Lattices and Homomorphic Encryption, Spring 2013 Instructors: Shai Halevi, Tal Malkin

LWE Hardness

March 12, 2013

We sketch the proof due to Regev [Reg09] and Peikert [Pei09] that (under certain conditions) it is possible to relate the *average-case* hardness of the learning with errors problem (LWE) to the *worst-case* hardness of bounded distance decoding in a given lattice (BDD).

Preliminaries. We have the following parameters:
n - security parameter.
α - noise parameter ($= \frac{1}{\mathrm{poly}(n)}$).
q - modulus ($\gg \frac{1}{\alpha}$, sometimes even $q = \exp(n)$).

We use $\mathcal{D}_s$ to denote the continuous Gaussian distribution with parameter s, and $\mathcal{D}_{L,s}$ to denotes a discrete distribution over a lattice (or coset of a lattice) L, such that every vector $\vec{z} \in L$ has probability mass proportional to $\mathcal{D}_s(\vec{z})$.

1 The Main Lemma

In addition to an oracle that solves LWE, the reduction from BDD in a lattice $\mathcal{L}$ to the average-case LWE, also needs access to an oracle that samples short vectors in $\mathcal{L}^*$. (Regev [Reg09] and Peikert [Pei09] show how to construct such a sampling oracle in specific settings, see Section 3). Additionally it relies on the following properties of the LWE error distribution:

- The LWE error distribution $\Phi_{\alpha q}$ is a continuous one-dimensional Gaussian, which is a projection of the spherical n-dimensional distribution $\mathcal{D}_{\alpha q}$ onto its first coordinate.
- The distribution $\mathcal{D}_{\alpha q}$ is smooth in the following sense: If $\mathcal{L}$ is some lattice (or coset of a lattice) with $\lambda_n(\mathcal{L}) \ll \alpha q$, then if we choose $\vec{x} \leftarrow \mathcal{D}_{\mathcal{L},r}$ and $\vec{y} \leftarrow D_s$ such that $r^2 + s^2 = (\alpha q)^2$ then the induced distribution on $\vec{x} + \vec{y}$ is close to the continuous distribution $\mathcal{D}_{\alpha q}$.

Lemma 1 ([Reg09]). *There is an efficient algorithm that takes as input a basis B of an n-dimensional lattice $\mathcal{L} = \mathcal{L}(B)$, another parameter $r \gg \frac{q}{\lambda_1(\mathcal{L})}$ and a point $\vec{x} \in \mathbb{R}^n$ such that* $\mathrm{dist}(x, \mathcal{L}) < \frac{\alpha q}{\sqrt{2}r}$ *and has access to two oracles:*

- *A "global" solver for* $\mathsf{LWE}[n, \alpha, q]$ *("global" in the sense that it is unrelated to the input lattice).*
- *A "lattice specific" sampler from $D_{\mathcal{L}^*,r}$.*

The algorithm finds (with overwhelming probability) the (unique) point $\vec{v} \in \mathcal{L}$ closest to $\vec{x}$.

2 Proof Sketch of Lemma 1

Let $\vec{v} \in \mathcal{L}$ be the closest point to $\vec{x}$ in $\mathcal{L}$ and let $\vec{t} \in \mathbb{Z}^n$ be the coefficients of $\vec{v}$ when expressed in basis B (i.e., $\vec{v} = B\vec{t}$) and denote $\vec{s} \stackrel{\mathrm{def}}{=} \vec{t} \bmod q$. We show a procedure that uses the sampler for $\tilde{D}_{\mathcal{L}^*,r}$ to generate instances of the distribution $\mathsf{LWE}_{\vec{s}}$. Then, we use the LWE solver to find $\vec{s}$. (Note that $\vec{s}$ was not chosen uniformly at random in this case, but we previously showed a random self reduction for LWE from a random $\vec{s}$ to any specific $\vec{s}$.) Later we show how from $\vec{s}$ one can find $\vec{t}$ thereby solving BDD.

LWE-Generate(B, $\vec{x}$) (With access to $\tilde{D}_{\mathcal{L}^*,r}$)

1. Draw a sample $\vec{y} \leftarrow \tilde{D}_{\mathcal{L}^*,r}$. Let $\vec{a}$ be the coefficients of $\vec{y}$ in basis B^* (i.e. $\vec{a} = B^T\vec{y}$).
2. Draw an error term $e \leftarrow \Phi_{\frac{\alpha}{2\sqrt{\pi}}}$.
3. Output $(\vec{a}, b = \langle \vec{x}, \vec{y}\rangle + e \bmod q)$.

Claim 1. *The output of LWE-Generate is statistically close to the LWE distribution with secret $\vec{s}$, $\mathsf{LWE}_{\vec{s}}$, except that the error parameter is some $\beta \leq \alpha$.*

Proof. We need to show that (A) $\vec{a}$ is close to uniform in $\mathbb{Z}_q^n$, and (B) once $\vec{a}$ is fixed, $\vec{b} = \langle \vec{s}, \vec{a}\rangle + \Phi_{\beta q}$ for some $\beta \leq \alpha$.

(A.) Consider the lattice $q \cdot \mathcal{L}^*$ and all its q^n cosets

$$\vec{a}\text{-coset} = \{B^*\vec{a} + q\mathcal{L}^*\} = \{B^*\vec{z} : \ \vec{z} = \vec{a} \bmod q\}$$

The vector $\vec{a}$ output by the LWE-Generate procedure is exactly the coset of $\vec{y}$. Due to our choice of parameters, all cosets are (almost) equally likely. Indeed, since $r \gg \frac{q}{\lambda_1(\mathcal{L})} \geq \frac{q\lambda_n(\mathcal{L}^*)}{n}$ then $\tilde{D}_{\mathcal{L}^*,r}$ is nearly uniform among the cosets.

(B.) Conditioned on any fixed $\vec{a} \in \mathbb{Z}_q^n$, the vector $\vec{y}$ is chosen from the discrete distribution on the $\vec{a}$-coset, $\vec{a} + D_{q\mathcal{L}^*,r}$. Denoting $\vec{w} \stackrel{\text{def}}{=} \vec{x} - \vec{v}$ we have

$$\begin{aligned}
\langle \vec{x}, \vec{y}\rangle &= \langle \vec{v} + \vec{w}, \vec{y}\rangle \\
&= \langle \vec{v}, \vec{y}\rangle + \langle \vec{w}, \vec{y}\rangle \\
&= \langle B\vec{t}, \vec{y}\rangle + \langle \vec{w}, \vec{y}\rangle \\
&= \langle \vec{t}, B^T\vec{y}\rangle + \langle \vec{w}, \vec{y}\rangle \\
&= \langle \vec{s}, \vec{a}\rangle + \langle \vec{w}, \vec{y}\rangle \bmod q
\end{aligned}$$

hence $b = \langle \vec{s}, \vec{a}\rangle + \langle \vec{w}, \vec{y}\rangle + e \bmod q$. Notice that $\vec{s}$, $\vec{a}$ and $\vec{w}$ are fixed and the random part is just $\vec{y}$ and e.

Recall that $\Phi_{\frac{\alpha}{2\sqrt{\pi}}}$ is the projection of $\mathcal{D}_{\frac{\alpha}{2\sqrt{\pi}}}$ onto the first coordinate, namely $\langle \vec{e_1}, \mathcal{D}_{\frac{\alpha}{2\sqrt{\pi}}}\rangle$ and since $\mathcal{D}$ is spherical then this is also the same as $\langle \vec{u}, D_{\frac{\alpha}{2\sqrt{\pi}}}\rangle$ for any other unit vector $\vec{u}$. In particular, $\Phi_{\frac{\alpha}{2\sqrt{\pi}}} \equiv \langle \vec{w}, \mathcal{D}_{\frac{\alpha}{2\sqrt{\pi}}}\rangle \frac{1}{||\vec{w}||} \equiv \langle \vec{w}, \mathcal{D}_{\frac{\alpha}{2\sqrt{\pi}||\vec{w}||}}\rangle$.

Hence $\langle \vec{w}, \vec{y}\rangle + e \equiv \langle \vec{w}, \vec{y}\rangle + \langle \vec{w}, \vec{z}\rangle = \langle \vec{w}, \vec{y} + \vec{z}\rangle$ where $y \in_R \mathcal{D}_{\vec{a}+q\mathcal{L}^*,r}$ and $z \in_R \mathcal{D}_s$ where $s = \frac{\alpha}{2\sqrt{\pi}||\vec{w}||}$. Now $||\vec{w}||$ is "short" so s is "large". The parameters r, s are chosen large enough so that $\mathcal{D}_{q\vec{a}+\mathcal{L}^*,r}$ is close to the continuous $\mathcal{D}_t$ where $t = \sqrt{r^2 + s^2}$. Therefore $\langle \vec{w}, \vec{y}\rangle + e \approx \langle \vec{w}, \mathcal{D}_t\rangle = \Phi_{||\vec{w}||\cdot t}$ and the parameters are such that $||\vec{w}|| \cdot t \leq \alpha q$. □

To solve BDD for $\vec{x}$ we can apply the LWE-solver with samples from LWE-Generate to find the vector $\vec{s}$. However, to solve BDD we need to find $\vec{t}$ (recall $\vec{s} = \vec{t} \bmod q$). To do this, first observe that $\vec{v} = B\vec{t} = B\vec{s} + B(q\vec{z})$ for some $\vec{z} \in \mathbb{Z}^n$ and consider $\vec{x'} = \frac{\vec{x} - B\vec{s}}{q} = \frac{\vec{x} - \vec{v}}{q} + B\vec{z}$. Notice that by this calculation, the vector $\vec{x'}$ is at distance $\frac{||\vec{w}||}{q}$ (where $\vec{w} = \vec{x} - \vec{v}$) from the lattice (specifically the point $B\vec{z}$). If we could find the closest lattice point to $\vec{x'}$ we would have $\vec{z}$ and therefore also $\vec{v}$. To do this just repeat the above argument again and again and at each iteration the distance from the lattice is reduced by a factor of q. After n such iterations we can solve the problem by using, e.g., Babai's nearest plane algorithm.

3 The Lattice-Specific Sampler

Regev [Reg09] described a quantum algorithm for implementing the lattice-specific sampling oracle, thus obtaining a quantum reduction of BDD to LWE. Peikert observed [Pei09] that in some cases the sampler can also be implemented using a standard (non-quantum) efficient algorithm, specifically when the parameter α is small enough relative to $\lambda_1(\mathcal{L})$. This yields a reduction from the problem of approximating the number $\lambda_1(\mathcal{L})$ to LWE: Roughly we try the reduction with different size of α until it fails, and that value of α is an approximation of $\lambda_1(\mathcal{L})$. Peikert's observation is based on the following theorem of Gentry et al. [GPV08]:

Theorem 1 (Informal)**.** *Given a basis $B = (b_1 \dots b_n)$ for a lattice $\mathcal{L} = \mathcal{L}(B)$, it is possible to sample efficiently from the discrete Gaussian distribution distribution $\mathcal{D}_{\mathcal{L},s}$ for a parameter $s \geq$ poly$(n) \cdot \max_i \|b_i\|$.*

(The poly(n) term can be as small as $\sqrt{n}$.) Moreover, using the LLL algorithm we can find a basis B^* for $\mathcal{L}^*$ such that $\max_i \|b_i^*\| \leq 2^{n/2}\lambda_1(\mathcal{L}^*) \leq 2^{n/2}n/\lambda_1(\mathcal{L})$. Hence we can use the GPV sampler to sample from $D_{\mathcal{L}^*,r}$ whenever (say) $r \geq 2^n/\lambda_1(n)$.

Theorem 2 ([Pei09] Let $\alpha = 1/\text{poly}(n)$, $\gamma = n/\alpha$ and $q = \exp(n)$. Given oracle access to a solver for LWE$[m, \alpha, q)$**.** *and any basis B for an n-dimensional lattice $\mathcal{L} = \mathcal{L}(B)$, we can approximate the number $\lambda_1(\mathcal{L})$ to within a γ factor.*

Proof. We first use LLL to find an approximation β such that $\lambda_1(\mathcal{L}) \leq b \leq 2^{n/2}\lambda_1(\mathcal{L})$. For $i = 0, 1, 2, \dots$ we define $\beta_i = \beta/\gamma_i$.

Below we describe a procedure to distinguish the two cases $\lambda_1(\mathcal{L}) < \beta_{i+1}$ and $\lambda_1(\mathcal{L}) \geq \beta_i$. Running this procedure and denoting by i^* the first index in which the procedure outputs "$\lambda_1(\mathcal{L}) \geq \beta_i$", it is clear that β_{i^*} is a λ approximation, as needed. I.e., if $\lambda_1(\mathcal{L}) \in [\beta_{i+1}, \beta_i)$ for some i, the we would output either β_{i+1} or β_i.

Distinguishing procedure. The following gets as input a basis B or $\mathcal{L} = \mathcal{L}(B)$ and a number d, and it needs to distinguish the two cases $\lambda_1(\mathcal{L}) < d$ and $\lambda_1(\mathcal{L}) \geq d \cdot \gamma$.

Distinguish(B, d).
0. Set $d' = d \cdot \sqrt{n}/2$
1. For $j = 1$ to $N = \text{poly}(n)$ do:
   (i) Draw w_i uniformly at random from the n dimensional sphere of radius d';
   (ii) Reduce w modulo $\mathcal{P}(X)$ to get $x = w \bmod \mathcal{P}(B)$;
   (iii) Run the algorithm from Lemma 1 on input basis B, parameter $r = q \cdot \sqrt{2n}/(d\gamma)$ and point x. For the two oracles use:
      — The LWE solver that you have access to as the "global" oracle
      — The GPV sampler using an LLL-reduced basis for B^*, for the "lattice specific" oracle
   (iv) Let v be the point that the algorithm from Lemma 1 returns (set $v = 0$ if the algorithm fails). If $x - w = v$ then record a vote for "$\lambda_1(\mathcal{L}) \geq d\gamma$", else record a vote for "$\lambda_1(\mathcal{L}) < d$".
2. Output "$\lambda_1(\mathcal{L}) \geq d\gamma$" if *all votes* say "$\lambda_1(\mathcal{L}) \geq d\gamma$", else output "$\lambda_1(\mathcal{L}) < d$.

Analysis. We show that (a) when $\lambda_1(\mathcal{L}) \geq d\gamma$ then all the conditions of Lemma 1 are satisfied and the closest lattice point to x is $x - w$, so in this case the algorithm from Lemma 1 will return $x - w$, and (b) when $\lambda_1(\mathcal{L}) < d$ then the view of the algorithm from Lemma 1 does not determine a unique w, so with non-negligible probability it will return a point different from $x - w$.

Case (a): $\lambda_1(\mathcal{L}) \geq d\gamma$. Recall that the distance between x and the lattice $\mathcal{L}$ is less than $d\sqrt{n}/2 = d \cdot \frac{\alpha\gamma \cdot q}{\sqrt{n}q\sqrt{2n}} = \frac{\alpha q}{\sqrt{2}r}$, as needed for the lemma. Also we have $r = \frac{q\sqrt{2n}}{d\gamma} > \frac{q\sqrt{2n}}{\lambda_1(\mathcal{L})}$. Finally, since $q = \exp(n)$ then the GPV sampler gives good enough samples. Hence the reduction works and we always get the unique closes point to x, which is $x - w$.

Case (b): $\lambda_1(\mathcal{L}) < d$. Let y be the shortest nonzero vector in $\mathcal{L}$, $\|y\| = \lambda_1(\mathcal{L}) < d$, then with non-negligible probability both $x - w$ and $x - w - y$ are within distance d' from x. Hence both are equally likely given the view of the algorithm, so it will output $x - w$ with probability at most 1/2. $\square$

References

[GPV08] Craig Gentry, Chris Peikert and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions, In *40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 197–206. ACM, 2008.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 333–342. ACM, 2009.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *JACM*, 56(6), 2009.