then $|\tilde{b}_{i+1}| \geq (\delta - \mu_{i+1,i}^2)|\tilde{b}_i|^2$. This last expression is a generalization of $|b_2'| \geq |b_1'|$. To see this, not that the expression when $\delta = 1$ is $|\tilde{b}_2|^2 \geq (1 - \mu_{2,1}^2)|\tilde{b}_1|^2$. Note that $|\tilde{b}_2|^2 = \langle b_2' - \mu_{2,1} b_1', b_2' - \mu_{2,1} b_1' \rangle = \langle b_2', b_2' \rangle - 2\mu_{2,1}\langle b_1', b_2' \rangle + \mu_{2,1}^2 \langle b_1', b_1' \rangle = |b_1'|^2 - \mu_{2,1}^2 |b_1'|^2 = (1 - \mu_{2,1}^2)|b_1'|^2$. Note that the tilde variables are the Gram Schmidt orthogonalization of the primed elements.

The intuition of the LLL algorithm is that you take your basis, check if it is LLL reduced. If it's not, there's a violation, you can do the Extended Euclidean Algorithm step until the violation.

Theorem: If B is $\delta-$LLL reduced, then $b_1 \leq (\frac{2}{\sqrt{\mu\delta - 1}})^{n-1} \lambda_1(L)$. This gives a $2^{O(n)}$ approximate shortest vector.

This solves the gap SVP problem for $2^{O(n)}$ approximation ratios. Next time we'll see how to use this algorithm to solve other problems.