6.876 Advanced Topics in Cryptography: Lattices *Instructor: Vinod Vaikuntanathan*

Problem Set 1

Handed Out: October 1, 2015 *Due: October 24, 2015*

Notes

- This problem set is worth 100 points.
- Collaboration is allowed, *but you must write up the solutions by yourself without consulting to notes from the discussions.* You must also reference your sources.
- Grading is based on correctness as well as the clarity of the solutions. When writing proofs, it is generally a good idea to first explain the intuition behind your solution in words (wherever appropriate), before jumping in to the formalisms.
- *Notation:* $\mathbb{N}$ denotes the natural numbers, $\mathbb{Z}$ denotes the integers, $\mathbb{Q}$ denotes rational numbers and $\mathbb{R}$ the set of real numbers.

Warmup: Lattice Bases (10 points)

Consider the basis

$$\mathbf{B} = \begin{pmatrix} 123 & 1 \\ 6764 & 55 \end{pmatrix}$$

- Which of the following vectors belong to the lattice $\mathcal{L}(\mathbf{B})$?

$$\mathbf{v}_1 = \begin{pmatrix} 129 \\ 143 \end{pmatrix} \qquad \mathbf{v}_2 = \begin{pmatrix} 1/2 \\ 10 \end{pmatrix} \qquad \mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

- What is the determinant of $\mathcal{L}(\mathbf{B})$?
- Find the Gram-Schmidt orthogonalization of $\mathbf{B}$.
- Find a shortest vector in $\mathcal{L}(\mathbf{B})$ (note that there may be many).
- Find a shortest basis of $\mathcal{L}(\mathbf{B})$ (note that there may be many).

Problem 2: Bases (20 points)

- Given a basis B, check if $\mathcal{L}(B)$ is a cyclic lattice, where a lattice $\mathcal{L}$ is called cyclic if for every lattice vector $\mathbf{x} \in \mathcal{L}$, any cyclic rotation of the coordinates of x is also in $\mathcal{L}$. For example, the lattice $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ where $\mathbf{b}_1 = (2, 0, 0)^T$, $\mathbf{b}_2 = (0, 2, 0)^T$ and $\mathbf{b}_3 = (1, 1, 1)^T$ is cyclic.
- Describe a procedure that given any set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$, find a basis for the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ (notice that these vectors are not necessarily linearly independent and that in particular, n might be greater than m). There is no need to analyze the running time. A corollary is that any set of vectors in $\mathbb{Z}^m$ spans a lattice.

Problem 3: Minkowski's First Theorem (20 points)

- (5 points) Find the analog of Minkowski's first theorem for the ℓ_1 and ℓ_∞ norms.

  [*Hint: Which part of the proof of Minkowski's first theorem is specific to the ℓ_2 norm?*]

- (15 points) Despite lattices with much shorter vectors than predicted, Minkowski's theorem is tight for general lattices. In particular, there is a family of lattices $\{\mathcal{L}_n\}_{n\in\mathbb{N}}$ where $\mathcal{L}_n$ lives in n dimensions, and

  $$\lambda_1(\mathcal{L}_n) \geq c \cdot \sqrt{n} \cdot \det(\mathcal{L}_n)^{1/n}$$

  where c is a universal constant independent of n.

  Show that such a family of lattices exists (your proof doesn't have to construct this family, you merely have to show existence).

Problem 4: Properties of LLL-Reduced Bases (20 points)

Show that a δ-LLL reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of a lattice L with $\delta = 3/4$ satisfies the following properties.

1. $\|\mathbf{b}_1\| \leq 2^{(n-1)/4} \cdot \det(L)^{1/n}$.
2. For any $1 \leq i \leq n$, $\|\mathbf{b}_i\| \leq 2^{(i-1)/2} \cdot \|\widetilde{\mathbf{b}}_i\|$.
3. $\prod_{i=1}^n \|\mathbf{b}_i\| \leq 2^{n(n-1)/4} \cdot \det(L)$.
4. For $1 \leq i \leq n$, consider the hyperplane $H = \mathsf{Span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n)$. Show that

   $$2^{-n(n-1)/4}\|\mathbf{b}_i\| \leq \mathsf{dist}(H, \mathbf{b}_i) \leq \|\mathbf{b}_i\|$$

   Hint: use (3).

Problem 5: Rounding to find an Approximately Close Lattice Vector (15 points)

Show that there is a constant $c > 0$ such that the following algorithm, given a basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, finds a lattice point $\mathbf{y} \in \mathcal{L}(\mathbf{B})$ where

$$\|\mathbf{y} - \mathbf{t}\| \leq 2^{cn} \cdot \mathsf{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$$

Algorithm Round$(\mathbf{B}, \mathbf{t})$:

1. Run the LLL-reduction algorithm on $\mathbf{B}$ to get an LLL-reduced basis $\mathbf{B}'$.
2. Find $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{R}^n$ such that $\mathbf{B}'\mathbf{s} = \mathbf{t}$, say, by Gaussian Elimination.
3. Let $\hat{\mathbf{s}} \triangleq (\lfloor s_1 \rceil, \ldots, \lfloor s_n \rceil)$ be the vector consisting of the entries of $\mathbf{s}$ rounded to the nearest integer. (e.g., $\lfloor 0.5 \rceil = 1$ and $\lfloor 0.49 \rceil = 0$).

   Output $\mathbf{y} = \mathbf{B}'\hat{\mathbf{s}}$.

Problem 6: Running Time of LLL (15 points)

Show that our analysis of the LLL algorithm using LLL-reduced bases is tight (up to some constant). More specifically, find a δ-LLL reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for $\delta = 3/4$ such that $\mathbf{b}_1$ is longer than the shortest vector by a factor or $c \cdot 2^{n/2}$, for some constant c.
(Note that this does not mean that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is the output of the LLL algorithm when run on some input basis. You do not have to demonstrate that.).

Extra Credit*

For any vector $\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{Z}^n$, let $\mathsf{Rot}(\mathbf{v}) \triangleq (v_2, v_3, \ldots, v_n, v_1)$ denote the cyclic rotation of $\mathbf{v}$. A cyclic lattice is one that is closed under the $\mathsf{Rot}(\cdot)$ operation. That is, a lattice L is cyclic if for every $\mathbf{v} \in L$, $\mathsf{Rot}(\mathbf{v}) \in L$ too. Show any of the following:

- CVP on cyclic lattices is NP-hard (Recall, we saw in class that CVP for general lattices is NP-hard).
- An interactive proof for gapCVP_γ on cyclic lattices, for any $\gamma = o(\sqrt{n/\log n})$, improving on the Goldreich-Goldwasser interactive proof we saw in class.
- A polynomial-time algorithm that finds $2^{o(n)}$-approximate shortest vectors on cyclic lattices, improving on the LLL algorithm.