Key Recovery for LWE in Polynomial Time

Kim Laine and Kristin Lauter

Microsoft Research

January 9, 2016

Review of LWE

Learning With Errors (LWE)

- Asymptotically hard computational problem
- Typically used in homomorphic encryption
- Introduced by Oded Regev
- *On Lattices, Learning With Errors, Random Linear Codes and Cryptography*, 2005

Rough idea:

Solve $\mathbf{s}$ from the following system.

$$\begin{cases} \langle \mathbf{a}_0, \mathbf{s} \rangle \approx b_0 \pmod{q} \\ \langle \mathbf{a}_1, \mathbf{s} \rangle \approx b_1 \pmod{q} \\ \langle \mathbf{a}_2, \mathbf{s} \rangle \approx b_2 \pmod{q} \\ \quad \vdots \qquad\qquad \vdots \\ \langle \mathbf{a}_{d-1}, \mathbf{s} \rangle \approx b_{d-1} \pmod{q} \end{cases}$$

Notation for LWE:

- q an odd prime
- $\mathbf{a}_i, \mathbf{s} \in \mathbb{Z}_q^n$
- $e_j \in \mathbb{Z}_q$ small
- $b_j \in \mathbb{Z}_q$

Learning With Errors:

It is hard to solve secret $\mathbf{s}$ from the linear system

$$\begin{cases} \langle \mathbf{a}_0, \mathbf{s} \rangle + e_0 = b_0 \pmod q \\ \langle \mathbf{a}_1, \mathbf{s} \rangle + e_1 = b_1 \pmod q \\ \langle \mathbf{a}_2, \mathbf{s} \rangle + e_2 = b_2 \pmod q \\ \quad \vdots \qquad\qquad \vdots \\ \langle \mathbf{a}_{d-1}, \mathbf{s} \rangle + e_{d-1} = b_{d-1} \pmod q \end{cases}$$

unless e_j are known.

Definition 1 (LWE sample)

An LWE sample $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is one such equation.

Error distribution:

The discrete Gaussian distribution with standard deviation σ is a distribution $D_{\mathbb{Z},\sigma}$ on the integers such that

$$\text{Prob}(x) \propto \exp\left(-\frac{x^2}{2\sigma^2}\right) .$$

For security reductions:

In LWE the errors e_i must be sampled from wide enough $D_{\mathbb{Z},\sigma}$.

Security Properties

Definition 2 (GapSVP (roughly))

Is the shortest non-zero vector in a lattice Λ longer than a given gap γ?

Assumption: $\text{GapSVP}_{\gamma(n)}$ is very hard when $\gamma(n) = \text{poly}(n)$ and n is large enough.

Theorem 3 (Regev, Peikert (very roughly))

Suppose σ is large enough[1]. Then $\text{GapSVP}_{\tilde{O}(nq/\sigma)}$ is easy if LWE is easy.

[1] Say, bigger than $\sqrt{n}$.

Questions and concerns:

1. GapSVP is easy when the gap γ is exponential in n.
2. So if q is very large, also σ must be large for security guarantee.
3. If n, σ fixed, expect security to decrease as q grows.
4. Security for concrete parameters?

How hard is breaking LWE?

Lots of research about estimating hardness of LWE for very secure (large) parameters.

Difficult, because attacks typically use BKZ-2.0 or other lattice reduction techniques, whose performance/complexity is tricky to estimate.

For less secure (smaller) parameters the hardness has not been studied as much. What CAN we break in reasonable time?

Smaller parameters result in significantly better performing cryptosystems, so want tight security estimates.

How hard is breaking LWE?

GapSVP$_{\widetilde{O}(nq/\sigma)}$ gets easier when q increases, other parameters fixed.

No security guarantees for q exponential in n, $\sigma \ll q$.

Theorem 4 (L.-Lauter)

Any instance of LWE with $q > 2^{2n}$ can be broken in polynomial-time using roughly $2n$ samples. In practice significantly smaller q are vulnerable.

Examples of recovering the LWE secret: $(\sigma = 8/\sqrt{2\pi})$

n	Samples	$\log_2 q$	Time
80	255	16	10m
100	300	19	24m
120	335	22	61m
140	380	24	1.6h
160	420	27	2.9h
180	460	29	4.4h
200	500	32	7.2h
250	600	39	19h
300	705	45	1.8d
350	805	52	3.7d

Consider d LWE samples. Let Λ be the $(n+d)$-dimensional lattice generated by the rows of

$$\begin{bmatrix} q & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & q & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & q & 0 & 0 & \cdots & 0 \\ \mathbf{a}_0[0] & \mathbf{a}_1[0] & \cdots & \mathbf{a}_{d-1}[0] & 1/2^{\ell-1} & 0 & \cdots & 0 \\ \mathbf{a}_0[1] & \mathbf{a}_1[1] & \cdots & \mathbf{a}_{d-1}[1] & 0 & 1/2^{\ell-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{a}_0[n-1] & \mathbf{a}_1[n-1] & \cdots & \mathbf{a}_{d-1}[n-1] & 0 & 0 & \cdots & 1/2^{\ell-1} \end{bmatrix}$$

Then

$$\mathbf{v} = \left[\langle \mathbf{a}_0, \mathbf{s}\rangle_q, \langle \mathbf{a}_1, \mathbf{s}\rangle_q, \ldots, \langle \mathbf{a}_{d-1}, \mathbf{s}\rangle_q, \mathbf{s}[0]/2^{\ell-1}, \mathbf{s}[1]/2^{\ell-1}, \ldots, \mathbf{s}[n-1]/2^{\ell-1}\right] \in \Lambda$$

$$\mathbf{u} = \left[b_0, b_1, \ldots, b_{d-1}, 0, \ldots, 0\right] \notin \Lambda \text{ but is close to } \mathbf{v} \text{ if } \ell \text{ is big}$$

To recover s:

1. Use LLL to find a reduced basis for Λ.
2. Use Babai's `NearestPlanes` algorithm to find a lattice point close to $\mathbf{u}$.
3. `NearestPlanes` will recover $\mathbf{w} \in \Lambda$ with

$$||\mathbf{w} - \mathbf{u}|| = 2^{\mu(n+d)} \operatorname{dist}(\Lambda, \mathbf{u})$$

   where $\mu \leq 1/4$.

4. But $\mathbf{v}$ is such a lattice point!

How to ensure $\mathbf{v}$ is recovered and not some other $\mathbf{w} \in \Lambda$?

Theorem 5 (L.-Lauter)

If $q > 2^{2n}$ and σ is not "too large", then ℓ and the number of samples can be chosen in such a way that with overwhelming probability the only vector $\mathbf{w} \in \Lambda$ satisfying

$$||\mathbf{w} - \mathbf{u}|| \leq 2^{(n+d)/4}\, dist(\Lambda, \mathbf{u})$$

is $\mathbf{v}$.

In practice $\mu \ll 1/4$ is realized.

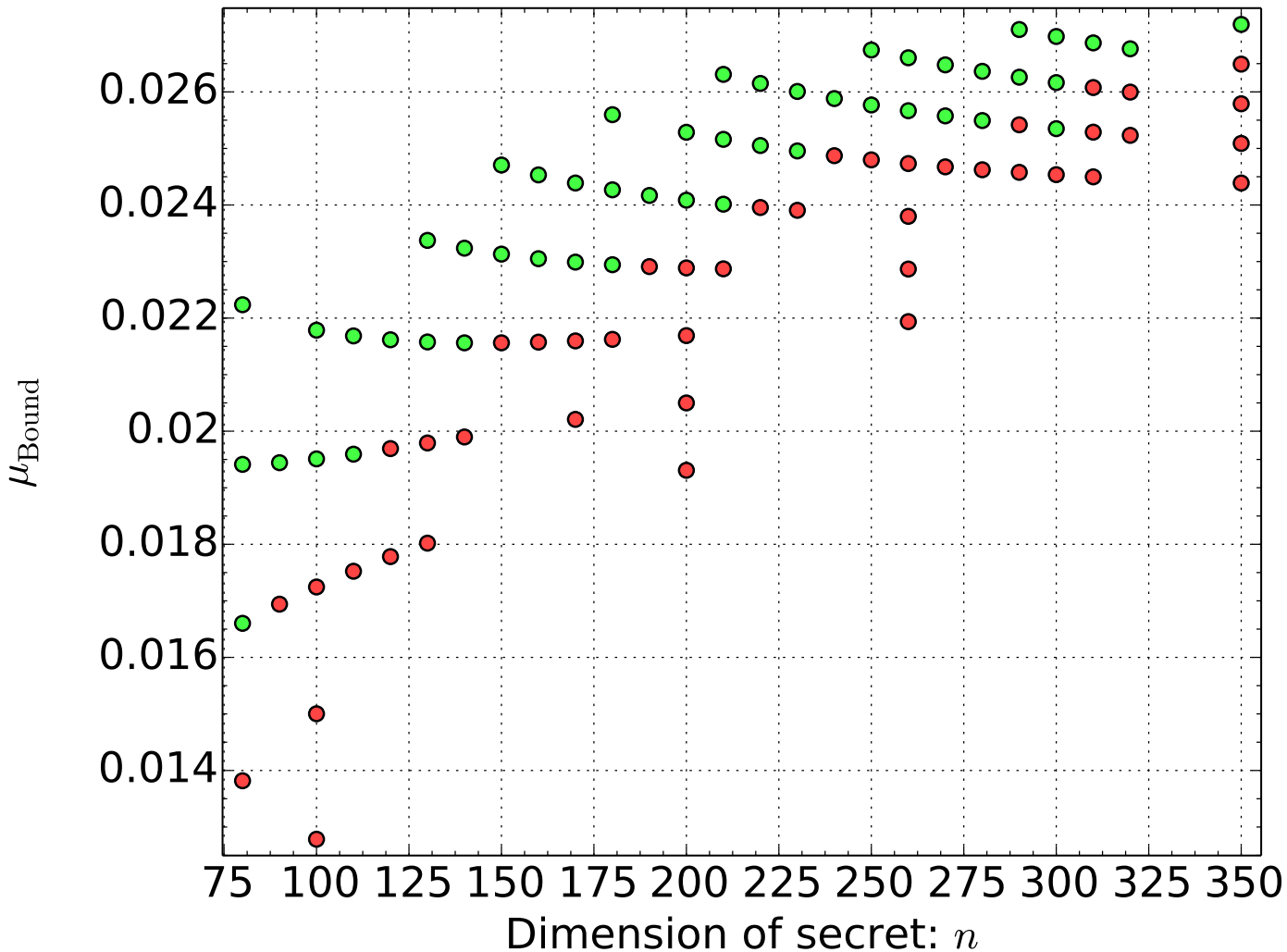
Practical attack:

1. **L.-Lauter:** Succeeds almost certainly when (d = number of samples)

$$\mu \leq \mu_{\text{Bound}} := \frac{1}{d} \log_2 \left[\frac{q^{1-n/d}}{2\sigma\sqrt{d}} - 1 \right] .$$

2. Choose d in a way that maximizes μ_{Bound}.
3. Run the lattice attack.
4. For security estimates, predict how realized μ is related to the lattice and quality of the basis.

Green dot: Secret recovery succeeded **Red dot:** Secret recovery failed



All experiments were done using SAGE, PARI/GP and fplll.

Open questions:

What happens for larger examples?

What happens if better lattice reduction is used?

What happens if Lindner-Peikert iterative `NearestPlanes` is used?

Can generalized modulus switching (switch both q and n) be used to improve the probability of the attack succeeding?

Distinguishing:

In crypto: Semantic security depends on hardness of distinguishing LWE samples from random data of the same form (*decision*-LWE)

Clear: If secret recovery (*search*-LWE) can be solved, then *decision*-LWE can be solved

Less clear: (Roughly speaking...) If *decision*-LWE can be solved, then also *search*-LWE can be solved, but this is tricky! (changes LWE parameters)

There exists a direct lattice attack against *decision*-LWE!

L.-Lauter: (Observation) Secret recovery becomes easy roughly when distinguishing becomes easy (for same LWE parameters), even without the search-to-decision reduction.

More more-or-less open questions:

- Can the special structure of the lattice be used for improved attacks?
- Does the special structure of the lattice cause phenomena that make slightly smaller parameters significantly more vulnerable than larger parameters?
- How does changing σ change the hardness of known lattice attacks on secret recovery and distinguishing? How does this change depend on whether the parameters are small or large?
- In practice a constant small σ is used. Can this be used somehow for improved attacks?

Thank you for listening!

Questions?