**6.876 Lattice Based Crypto**
**Fall 2015**

**Worst-case to Average-case Reduction for LWE**
**Date: 11/2/2015**

---

## So far:

- SIS and Applications: CRHF, OWF
- LWE and Applications: SK Enc, pK Enc, fhe

## Today:

- Sketch of LWE wc/ac reductions
- Trapdoors for lattices

We need new techniques to work with lattices and LWE. The main point is to start talking about trapdoors for lattices. This will let us construct not only one-way functions, but trapdoors.

Before going there, we will briefly sketch LWE wc/ac reductions.

## Sketch of LWE wc/ac reductions

**Theorem.** *Suppose you can solve $LWE_{n,q,\chi}$ (average-case) where $\chi$ is a Gaussian $D_{\alpha q}$ with standard deviattion $\alpha q$ with $\alpha \in [0,1)$. So each sample $(a, < a, s > +e)$ has $e \in D_{\alpha q}$. Then, you can solve $gapSVP_{n,1/\alpha \cdot n^{1.5}}$ (worst-case).*

This only applies to LWE with Gaussian distribution. Other distributions are open problems. The larger $\alpha$ is, the harder it is to solve LWE, then we should be able to find a better approximation to gapSVP, so parameter is $1/\alpha$.
Note: For SIVP, we showed a similar reduction: SIS average case to worst case $SIVP_{n,O(n)}$, and $gapSVP_{n,O(n)}$ by duality. SIS gives you a solution to shortest INDEPENDENT vector problem, but solving LWE, don't know how to solve a worst case search problem.

*Proof.* Idea: We will show that if you can solve average case $LWE_{n,q,\chi_{wc}}$ we can solve the worst case (search problem) $BDD_{n,\alpha_b dd}$. Then, if you can solve $BDD_{n,\alpha}$, you can solve worst case $gapSVP_{n,1/\alpha_b dd \cdot \sqrt{n/logn}}$.

**Definition. $\mathbf{LWE_{n,\alpha}}$:** *If $m > n \log q$, then the lattice generated by $A$, $L_q(A) = \{s^T A\} + q\mathbb{Z}^m$ is very sparse, and has a large minimum distance:*

$$\lambda_1(L_q(A)) = O(q)$$

**Definition.** $BDD_{n,\alpha}$ ***(Bounded Distance Decoding):*** *Given a point $v \in \mathbb{Z}^n$ that is at most $\alpha\lambda_1(L)$-far from $L$, with $\alpha \in [0, 1/2)$, find the closest lattice point.*

$BDD \implies gapSVP$**:** Given access to BDD how to we solve gapSVP problem? BDD only solves the closest vector problem if $v$ is $\alpha\lambda_1(L)$ away. The inputs to BDD should be close to the lattice. How should we generate good target point for which we do not know the answer?

If we have a lattice, we can pick a random lattice point, and add about $\sqrt{n}/\log n$ noise around it, call it $v$.

In one case, if $\lambda_1 \geq 2\sqrt{n/\log n}$, then $v$ is a valid input to BDD. BDD will returns $y$. In the second case, we have $\lambda_1 \leq 1$. It is unlikely that we have a unique closest vector to $v$. BDD algorithm will return $y$ or $y \pm w$ with equal probability. So if the BDD oracle return the same $y$ that we picked, then we know that $\lambda_1$ is large. If it doesn't, then $\lambda_1$ is small.

This generalizes to any $\alpha$ for BDD, by distinguishing the case with $\lambda_1 \geq 2\sqrt{n/logn} \cdot \frac{1}{\alpha}$ and $\lambda_1 \leq 1$. Thus, we can solve $gapSVP_{n,\frac{1}{\alpha}\cdot\sqrt{n/logn}}$.

$(ac)LWE_{n,q,\alpha_{wc}} \implies (wc)BDD_{n,\alpha_{bdd}}$: Given an average-case LWE oracle, and we want to solve worst-case BDD.

Input: Lattice L, Vector $v \in \mathbb{Z}^n$, and we want to find the closest lattice vector.

Let's say $y \in L$ is the closest lattice vector to $v$, in other words, $v = y + e$, where $\|e\| \leq \alpha_{bdd} \cdot \lambda_1(L)$. We can write $v = Bs + e$. Want to find $s$. How can we use ac LWE to solve this problem? $s$ is going to remain the same: $s$ is the worst case BDD secret, and $s$ is the average case LWE secret.

Idea: Sample $y^* \leftarrow D_{L*,r}$, where $L* = \{x \in \mathbb{R}^n :< x, y >\in \mathbb{Z} \forall y \in L\}$ and $r > q \cdot \eta_\epsilon(L)$. So $y* = B* \cdot a$ for some $a$, which will be our $a$'s to LWE.

$$\begin{aligned} < v, y* > &=< y + e, y* > \\ &=< y, y* > + < e, y* > \\ &= y^t y* + \dots \\ &= s^t B^t B^* a + \dots \\ &= s^t (I) a + \dots \\ &= s^t a + \dots \end{aligned}$$

But we need to show that $a$ mod $q$ looks random. If we pick a large discrete gaussian, this is possible. What happens to error? This is a discrete Gaussian with a worst case error with bounded length. Want to show that it behaves like a one-dimensional Gaussian. This would be true if $y*$ were a continuous Gaussian. To prove it for a discrete Gaussian is harder, and can be done if $r$ is large enough, so we can kill it by adding a continuous Gaussian $e'$ (this is the rough idea):

$$< v, y* > + e' \bmod q$$

□

The above work is the work of Micc-Lyubaskhecsky and Regev+Peikert.

## Lattice Trapdoors [Ajtai'99, ..., MP'12]

Recall that we have a one-way function from LWE:

$$f_A(s, w) = s^T A + e^T$$

where $A$ is an $n \times m$ lattice.

Inverting this is uniformly hard. Is it possible to make it hard for you but easy for me? In other words, can this be a trapdoor function where:

1. Given $A$, $f_A(s, e)$, it is hard to find $s$
2. Given $(A, T_A)$ ($T_a$ = trapdoor), it is easy to find $s$
3. Should be easy to sample $(A, T_A)$.

In this situation, given $s^T A + e^T$, we want some special information about $A$ that lets me distinguish that from randomness. This special information can be a short vector in the kernel of $A$ – a short solution to $At = 0 \bmod q$, a solution to SIS on $A$. Why? Given $t$, we can compute $y^T t$. If $y$ is random, we can randomness. If $y$ is LWE, we get $(s^T A_e^T)t = e^T t$ which is small. This is my **trapdoor**.

Want to use this trapdoor to not only distinguish between LWE and random, but also to break one-way-ness: we want to recover $s$ and $e$.

**Definition.** *A **trapdoor** for $A$ is a matrix of linearly independent solutions to SIS. $T_a \in \mathbb{Z}^{m \times m}$ such that:*

- *$\|T_a\|$ is small (each vector is small)*
- *$A \cdot T_a = 0 \bmod q$*
- *$T_a$ has rank $m$ over $\mathbb{Z}$*

Note: How many linearly independent vectors can we have in the kernel? At most $n - m$. So we need to have rank $m$ over the integers $\mathbb{Z}$, but they wouldn't be linearly independent mod $q$.

Given $T$, we can:

$$(s^t A + e^t) \cdot T = e^t T \bmod q$$

So we get $m$ equations for the error term. Since they are both small, we actually have $e^t T$ over the integers. We know that $T$ has full rank over $\mathbb{Z}$, so we can invert $T$ and retreive $e$. Once we have $e$, we have $s$.

Not only can we use $T_A$ to solve LWE, but we can also solve inhomogeneous SIS (ISIS):

**ISIS**: Given $v \in \mathbb{Z}_q^n$, find short $r$ such that $Ar = v \bmod q$.

To find $T_A$: Take $q = 2^k$, $g = (1, 2, 4, \ldots, 2^{k-1})$ so $L_g = \{z \in \mathbb{Z}^k :< z, g >= 0 \bmod q\}$. Find $T_g$.

$$(1, 2, 4, \ldots, 2^{k-1})T_g = 0 \bmod q$$

Easy to find some vectors $(2, -1, 0, \ldots, 0)$, $(0, 2, -1, 0, \ldots, 0)$, $(0, \ldots, 0, 2, -1)$, and $(0, \ldots, 0, 2)$. each vector is small, and rank is full, because easy to check determinant is nonzero, and not full rank mod $q$.