Notes for Lecture 8 - Lattices Continued

1 Review

Definition 1 *Operational Definition of a Lattice*

The lattice $L(B)$ is the set of integer linear combinations of the column vectors of the matrix B. This condition is equivalent to that of being a discrete subgroup of $\mathbb{R}^n$ but easier to reason about in practice.

Definition 2 *Shortest Vector Problem (SVP)*

Given some matrix B find the vector $v \in L(B)\backslash\{0\}$ which minimizes $||v||_2$

We can also define an approximate problem SVP_γ which is to find some vector v' such that $||v'||_2 \leq \gamma||v||_2$ where v is the optimal vector.

Definition 3 *Closest Vector Problem (CVP)*

Given some matrix B and some point t find the vector $v \in L(B)$ which minimizes $||v - t||_2$

Note that we can also define gap and approximation variants for both of these.

2 Special Classes of Lattices

Let $q \in \mathbb{Z} : q \geq 2$ where q is not necessarily prime and $A \in {\mathbb{Z}_q}^{m\times n}$ where $m, n \in \mathbb{Z} : m > n > 0$

1. $\Lambda_q^\perp(A) = \{x \in \mathbb{Z}^m : x^T \cdot A = 0^n \mod q$

   We note that this is clearly a discrete subgroup of $\mathbb{Z}^m$ as it trivially contains the identity and inverses. As for addition if $x^T \cdot A, y^T \cdot A = 0 \mod q$ then $(x + y)^T \cdot A = x^T \cdot A + y^T \cdot A = 0 \mod q$

   We can additionally go by our more operational definition and construct a set of column vectors B. Assume that A is of full rank. We can then find the left kernel of A, $C \in \mathbb{Z}^{m-n} : C^T A = 0 \mod q$. This is by itself insufficient as the lattice contains all vectors for which each coordinate is a multiple of q, this subset forms a full rank sub-lattice. But we can easily resolve this by setting $B = (C|qI_m)$.

2. $\Lambda_Q(A) = \{x \in \mathbb{Z}^m : \exists r : x = Ar \mod q\}$

   The analysis of this is similar. It is obviously a discrete subgroup and can be generated by $(A|qI_m)$

Remark 4 *On Lattice Bases*

Although neither set of generating vectors is a basis (there are two many vectors) this is fine because any set of integer vectors generates a lattice as it is a subgroup of $\mathbb{Z}$*. This is not in general true for real vectors (consider the set of vectors in* $\mathbb{R}^1$ $\{1, \sqrt{2}\}$

3 Deriving Cryptographic problems from Lattices

1. Short Integer Solution (SIS)

   Let q, m, β be functions of our security parameter n. Then sample $A \leftarrow \mathbb{Z}_q^{m \times n}$ uniformly at random. We then attempt to find some $x \in \mathbb{Z}^m \setminus \{0\}$ such that $||x||_2 \leq \beta$ and $A^T \cdot x = 0^n \mod q$.

   Note that this is equivalent to solving SVP_γ over $\Lambda_q^\perp(A)$ where $\gamma = \frac{\beta}{optDist}$ where optDist is the shortest vector and $optDist \approx \sqrt{m}$.

   This is because if $m > cn \log q$ where c is some constant then with high probability there exists some $x \in \Lambda_q^\perp(A) : x \in \{0, 1\}^m$.

   From this we can derive the assumption that $SIS_{q,m,\beta}$ is hard for appropriate choices of q, m, β if SVP_γ is hard.

2. Hash Functions from SIS

   For $A \in \mathbb{Z}_q^{m \times n}$ define $f_A : \{0, 1\}^m \to \mathbb{Z}_q^n$ where $f_A(x) = A^T x \mod q$.

   We will now show that this is a good hash function

   (a) f_A will be compressing for $m > n \log(q)$ simply by looking at the number of bits necessary to represent an arbitrary vector in $\mathbb{Z}_q^n$

   (b) We can derive collision resistance by hardness of SIS. Assume we have some $x_0, x_1 \in \{0, 1\}^m : x_0 \neq x_1, f_A(x_0) = f_A(x_1)$ In that case we know that $A^T(x_0 - x_1) = 0 \mod q$. But as $x_0 - x_1$ is a vector in $\{-1, 0, 1\}^m$ and thus also has a norm $\leq \sqrt{m}$ which will with high probability be short relative to the optimal vector.

Remark 5 *This is a very hash function as it can be written as just a subset sum of the set of columns. However, the matrix A can take a significant amount of memory to store*

3. Learning with Errors (LWE)

Definition 6 *Discrete Gaussian Distribution*

There is a distribution D' over $\mathbb{Z}$ where we will define $D'_{\sigma,\mu}(x) = \mathbb{P}[x : x \leftarrow G_{\sigma,\mu}] = e^{-\pi\frac{(x-\mu)^2}{\sigma^2}}$ where G corresponds to a continuous Gaussian. Note that the constant here is slightly different than the standard case

We will then construct the actual distribution D simply by normalizing such that $D_{\sigma,\mu} = \frac{D'_{\sigma,\mu}(x)}{\sum_{s\in\mathbb{Z}} D'_{\sigma,\mu}(s)}$

Letting A be a matrix again. We note that if given $u = A \cdot s$ it is easy to retrieve s with linear algebra. To get the LWE problem we let q, m, σ me functions of m and sample A from $\mathbb{Z}^{m\times n}$, s from $\mathbb{Z}_q^n$ and e from $D^m_{,0}$ (that is a length m vector of errors).

We then compute $u = A \cdot S + e \mod q$.

From here we have two variants of the problem:

(a) Search: Given A, u find s. Note that this is similar to solving CVP_γ on $\Lambda_q(A)$

(b) Decision: Distinguish A, u from a random A and random u.

(c) We note that these problems are equally hard, given search we can easily solve decision by finding a candidate for s and seeing if it works.

To go in the other direction is slightly harder but we can use decision to solve search one coordinate at a time. Assume we are trying to find the first coordinate of s, we guess some $c \in \mathbb{Z}_q$. We then sample some random $r \in \mathbb{Z}_q$. Add r to the relevant coordinate in the columns of A and add rk to the relevant coordinate of u. Then run the decision algorithm. With high probability this will only work if k is the correct guess for this coordinate. We can then repeat for each coordinate.

(d) Defining Public Key Encryption from LWE

$Gen()$ samples $A \in \mathbb{Z}_q^{m\times n}, s \in \mathbb{Z}_q^n, e \in D^m_{\sigma,0}$ as before. We then let the secret key $sk = (A, s)$ and the public key $pk = (A, As + e \mod q)$.

We then define $Enc(pk, M)$ where $M \in \{0,1\}$. Choose $x \in \{0,1\}^n$ and return $c = (x^T A \mod q, u \cdot x + \lceil \frac{q}{2} \rfloor M \mod q$ (The $\frac{q}{2}$ is simply a number near $\frac{q}{2}$. We mask the message with u and multiplying by $\frac{q}{2}$ ensures that messages that are nearby in message space aren't close together in ciphertext space.

We can similarly define $Dec(sk = (A, s), c = (y, z))$. We can then let compute $z - y \cdot s = (u \cdot x + \lceil \frac{q}{2} \rfloor M) - (x^T A) \cdot s \mod q = ((As + e) \cdot x + \lceil \frac{q}{2} \rfloor M) - (x^T A) \cdot s \mod q = x^T e + \frac{q}{2} \rfloor M \mod q$. But because $x^T e$ is just a sum of a subset of the entries of e and e is pulled from a distribution with width $\approx sigma$ and thus $|x^T e| \approx \sqrt{m} \sigma n^{o(1)}$. We can then define our residues to be within the range $-\frac{q}{2}$ to $\frac{q}{2}$. In this case if $M = 0$ then the value we compute should be near 0 if it is 1 then it should be far away from 0 which are distinguishable with high probability.

This can be scaled to larger messages by choosing sufficiently large q to allow for spacing of messages.

(e) Proving that this is CPA Secure *Because we are only allowing messages 0,1 the adversary will always submit exactly both in the CPA experiment*

We construct a hybrid proof where H_0 is the CPA-experiment, H_1 is the same but the encryption function replaces u with a random vector. H_0 and H_1 can't be distinguished by the LWE assumption. For H_2 we note that if A, x, u are random then $A, u, x^T A, x^T u$ are statistically indistinguishable from random and thus security holds.

Remark 7 *Learning Parity with Noise (LPN): There is a similar problem LPN where* $q = 2$ *and* e *is sampled from some Bernoulli distribution where it is* 1 *with some small probability* ϵ *and* 0 *otherwise. Unlike LWE - LPN doesn't have a connection with Lattices where* LWE *and* SIS *are as hard* $GAP - CVP_\gamma, GAP_S VP_\gamma$ *in the worst case.*

Additionally, LWE instances can be added together to get a new LWE instance which isn't true for LPN.