The Learning with Errors Problem

Advanced Topics in ~~Cybersecurity~~Cryptography (7CCSMATC)

Martin R. Albrecht

Outline

Learning with Errors

Main reference

Oded Regev. **On lattices, learning with errors, random linear codes, and cryptography.** In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: `http://doi.acm.org/10.1145/1568318.1568324`

TL;DR: The Internet will run on this stuff

NIST | COMPUTER SECURITY RESOURCE CENTER
CSRC

UPDATES 2023

Comments Requested on Three Draft FIPS for Post-Quantum Cryptography

August 24, 2023

NIST requests comments on the initial public drafts of three Federal Information Processing Standards (FIPS):

1. FIPS 203, *Module-Lattice-Based Key-Encapsulation Mechanism Standard*
2. FIPS 204, *Module-Lattice-Based Digital Signature Standard*
3. FIPS 205, *Stateless Hash-Based Digital Signature Standard*

These proposed standards specify key establishment and digital signature schemes that are designed to resist future attacks by quantum computers, which threaten the security of current standards. The three algorithms specified in these standards are each derived from different submissions to the NIST Post-Quantum Cryptography Standardization Project.

“Small Elements” mod q

We can represent $\mathbb{Z}_q$ with integers $\{0, 1, \ldots, q-1\}$

We can also represent $\mathbb{Z}_q$ with integers $\{-\lfloor q/2 \rfloor, -\lfloor q/2 \rfloor + 1, \ldots, \lfloor q/2 \rfloor\}$

Example:

```
q = 17
K = GF(q)
[[e.lift() for e in K], [e.lift_centered() for e in K]]
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	3	4	5	6	7	8	-8	-7	-6	-5	-4	-3	-2	-1

The latter representation is called “centred” or “balanced”.

We often implicitly assume the “centred” representation.

We informally say that $e \in \mathbb{Z}_q$ is “small” if its balanced representation is small in absolute value.

1-dim LWE (even easier than RSA)

KeyGen

Pick a prime $q \approx 2^{10,000}$

Pick a random integer $s \in \mathbb{Z}_q$

Pick about $t = 20,000$ random $a_i \in \mathbb{Z}_q$ and small $e_i \approx 2^{9,850}$

Publish pairs $a_i, c_i = a_i \cdot s + e_i \bmod \mathbb{Z}_q$

Encrypt $m \in \{0, 1\}$

Pick $b_i \in \{0, 1\}$

$d_0 = \sum_{i=0}^{t-1} b_i \cdot a_i$

$d_1 = \lfloor \frac{q}{2} \rfloor \cdot m + \sum_{i=0}^{t-1} b_i \cdot c_i$

Return d_0, d_1

Decrypt

Compute $d = d_1 - d_0 \cdot s$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \sum_{i=0}^{t-1} b_i \cdot c_i - \sum_{i=0}^{t-1} b_i \cdot a_i \cdot s$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \sum_{i=0}^{t-1} b_i \cdot (a_i \cdot s + e_i) - \sum_{i=0}^{t-1} b_i \cdot a_i \cdot s$$

$$= \left\lfloor \frac{q}{2} \right\rfloor \cdot m + \sum_{i=0}^{t-1} b_i \cdot e_i$$

Return 1 if $|d| > q/4$ and 0 otherwise.

Toy Implementation

```
t = 10000
q = next_prime(2^10000, proof=False); q2 = q//2

# KeyGen
s = ZZ.random_element(y=q-1)
a_ = [ZZ.random_element(y=q-1) for _ in range(t)]
e_ = [ZZ.random_element(y=2^9850) for _ in range(t)]
c_ = [(a_[i]*s + e_[i]) % q for i in range(t)]

# Enc
m = 1
b_ = [ZZ.random_element(x=0,y=2) for _ in range(t)]
d0 = sum(b_[i]*a_[i] for i in range(t)) % q
d1 = (q2 * m + sum(b_[i]*c_[i] for i in range(t))) % q

# Dec
round(((d1 - d0*s) % q)/q2), m
```

```
(1, 1)
```

The Learning with Errors Problem (LWE)

Given $(\mathbf{A}, \mathbf{c})$ with $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$ and small $\mathbf{e} \in \mathbb{Z}^m$ is

$$\begin{pmatrix} \\ \\ \mathbf{c} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \leftarrow & n & \rightarrow \\ & & \\ & \mathbf{A} & \\ & & \\ & & \end{pmatrix} \times \begin{pmatrix} \mathbf{s} \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{e} \\ \\ \\ \end{pmatrix}$$

or $\mathbf{c} \leftarrow_{\$} \mathcal{U}\left(\mathbb{Z}_q^m\right)$.

The Learning with Errors Problem (LWE)

Definition (LWE)

Let n, q be positive integers, χ be a probability distribution on $\mathbb{Z}$ and $\mathbf{s}$ be a uniformly random vector in $\mathbb{Z}_q^n$. We denote by $\mathcal{L}_{\mathbf{s},\chi}$ the probability distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{Z}$ according to χ and considering it in $\mathbb{Z}_q$, and returning $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

Decision-LWE is the problem of deciding whether pairs $(\mathbf{a}, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ are sampled according to $\mathcal{L}_{\mathbf{s},\chi}$ or the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Search-LWE is the problem of recovering $\mathbf{s}$ from pairs $(\mathbf{a}, c) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ sampled according to $\mathcal{L}_{\mathbf{s},\chi}$.

A Fair Warning: Gaussian Distributions

In this lecture I am ignoring the specifics of the distribution χ. That is, the only slide with the phrase "Discrete Gaussian distribution" is this slide.

In practice, **for encryption** the shape of the error does not seem to matter much.

Ignoring the distribution allows to brutally simply proof sketches: almost all technical difficulty in these proofs derives from arguing about two distributions being close.

Normal Form LWE

Consider

$A_i \in \mathbb{Z}_q^{n\times n}$, $s \in \mathbb{Z}_q^n$, $e_i \leftarrow\$ \chi^n$,

$c_0 = A_0 \cdot s + e_0$ and

$c_1 = A_1 \cdot s + e_1$

We have with high probability

$$
\begin{aligned}
c' &= c_1 - A_1 \cdot A_0^{-1} \cdot c_0 \\
&= A_1 \cdot s + e_1 - A_1 \cdot A_0^{-1}(A_0 \cdot s + e_0) \\
&= A_1 \cdot s + e_1 - A_1 \cdot s - A_1 \cdot A_0^{-1} \cdot e_0 \\
&= -A_1 \cdot A_0^{-1} \cdot e_0 + e_1 \\
&= A' \cdot e_0 + e_1
\end{aligned}
$$

We might as well assume that our secret is also sampled from χ.

Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. **Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems.** In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Berlin, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35

Dimension/Modulus Trade-Off

Consider $\mathbf{a}, \mathbf{s} \in \mathbb{Z}_q^d$ where $\mathbf{s}$ is small, then

$$q^{d-1} \cdot \langle \mathbf{a}, \mathbf{s} \rangle \approx \left(\sum_{i=0}^{d-1} q^i \cdot a_i \right) \cdot \left(\sum_{i=0}^{d-1} q^{d-i-1} \cdot s_i \right) \bmod q^d = \tilde{a} \cdot \tilde{s} \bmod q^d.$$

If there is an efficient algorithm solving the problem in $\mathbb{Z}_{q^d}$, we can solve the problem in $\mathbb{Z}_q^d$.

Example ($\mathbb{Z}_{q^2}$)

$$q \cdot (a_0 \cdot s_0 + a_1 \cdot s_1) + a_0 \cdot s_1 + q^2 \cdot a_1 \cdot s_0 \bmod q = (a_0 + q \cdot a_1) \cdot (q \cdot s_0 + s_1)$$

Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In: *45th ACM STOC*. ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584. DOI: 10.1145/2488608.2488680

LWE and Lattices

Lattices

A lattice is a discrete subgroup of $\mathbb{R}^d$

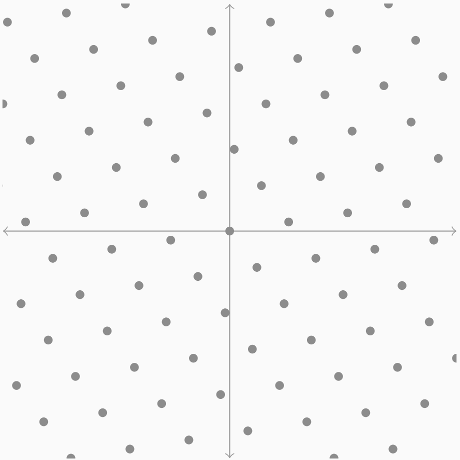
It can be written as

$$\Lambda = \left\{ \sum_{i=0}^{d-1} v_i \cdot \mathbf{b}_i \mid v_i \in \mathbb{Z} \right\}$$

for some basis vectors $\mathbf{b}_i$.

We write $\Lambda(\mathbf{B})$ for the lattices spanned by the columns of $\mathbf{B}$.

A lattice is q-ary if it contains $q\,\mathbb{Z}^d$, e.g. $\{\mathbf{x} \in \mathbb{Z}_q^d \mid \mathbf{x} \cdot \mathbf{A} \equiv \mathbf{0}\}$ for some $\mathbf{A} \in \mathbb{Z}^{d \times d'}$.



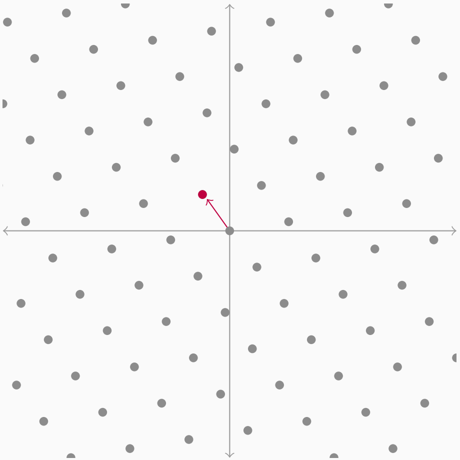
Picture credit: David Wong

Shortest Vector Problem

Definition

Given a lattice basis $\mathbf{B}$, find a shortest non-zero vector in $\Lambda(\mathbf{B})$.

- The most natural problem on lattices
- We write $\lambda_1(\Lambda)$ for the Euclidean norm of a shortest vector.
- NP-hard to solve exactly
- Cryptography relies on approximate variants without such a reduction



Picture credit: David Wong

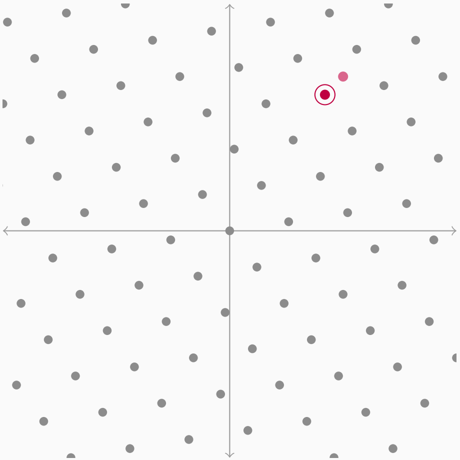
Bounded Distance Decoding

Definition

Given a lattice basis $\mathbf{B}$, a vector $\mathbf{t}$, and a parameter $0 < \alpha$ such that the Euclidean distance $\mathrm{dist}(\mathbf{t}, \mathbf{B}) < \alpha \cdot \lambda_1(\Lambda(\mathbf{B}))$, find the lattice vector $\mathbf{v} \in \Lambda(\mathbf{B})$ which is closest to $\mathbf{t}$.

When $\alpha < 1/2$ unique decoding is guaranteed but for $\alpha < 1$ we typically still expect unique decoding.

BDD is a special case of the Closest Vector Problem where there is no bound on the distance to the lattice.



Picture credit: David Wong

LWE is Bounded Distance Decoding (BDD) on Random q-ary Lattices

Let

$$\mathbf{L} = \begin{pmatrix} q\mathbf{I} & \mathbf{A} \\ 0 & \mathbf{I} \end{pmatrix}$$

We can reformulate the matrix form of the LWE equation $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} \equiv \mathbf{c} \bmod q$ as a linear system over the Integers as:

$$\mathbf{L} \cdot \begin{pmatrix} * \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} \mathbf{e} \\ -\mathbf{s} \end{pmatrix} = \begin{pmatrix} q\mathbf{I} & -\mathbf{A} \\ 0 & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} * \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} \mathbf{e} \\ -\mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix}$$

The vector $(\mathbf{c}^T, \mathbf{0}^T)^T$ is close to the lattice $\Lambda(\mathbf{L})$ with offset $(\mathbf{e}^T, -\mathbf{s}^T)^T$.

Is that a Good Choice?

Maybe BDD on random q-ary lattices is easier than BDD in general?

Maybe BDD is easier than SVP?

Sketch: BDD on Random q-ary Lattices solves BDD on any Lattice

We are given some basis $\mathbf{B} \in \mathbb{Z}^{d \times d}$ and some target $\mathbf{t}$ s.t. $\mathbf{t} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$ with $\mathbf{e}$ small

Pick some large $q \geq 2^{2d}$

Sample some $\mathbf{U}$ (see below)

Set $\mathbf{A} = \mathbf{U} \cdot \mathbf{B} \bmod q$ and consider $\mathbf{c} = \mathbf{U} \cdot \mathbf{t} + \mathbf{e}'$ with $\mathbf{e}'$ small

$$\mathbf{c} = \mathbf{U} \cdot \mathbf{t} + \mathbf{e}' = \mathbf{U} \cdot (\mathbf{B} \cdot \mathbf{s} + \mathbf{e}) + \mathbf{e}' = \mathbf{U} \cdot \mathbf{B} \cdot \mathbf{s} + \mathbf{U} \cdot \mathbf{e} + \mathbf{e}' = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}''$$

We can pick $\mathbf{U}$

- large enough to make $\mathbf{A}$ uniform mod q and
- small enough to make $\mathbf{U} \cdot \mathbf{e} + \mathbf{e}'$ small and well distributed

using “smoothing parameter” arguments on $\Lambda(\mathbf{B}^{-T})$

Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: `http://doi.acm.org/10.1145/1568318.1568324`

Sketch: Solving BDD on any Lattice implies solving GapSVP

Say we want to decide if $\lambda_1(\Lambda) \leq 1$ or $\lambda_1(\Lambda) > \gamma$ and we have a BDD solver with $\alpha = c \cdot \gamma$.

- Pick a random $\mathbf{z} \in \Lambda$, add a small error $\mathbf{e}$ of norm $c \cdot \gamma$
- Run the BDD solver.
- If it returns $\mathbf{z}$ then output $\lambda_1(\Lambda) > \gamma$, else output $\lambda_1(\Lambda) \leq 1$.[1]
- Regev showed: If you have a BDD solver you can find a short basis on a quantum computer[2]

[1] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: *41st ACM STOC*. ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 333–342. DOI: 10.1145/1536414.1536461.

[2] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: http://doi.acm.org/10.1145/1568318.1568324.

Concrete Hardness: Cryptanalysis

This tells us random q-ary lattices are not a terrible choice

To establish how long it actually takes to solve LWE, we rely on cryptanalysis

```
from estimator import *
schemes.Kyber512
```

```
LWEParameters(n=512, q=3329, Xs=D(σ=1.22), Xe=D(σ=1.22), m=512, tag='Kyber 512')
```

```
LWE.primal_usvp(schemes.Kyber512)
```

```
rop: ≈2^143.8, red: ≈2^143.8, δ: 1.003941, β: 406, d: 998, tag: usvp
```

https://github.com/malb/lattice-estimator/

Algebraic Variants

LWE

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} & a_{0,5} & a_{0,6} & a_{0,7} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & a_{1,7} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & a_{2,7} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} & a_{3,6} & a_{3,7} \\ a_{4,0} & a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} & a_{4,6} & a_{4,7} \\ a_{5,0} & a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} & a_{5,5} & a_{5,6} & a_{5,7} \\ a_{6,0} & a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} & a_{6,5} & a_{6,6} & a_{6,7} \\ a_{7,0} & a_{7,1} & a_{7,2} & a_{7,3} & a_{7,4} & a_{7,5} & a_{7,6} & a_{7,7} \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix}$$

Performance

Storage: $\mathcal{O}(n^2)$; Computation $\mathcal{O}(n^2)$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} a_0 & -a_7 & -a_6 & -a_5 & -a_4 & -a_3 & -a_2 & -a_1 \\ a_1 & a_0 & -a_7 & -a_6 & -a_5 & -a_4 & -a_3 & -a_2 \\ a_2 & a_1 & a_0 & -a_7 & -a_6 & -a_5 & -a_4 & -a_3 \\ a_3 & a_2 & a_1 & a_0 & -a_7 & -a_6 & -a_5 & -a_4 \\ a_4 & a_3 & a_2 & a_1 & a_0 & -a_7 & -a_6 & -a_5 \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & -a_7 & -a_6 \\ a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & -a_7 \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{pmatrix} \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix}$$

Ring-LWE/Polynomial-LWE

$$\sum_{i=0}^{n-1} c_i \cdot X^i = \left(\sum_{i=0}^{n-1} a_i \cdot X^i\right) \cdot \left(\sum_{i=0}^{n-1} s_i \cdot X^i\right) + \sum_{i=0}^{8} e_i \cdot X^i \bmod X^n + 1$$

$$c(X) = a(X) \cdot s(X) + e(X) \bmod \phi(X)$$

Performance (n is a power of two)

Storage: $\mathcal{O}(n)$; Computation $\mathcal{O}(n \log n)$

Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Berlin, Heidelberg, Dec. 2009, pp. 617–635. DOI: 10.1007/978-3-642-10366-7_36; Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Berlin, Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1

Module-LWE

$$\begin{pmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{1,0} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \end{pmatrix} = \left(\begin{array}{cccc|cccc} a_{0,0} & -a_{0,3} & -a_{0,2} & -a_{0,1} & a_{1,0} & -a_{1,3} & -a_{1,2} & -a_{1,1} \\ a_{0,1} & a_{0,0} & -a_{0,3} & -a_{0,2} & a_{1,1} & a_{1,0} & -a_{1,3} & -a_{1,2} \\ a_{0,2} & a_{0,1} & a_{0,0} & -a_{0,3} & a_{1,2} & a_{1,1} & a_{1,0} & -a_{1,3} \\ a_{0,3} & a_{0,2} & a_{0,1} & a_{0,0} & a_{1,3} & a_{1,2} & a_{1,1} & a_{1,0} \\ \hline a_{2,0} & -a_{2,3} & -a_{2,2} & -a_{2,1} & a_{3,0} & -a_{3,3} & -a_{3,2} & -a_{3,1} \\ a_{2,1} & a_{2,0} & -a_{2,3} & -a_{2,2} & a_{3,1} & a_{3,0} & -a_{3,3} & -a_{3,2} \\ a_{2,2} & a_{2,1} & a_{2,0} & -a_{2,3} & a_{3,2} & a_{3,1} & a_{3,0} & -a_{3,3} \\ a_{2,3} & a_{2,2} & a_{2,1} & a_{2,0} & a_{3,3} & a_{3,2} & a_{3,1} & a_{3,0} \end{array}\right) \cdot \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{pmatrix}$$

Module-LWE

$$\begin{pmatrix} c_0(X) \\ c_1(X) \end{pmatrix} = \begin{pmatrix} a_0(X) & a_1(X) \\ a_2(X) & a_3(X) \end{pmatrix} \cdot \begin{pmatrix} s_0(X) \\ s_1(X) \end{pmatrix} + \begin{pmatrix} e_0(X) \\ e_1(X) \end{pmatrix}$$

Performance (n is a power of two)

Storage: $\mathcal{O}(k^2 \cdot n)$; Computation $\mathcal{O}(k^2 \cdot n \log n)$

Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. In: *Designs, Codes, and Cryptography* 75.3 (June 2015), pp. 565–599. ISSN: 0925-1022 (print), 1573-7586 (electronic). DOI: http://dx.doi.org/10.1007/s10623-014-9938-4. URL: http://link.springer.com/article/10.1007/s10623-014-9938-4

LWE Encryption

Convention

I am going to use the Ring-LWE formulation

$$c_i(X) = a_i(X) \cdot s(X) + e_i(X)$$

Thus, each sample corresponds to "n LWE samples"

I will suppress the "(X)" in "$a(X)$" etc.

I will assume s is "small" and that the product of two "small" things is "small".

I will write e_i to emphasise that e_i is small.

TL;DR: I will write

$$c_i = a_i \cdot s + e_i$$

DH to Ring-LWE Dictionary

DH Land	Ring-LWE Land
g	a
g^x	$a \cdot s + e$
$g^x \cdot g^y = g^{x+y}$	$(a \cdot s + e_0) + (a \cdot t + e_1) = a \cdot (s + t) + e'$
$(g^a)^b = (g^b)^a$	$(a \cdot s + e) \cdot t = (a \cdot s \cdot t + e \cdot t)$ $\approx a \cdot s \cdot t \approx (a \cdot t + e) \cdot s$
(g, g^a, g^b, g^{ab}) $\approx_c (g, g^a, g^b, u)$	$(a,\ a \cdot s + e,\ a \cdot t + d,\ a \cdot s \cdot t + e')$ $\approx_c (a,\ a \cdot s + e,\ a \cdot t + d,\ u)$

Regev's Encryption Scheme

You have already seen it.

KeyGen Publish $c_i = a_i \cdot s + e_i$ for $i = 0, \ldots, \lceil 2\, n \log q \rceil$

Encrypt

$$d_0 = \sum b_i \cdot a_i, \quad d_1 = \left(\sum b_i \cdot c_i\right) + \lfloor q/2 \rfloor \cdot m \text{ with } b_i \in \{0,1\}, m \in \{0,1\}^n$$

Decrypt

$$\begin{aligned}
\left\lfloor \frac{2}{q} \cdot (d_1 - d_0 \cdot s) \right\rceil &= \left\lfloor \frac{2}{q} \cdot \left(\left(\sum b_i \cdot c_i\right) + \left\lfloor \frac{q}{2} \right\rfloor \cdot m - \sum b_i \cdot a_i \cdot s \right) \right\rceil \\
&= \left\lfloor \frac{2}{q} \cdot \left(\left(\sum b_i \cdot (a_i \cdot s + e_i)\right) + \frac{q}{2} \cdot m - \sum b_i \cdot a_i \cdot s \right) \right\rceil \\
&= \left\lfloor \frac{2}{q} \cdot \left(\left(\sum b_i \cdot e_i\right) + \left\lfloor \frac{q}{2} \right\rfloor \cdot m \right) \right\rceil = m
\end{aligned}$$

The public key is indistinguishable from uniform by the LWE assumption and $\sum b_i \cdot a_i$ is statistically close to uniformly random by the Leftover Hash Lemma (LHL).

ElGamal

KeyGen $h = g^x$

Encrypt $d_0,\ d_1 = (g^r,\ m \cdot h^r)$ for some random r

Decrypt $d_1/d_0^x = m \cdot (g^x)^r/(g^r)^x = m$

[LPR10]

KeyGen $c = a \cdot s + e$

Encrypt $d_0,\ d_1 = v \cdot a + e',\ v \cdot c + e'' + \left\lfloor \frac{q}{2} \right\rfloor \cdot m$

Decrypt

$$\left\lfloor \frac{2}{q} \cdot (d_1 - d_0 \cdot s) \right\rceil = \left\lfloor \frac{2}{q} \cdot \left(v \cdot (a \cdot s + e) + e'' + \left\lfloor \frac{q}{2} \right\rfloor \cdot m - (v \cdot a + e') \cdot s \right) \right\rceil$$
$$= \left\lfloor \frac{2}{q} \cdot \left(v \cdot e + e'' + \left\lfloor \frac{q}{2} \right\rfloor \cdot m - e' \cdot s \right) \right\rceil = m$$

Proof Sketch

KeyGen $c = a \cdot s + e$

The public key (a, c) is indistinguishable from uniform (u', u'') by the (Ring-)LWE assumption

Encrypt $d_0,\ d_1 = v \cdot a + e',\ v \cdot c + e'' + q/2 \cdot m$

Then $v \cdot u' + e'',\ v \cdot u'' + e''$ is indistinguishable from uniform by the (Ring)-LWE assumption

Reconciliation I

Once you have ElGamal, recovering Diffie-Hellman is straight forward.

Common a

Alice $c_0 = s \cdot a + e_0$

Bob $c_1 = a \cdot t + e_1$

Shared

$$c_0 \cdot t = (s \cdot a + e_0) \cdot t \approx s \cdot a \cdot t \approx s \cdot (a \cdot t + e_1) = s \cdot c_1$$

$$c_0 \cdot t = (s \cdot a + e_0) \cdot t \approx s \cdot a \cdot t \approx s \cdot (a \cdot t + e_1) = s \cdot c_1$$

The problem with this construction is that "$\approx$" $\neq$ "$=$"

Need to send a "hint" how to round correctly (2nd most significant bit)[3]

Cannot have efficient Non-interactive Key Exchange (NIKE) without new ideas[4]

[3] Jintai Ding, Xiang Xie, and Xiaodong Lin. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. Cryptology ePrint Archive, Report 2012/688. 2012. URL: https://eprint.iacr.org/2012/688.

[4] Siyao Guo, Pritish Kamath, Alon Rosen, and Katerina Sotiraki. Limits on the Efficiency of (Ring) LWE-Based Non-interactive Key Exchange. In: *Journal of Cryptology* 35.1 (Jan. 2022), p. 1. DOI: 10.1007/s00145-021-09406-y.

Practical Performance (Zen4)

Curve25519

Key generation	$\approx$ 100,000 cycles
Key agreement	$\approx$ 110,000 cycles
Public key	32 bytes
Key Share	32 bytes

`https://bench.cr.yp.to/results-dh.html`

Kyber-768

Key generation	$\approx$ 30,000 cycles
Encapsulation	$\approx$ 40,000 cycles
Decapsulation	$\approx$ 32,000 cycles
Ciphertext	1,088 bytes
Public key	1,184 bytes

`https://bench.cr.yp.to/results-kem.html`

Interpretation

An Ethernet frame takes 1,500 bytes

Your laptop does about $2 \cdot 10^9$ cycles per second

FIN

… NOISY LINEAR ALGEBRA MOD q

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Berlin, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35.

[BLPRS13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In: *45th ACM STOC*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 575–584. DOI: 10.1145/2488608.2488680.

[DXL12] Jintai Ding, Xiang Xie, and Xiaodong Lin. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. Cryptology ePrint Archive, Report 2012/688. 2012. URL: https://eprint.iacr.org/2012/688.

References II

[GKRS22] Siyao Guo, Pritish Kamath, Alon Rosen, and Katerina Sotiraki. Limits on the Efficiency of (Ring) LWE-Based Non-interactive Key Exchange. In: *Journal of Cryptology* 35.1 (Jan. 2022), p. 1. DOI: 10.1007/s00145-021-09406-y.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Berlin, Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. In: *Designs, Codes, and Cryptography* 75.3 (June 2015), pp. 565–599. ISSN: 0925-1022 (print), 1573-7586 (electronic). DOI: http://dx.doi.org/10.1007/s10623-014-9938-4. URL: http://link.springer.com/article/10.1007/s10623-014-9938-4.

References III

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, 2009, pp. 333–342. DOI: 10.1145/1536414.1536461.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In: *Journal of the ACM* 56.6 (Sept. 2009), 34:1–34:40. ISSN: 0004-5411 (print), 1557-735X (electronic). DOI: http://doi.acm.org/10.1145/1568318.1568324.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Berlin, Heidelberg, Dec. 2009, pp. 617–635. DOI: 10.1007/978-3-642-10366-7_36.