Post-Quantum Cryptography, Lattices and Learning with Errors

A Primer

Martin R. Albrecht

"'Cryptographers seldom sleep well' (Silvio Micali). Their careers are frequently based on very precise complexity-theoretic assumptions, which could be shattered the next morning. **A polynomial time algorithm for factoring would certainly prove more crushing than any paltry fluctuation of the Dow Jones.**" – [Kil88]

Outline

The Poverty of Public-Key Cryptography

Post-Quantum Era

Learning with Errors

Algebraic Variants

LWE and Lattices

LWE Encryption

Other Stuff

The Poverty of Public-Key Cryptography

Cryptographic Primitives

Symmetric Primitives

- Block and stream ciphers (AES, ChaCha20, …)
- Authentication codes (HMAC, Poly1305, …)
- Hash functions (SHA-2, SHA-3, …)

Asymmetric Primitives

- Key agreement and public-key encryption (RSA, DH, ECDH, …)
- Digital signatures (RSA, DSA, ECDSA, …)

The Poverty of Public-Key Cryptography

The Internet runs on factoring and discrete logarithms

Rivest-Shamir-Adleman (RSA)

Factoring

Let p, q be primes of λ bits. Given $n := p \cdot q$, find p.

RSA

Let $n := p \cdot q$ be the product of two λ-bit primes. Let $e \nmid (p-1) \cdot (q-1)$. Given n, e and $c := m^e \bmod n$ for $m \leftarrow\$ \mathbb{Z}_n$, find m.

Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: *Communications of the Association for Computing Machinery* 21.2 (Feb. 1978), pp. 120–126. DOI: 10.1145/359340.359342

- We'd **like** to say that RSA encryption/decryption is based on factoring, but such a reduction is not know
  - If factoring is easy then RSA is insecure
  - RSA could be insecure and factoring hard[a]
- Rabin encryption ($\approx$ RSA with $e = 2$) is based on factoring

[a] … on a classical computer, we'll get to that shortly

Diffie-Hellman (DH)

Discrete Logarithms

Let p be a λ-bit prime and let g be a generator of the multiplicative group $\mathbb{Z}_p^*$. Given $g^a \bmod p$ find a.

DH

Let p be a λ-bit prime and let g be a generator of the multiplicative group $\mathbb{Z}_p^*$. Given $g^a \bmod p$, $g^b \bmod p$ and u, decide if $u = g^{ab}$ or random.

Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638

- We'd like to say that the DH key exchange is based on discrete logarithms, but such a reduction is not known
  - If discrete logs are easy, DH is insecure
  - DH could be easy and discrete logarithms hard[a]

[a] … on a classical computer, we'll get to that shortly

Diffie-Hellman (DH)

- We didn't use any properties of $\mathbb{Z}_p^*$ except that it is a group where discrete logarithms are hard.
- Elliptic curves are also groups where it is believed to be hard to compute discrete logarithms[1]
  - Indeed, it is believed only generic algorithms apply, in contrast to $\mathbb{Z}_p^* \Rightarrow$ much smaller parameters
- Elliptic curves are usually written additively and not multiplicative.

[1] on a classical computer ...

Diffie-Hellman (DH)

Discrete Logarithms

Let p be prime and let g be a generator of the multiplicative group $\mathbb{Z}_p^*$. Given $g^a \bmod p$ find a.

DH

Let p be prime and let g be a generator of the multiplicative group $\mathbb{Z}_p^*$. Given $g^a \bmod p$, $g^b \bmod p$ and u, decide if $u = g^{ab}$ or random.

Discrete Logarithms

Let $\mathcal{G}$ be a group of order p and let G be a generator of $\mathcal{G}$. Given $a \cdot G$ for $a \in \mathbb{Z}_p$ find a.[a]

DH

Let $\mathcal{G}$ be a group of order p and let G be a generator of $\mathcal{G}$. Given $(G, a \cdot G, b \cdot G, U)$ for $a, b \in \mathbb{Z}_p$ decide if $U = a \cdot b \cdot G$ or random in $\mathcal{G}$.

[a] Here, $a \cdot G$ means to add G to itself a times.

Post-Quantum Era

Quantum Computers

- A quantum computer makes use of quantum effects (superpositions and entanglement) to perform computations.
- Quantum computers are not **faster** than classical computers, they are **different**.
- Some computations are easy on a quantum computer that are – as far as we know – hard on a classical computer.

- Small universal quantum computers exist.
- Key challenge is to scale them up by making them more stable.
- There is a critical point where we can scale up further using error correction.


FINANCIAL TIMES

HOME WORLD UK COMPANIES TECH MARKETS GRAPHICS OPINION WORK & CAREERS LIFE & ARTS HOW TO SPEND IT

Latest on Quantum technologies

A quantum future could be bright if managed well

Rivals rubbish Google's claim of quantum supremacy

Quantum computing builds on Turing's legacy

Quantum technologies + Add to myFT

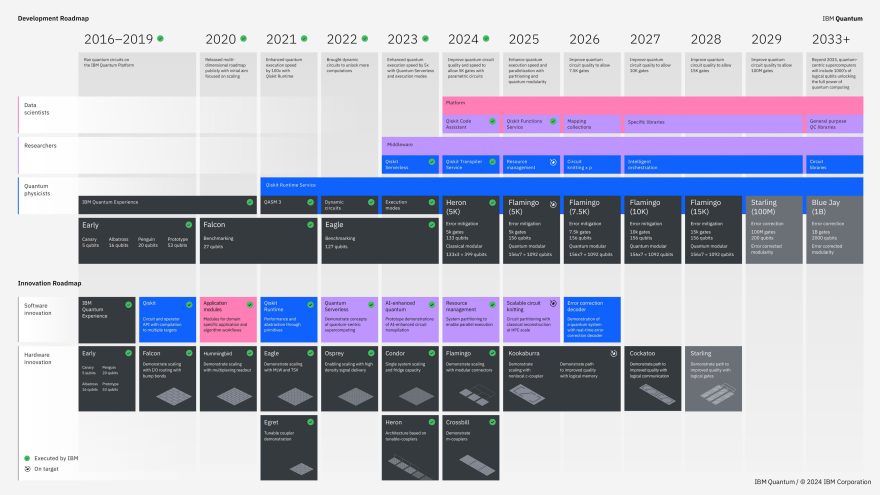
Google claims to have reached quantum supremacy

Researchers say their quantum computer has calculated an impossible problem for ordinary machines

Madhumita Murgia and Richard Waters SEPTEMBER 20 2019 162

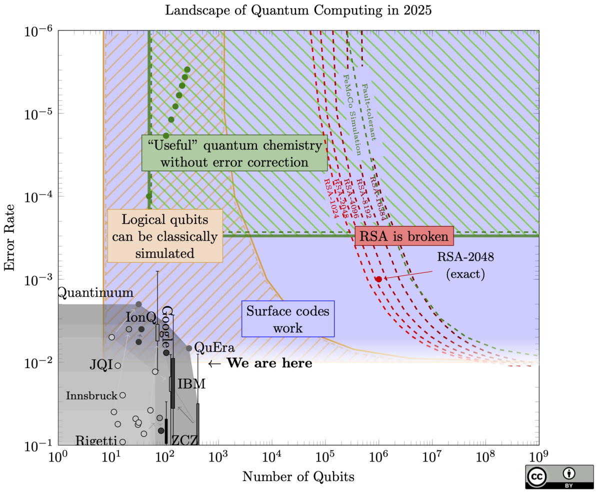
Google claims to have built the first quantum computer that can carry out calculations beyond the ability of today's most powerful supercomputers, a landmark moment that has been hotly anticipated by researchers.

IBM Quantum Computing Timeline



https://www.ibm.com/quantum/roadmap

Landscape of Quantum Computing in 2025



https://sam-jaques.appspot.com/quantum_landscape_2025

Symmetric Primitives: Quantum Computing Perspective (Good News)

Best known quantum algorithms for attacking symmetric cryptography are based on Grover's algorithm.

- Search key space of size 2^n in $2^{n/2}$ operations: AES-256 $\rightarrow$ 128 "quantum bits of security".
- Taking all costs into account: $> 2^{152}$ classical operations for AES-256.[2]
- Assuming a max depth of 2^{96} for a quantum circuit: overall AES-256 cost is $\approx 2^{190}$.
- Does not parallelise: have to wait for 2^X steps, cannot buy 2^{32} quantum computers and wait $2^X/2^{32}$ steps.

[2] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In: *EUROCRYPT 2020, Part II*. ed. by Anne Canteaut and Yuval Ishai. Vol. 12106. LNCS. Springer, Cham, May 2020, pp. 280–310. DOI: `10.1007/978-3-030-45724-2_10`.

“BASED ON VERY PRECISE COMPLEXITY-THEORETIC ASSUMPTIONS, WHICH COULD BE SHATTERED THE NEXT MORNING”

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

"BASED ON VERY PRECISE COMPLEXITY-THEORETIC ASSUMPTIONS, WHICH COULD BE SHATTERED THE NEXT MORNING"

Post-Quantum Standardisation of Primitives

NIST	**Post Quantum ~~Competition~~ Process**
ETSI	Cyber Working Group for Quantum Safe Cryptography
ISO	WG2 Standing Document 8 (SD8): Survey
IETF	Standardisation of **stateful** hash-based signatures, nothing further
CSA	Quantum-safe Security Working Group: position papers
NIST	**Post Quantum Process: Digital Signatures**

Post-Quantum Standardisation of Primitives

NIST	**Post Quantum ~~Competition~~ Process**
ETSI	Cyber Working Group for Quantum Safe Cryptography
ISO	WG2 Standing Document 8 (SD8): Survey
IETF	Standardisation of **stateful** hash-based signatures, nothing further
CSA	Quantum-safe Security Working Group: position papers
NIST	**Post Quantum Process: Digital Signatures**

Bottom Line

Essentially, everyone was waiting for NIST.

NIST PQC ~~Competition~~ Process

Timeline

Submission	November 2017
Round 2 Selection	January 2019
Round 3 Selection	July 2020
Winners and Round 4 Selection	July 2022
3/4 Final Standards	August 2024
Additional KEM Standard	March 2025
Final Standard for Falcon	???

"Key Establishment"/Key Encapsulation

- `(pk,sk) ← KeyGen()`
- `(c,k) ← Encap(pk)`
- `k ← Decap(c,sk)`

Digital Signature

- `(vk,sk) ← KeyGen()`
- `s ← Sig(m,sk)`
- `{0,1} ← Verify(vk,s,m)`

NIST PQC Outcome

NIST selected:

Kyber	A lattice-based KEM (MLWE Problem)
Dilithium	A lattice-based signature scheme (MSIS/MLWE Problems)
Falcon	A lattice-based signature scheme (NTRU Problem)
SPHINCS+	A hash-based signature scheme
HQC	A code-based KEM (decoding random quasi-cyclic codes)

Learning with Errors

"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.

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

A Fair Warning: Gaussian Distributions

- In this talk 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 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

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 $\mathcal{O}(n^2)$

Ring-LWE/Polynomial-LWE

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

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

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

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

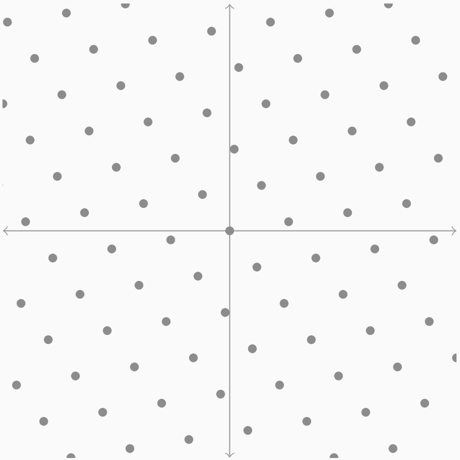
LWE and Lattices

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 $\mathbf{A} \in \mathbb{Z}^{d \times d'}$.



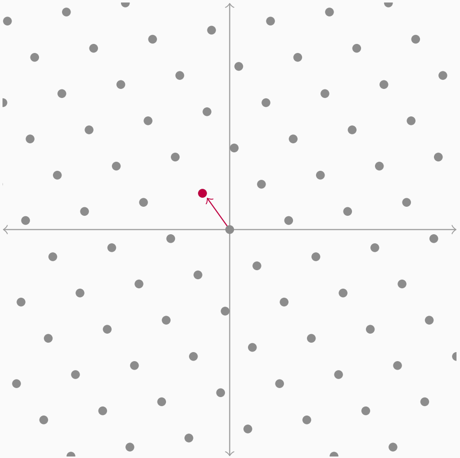
Picture credit: David Wong

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



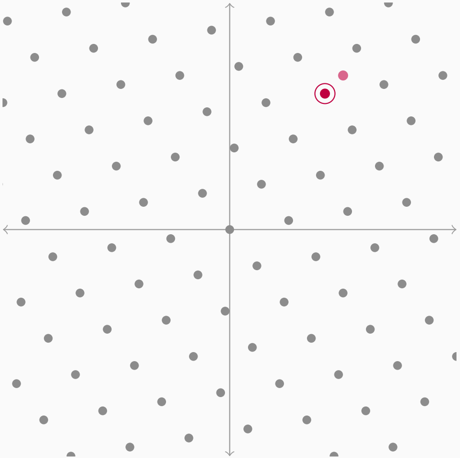
Picture credit: David Wong

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 $\lambda_1(\Lambda) \leq 1$.[3]

Regev showed: If you have a BDD solver you can find a short basis on a quantum computer.[4]

[3] 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.

[4] 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/

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 & LPR10

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

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

Fin

... NOISY LINEAR ALGEBRA MOD q

References I

[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.

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[JNRV20] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In: *EUROCRYPT 2020, Part II*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12106. LNCS. Springer, Cham, May 2020, pp. 280–310. DOI: 10.1007/978-3-030-45724-2_10.

[Kil88] Joe Kilian. Founding Cryptography on Oblivious Transfer. In: *20th ACM STOC*. ACM Press, May 1988, pp. 20–31. DOI: 10.1145/62212.62215.

[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`.

[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`.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: *Communications of the Association for Computing Machinery* 21.2 (Feb. 1978), pp. 120–126. DOI: `10.1145/359340.359342`.

[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`.

Other Stuff

QKD?

"Given the specialised hardware requirements of QKD over classical cryptographic key agreement mechanisms and the requirement for authentication in all use cases, the NCSC does not endorse the use of QKD for any government or military applications, and cautions against sole reliance on QKD for business-critical networks, especially in Critical National Infrastructure sectors. [...] NCSC advice is that the best mitigation against the threat of quantum computers is quantum-safe cryptography."[5]

[5] https://www.ncsc.gov.uk/whitepaper/quantum-security-technologies