CSC 2414 Lattices in Computer Science	October 11, 2011

Lecture 5

Lecturer: Vinod Vaikuntanathan *Scribe: Joel Oren*

In the last class, we studied methods for (approximately) solving the following two problems:

- The approximate shortest vector problem, denoted SVP_γ: We saw that for $\gamma = 2^{n/2}$, SVP_γ can be solved in time polynomial in the length of its input (this is the LLL algorithm). For $\gamma = 1$, the best known algorithm runs in time $2^{O(n)}$ (this follows from the work of Ajtai, Kumar and Sivakumar who exhibit a randomized $2^{O(n)}$-time algorithm, and Voulgaris and Micciancio who exhibit a deterministic algorithm with the same running time).
- The approximate closest vector problem, denoted CVP_γ: We saw that for $\gamma = 2^{n/2}$, CVP_γ can be solved in polynomial time (this is Babai's Nearest Plane algorithm). For $\gamma = 1$, the best known is a randomized $2^{O(n)}$-time algorithm (due to Ajtai, Kumar and Sivakumar) and an $n^{O(n)}$-time deterministic algorithm (due to Kannan).

  Obtaining a deterministic $2^{O(n)}$-time algorithm is a tantalizing open question.

Today:

- We will first show that CVP_γ is NP-complete for $\gamma = 1$. The NP-completeness is shown by a reduction from the Subset Sum problem, which is a well-known NP-complete problem.

  For $\gamma > \sqrt{n}$, it is generally believed that CVP_γ is unlikely to be NP-hard.
- We will then take a look at the NP-completeness in a different way, and use it to attack a cryptographic one-way function based on the Subset Sum problem.

  In particular, the NP-completeness reduction gives us a way to solve the Subset Sum problem, given an oracle that solves CVP exactly. Of course, CVP is hard to solve exactly in polynomial time. However, upon closer look at the reduction, we observe that the reduction in fact gives us a way to solve Subset Sum problems *on the average, for certain parameters*, given only an *approximate* solution to CVP. Instantiating this with Babai's nearest plane algorithm that computes a $2^{n/2}$-approximation to CVP, we obtain an algorithm that solves random subset sum instances for certain classes of parameters which suffices to break a proposed one-way function based on Subset Sum.

1 The Hardness of gapCVP

Before we give the hardness result, let us define the decision version of CVP_γ.

Definition 1 (gapCVP)**.** *Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, a target vector $\mathbf{t} \in \mathbb{Z}^m$, and $r \in \mathbb{Q}$, output: 1 if $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq r$, or 0 if $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) > r$.*

Definition 2 (gapCVP_γ)**.** *Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, a target vector $\mathbf{t} \in \mathbb{Z}^m$, and $r \in \mathbb{Q}$, output: 1 if $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq r$, or 0 if $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) > \gamma \cdot r$.*

Lemma 3 (without proof)**.** $\mathsf{CVP} \leq_t \mathsf{gapCVP}$

Theorem 4. gapCVP *is* **NP***-Complete.*

Proof. Observe that gapCVP is in **NP**. This is because a witness to a gapCVP instance $(\mathbf{B}, \mathbf{t}, r)$ is a vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{t} - \mathbf{x}\| \leq r$. First, it is easy to see that the witness $\mathbf{x}$ can be written down using a polynomial number of bits. Indeed, each coefficient of $\mathbf{x}$ differs from the corresponding coefficient of $\mathbf{t}$ by at

most r, meaning that representing $\mathbf{x}$ requires at most $n\lceil \log r \rceil$ bits. Furthermore, given such an $\mathbf{x}$, checking that it is a lattice vector and that $\|\mathbf{x} - \mathbf{t}\| \leq r$ can both be done in time polynomial in the input length.

We will now reduce the **NP**-complete problem of Subset Sum to gapCVP.

Definition 5 (Subset Sum)**.** *Given $A_1, \ldots, A_n \in \mathbb{Z}$ and $T \in \mathbb{Z}$ (a target value), decide if there are numbers $x_1, \ldots, x_n \in \{0,1\}$ such that $\sum_{i=1}^{n} A_i \cdot x_i = T$.*

Subset-Sum is known to be **NP**-Complete. We will now reduce it to the gapCVP problem. Given an instance of Subset-Sum $(A_1, \ldots, A_n, T)$, construct a lattice basis, target vector, and gap value as follows:

$$\mathbf{B} = \left[\mathbf{b_1}, \ldots, \mathbf{b_n}\right] = \begin{pmatrix} A_1 & A_2 & \cdots & \cdots & A_n \\ 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & & 0 & 2 \end{pmatrix} \in \mathbb{Z}^{(n+1)\times n}, \quad \mathbf{t} = \begin{pmatrix} T \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{Z}^{n+1}, \quad r = \sqrt{n}$$

Claim 6. *If $(A_1, \ldots, A_n, T)$ is a Subset-Sum instance, then $(\mathbf{B}, \mathbf{t}, r)$ is a gapCVP instance.*

Proof. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a solution to the Subset-Sum instance; i.e., $\sum_{i=1}^{n} A_i x_i = T$. Then the corresponding lattice vector $\mathbf{v} = \mathbf{B}\mathbf{x}^T = \left(\sum_{i=1}^{n} A_i x_i, 2x_1, 2x_2, \ldots, 2x_n\right)$. Thus,

$$||\mathbf{v} - \mathbf{t}|| = \left\| \begin{pmatrix} \sum_{i=1}^{n} A_i x_i \\ 2x_1 \\ \vdots \\ 2x_n \end{pmatrix} - \begin{pmatrix} T \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right\| = \sqrt{n}$$

where the last inequality follows from the fact that since $x_i \in \{0,1\}$, $|2x_i - 1| = 1$ for all $i = 1, \ldots, n$. □

Claim 7. *If there exists a $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $||\mathbf{v} - \mathbf{t}|| \leq \sqrt{n}$ then $(A_1, \ldots, A_n, T)$ is a Subset-Sum instance.*

Proof. Let $\mathbf{v} = (v_0, \ldots, v_n) \in \mathcal{L}(\mathbf{B})$ such that $\mathsf{dist}(\mathbf{v}, \mathbf{t}) \leq \sqrt{n}$. Now, $v_1, \ldots, v_n$ are all even. Thus, for all i, $|v_i - 1| \geq 1$, which gives us that

$$\mathsf{dist}(\mathbf{v}, \mathbf{t}) = \sqrt{(v_0 - T)^2 + (v_1 - 1)^2 + \cdots + (v_n - 1)^2} \leq \sqrt{n}$$

only if $v_0 = T$ (meaning, $(v_0 - T)^2 = 0$) and for all $1 \leq i \leq n$, $v_i \in \{0, 2\}$.

This shows us that $\sum_{i=1}^{n} A_i(v_i/2) = T$, concluding the proof. □

Clearly, since the reduction above runs in polynomial time, it shows that CVP is **NP**-complete. □

A natural thought to show the hardness of SVP is to come up with a reduction from CVP to SVP that works as follows: Given an instance $(\mathbf{B}, \mathbf{t})$ of CVP, consider the matrix $\mathbf{B}' = [\mathbf{B}|\mathbf{t}]$. The shortest vector would then be: $\arg\min ||\sum_{i=1}^{n} x_i b_i + x_{n+1}\mathbf{t}||$. However, this reduction does not help us in general.

2 Solving Subset Sum on the Average using Approximate CVP

Context: A proposal for a one-way function based on the Subset-Sum problem.

Definition 8 (one-way function)**.** *Let $F = \{f_n : D_n \to \mathbb{R}_n\}_{n>0}$ be a family of functions domain D_n. $f_n \in F$ is* one-way *if for all (probabilistic) polynomial time algorithm A, and for a sufficiently large n*

$$Pr[f_n(A(f_n(x))) = f_n(x)] < \frac{2}{3}$$

assuming that x is chosen from D_n according to the uniform distribution.

Proposal: Let p be an n-bit prime ($2^{n-1} < p \leq 2^n$). Define a function $f(A_1, \ldots, A_m, x_1, \ldots, x_m) = (A_1, \ldots, A_m, \sum_{i=1}^{n} A_i x_i)$, where $x_1, \ldots, x_n \in \{0, 1\}$, and $0 \leq A_i \leq p - 1 < 2^n$.

Theorem 9 (Lagarias-Odlyzko)**.** *If $n > 2m^2$ then f is not one-way.*

Proof. 1. Pick a large enough number $c > m \cdot 2^m$

2. Construct an instance of CVP:

$$\mathbf{B} = \begin{pmatrix} cA_1 & cA_2 & \cdots & cA_m \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & & 1 \end{pmatrix}_{(m+1)\times n} \qquad \mathbf{t} = \begin{pmatrix} cT \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

3. Run $\mathbf{x}^* \leftarrow NearestPlane(\mathbf{B}, \mathbf{t})$. Output $\mathbf{x}^*$.
Does this work?

Claim 10. $dist(\mathbf{t}, lat(\mathbf{B})) \leq \sqrt{m}$.

Proof Idea This is true since by definition $T = \sum_{i=1}^{n} x_i A_i$ for some $\mathbf{x} \in \{0, 1\}^m$. □

Reminder: The $NearestPlane$ algorithm returns a vector $\mathbf{x}^* \in \mathbb{Z}^m$ such that $||B\mathbf{x}^* - \mathbf{t}|| \leq \sqrt{m} \cdot 2^{m/2}$. In order to conclude the proof, two things need to be proved:

1. $\mathbf{x} \in \{0, 1\}^m$
2. $\sum_{i=1}^{n} A_i x_i^* = T$ (i.e. the returned vector is an exact solution).

We will begin with the second item first.

Claim 11. $\sum_{i=1}^{n} A_i x_i^* = T$

Proof. Say $\sum_{i=1}^{n} A_i x_i^* \neq T$. Then $||\mathbf{B}\mathbf{x}^* - \mathbf{t}|| \geq c > m \cdot 2^m$ (contradicting the approximation of $NearestPlane$). □

Claim 12. $|x_i^*| \leq 2^{m/2} \cdot \sqrt{m} < 2^m$

Proof. Suppose for contraction that the claim is false. Then $||\mathbf{B}\mathbf{x}^* - \mathbf{t}|| \geq |x_i^*| > 2^{m/2} \cdot \sqrt{m}$, which is a contradiction to the approximation of the $NearestPlane$ algorithm. □

We now claim that the probability of having a different solution vector $\mathbf{x} \in \{0,1\}^m$ than $\mathbf{x}^*$, returned by the $NearestPlane$ algorithm is small. We let $\mathbf{z} = \mathbf{x} - \mathbf{x}^*$. Note that due to the previous claim, $|z_i| \leq 2^m + 1$.

Claim 13.

$$Pr[\exists \mathbf{z} \neq 0 : \sum_{i=1}^{m} z_i A_i] \leq \frac{1}{2}$$

Proof. Fix $z_1, \ldots, z_m$, then since $A_1, \ldots, A_m$ are chosen uniformly at random, $Pr[\sum_{i=1}^{m} z_i \cdot A_i = 0 mod p] = \frac{1}{p}$. Note: we choose $1 < 2^{n-1} \leq 2^n$ in such a way that we get uniform randomness over $\mathbb{Z}_p$. Using the union bound we get:

$$Pr[\exists z : \sum_{i=1} m z_i \cdot A_i = 0] \leq (2 \cdot 2^m + 1)^m \cdot \frac{1}{p} \leq \frac{1}{p} \cdot 2^{m^2} \leq \frac{1}{2}$$

where in the first inequality we use the fact the bound on the sum in $\mathbb{Z}_p$ provides an upper bound on the sum in $\mathbb{Z}$. □

□

3 Small Solutions to Polynomial Equations Modulo Composites

Theorem 14 (Håstad, Coppersmith)**.** *Let $f(x) \in \mathbb{Z}_N[x]$ be of degree d and monic (the leading coefficient is 1). There is a polytime (in $(logN, d)$) algorithm that finds all x such that (1) $f(x) = 0(modN)$ (2) $|x| \leq N^{1/d} = \beta$. (today we will discus cases where $|x| \leq N^{\frac{2}{d(d+1)}}$)*

Note:

1. f has to be monic: consider $p \cdot x = 0(mod 2p)$. In that case, all the even numbers are solutions to this equation (in which case, the given running time is insufficient).
2. $|x| \leq N^{1/d}$: consider $x^r = 0(modp^r)$.

Proof Idea Let $f(x) = \sum_{i=0}^{d} a_i x^i$. If we assume that $|a_i \beta_i| \leq \frac{N}{d+1}, \forall i$, then $|f(x)| \leq |f(\beta)| < N \Rightarrow f(x) = 0$ (over $\mathbb{Z}$). Hence, we would want to find an equivalent polynomial with the same solutions that satisfies the inequality (but with small coefficients). Let $Z = \{N, Nx, \ldots, Nx^{d-1}, f(x)\}$.

Fact 15. *Let $a_1, \ldots, a_{d+1} \in \mathbb{Z}_N$. Then $\sum_{i=0}^{d} a_i N x^i + a_{d+1} f(x) = 0 modN$ iff $f(x) = 0 modN$.*

1. Consider the lattice defined by the following basis:

$$\mathbf{B} = \begin{pmatrix} N & 0 & \cdots & & & a_0 \\ 0 & N\beta & 0 & \cdots & & a_1\beta \\ \vdots & & N\beta^2 & \cdots & & a_2\beta^2 \\ & & & \ddots & & \\ & & \cdots & & N\beta^{d-1} & \vdots \\ 0 & & & & & a_d\beta^d = \beta^d \end{pmatrix}_{(d+1)\times(d+1)}$$

2. Run $LLL(\mathbf{B})$ to get $\mathbf{z} = (z_0, \ldots, z_{d+1}) \in \mathbb{Z}^{d+1}$ such that $||\mathbf{Bz}||$ is approximately small (i.e. an approximation to the SVP).

3. Output $\sum_{i=1}^{d} z_i(Nx_i) + z_{d+1} \cdot F(x) \triangleq g(x)$.

By LLL and Minkowski's first theorem:

$$||\mathbf{Bz}|| \leq 2^{d+1}(det(\mathcal{L}))^{1/(d+1)} = c_d \cdot (N^d(\prod_{i=1}^{d} \beta^i))^{\frac{1}{d+1}} = c_d \cdot N^{d/(d+1)} \cdot \beta^{d/2} < \frac{N}{d+1}$$

$$\Rightarrow \beta \leq c_d N^{\frac{2}{d(d+1)}}$$

□