Lattice based cryptography

Abderrahmane Nitaj

University of Caen Basse Normandie, France

LMNO

Kuala Lumpur, Malaysia, June 23, 2014

ماليزيا

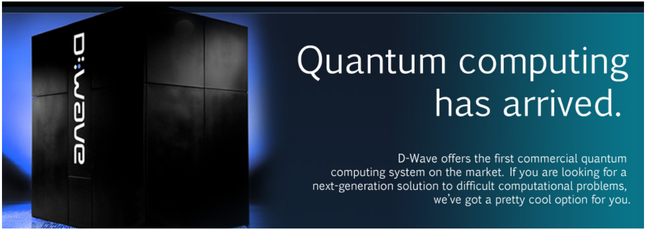
Contents

Contents

Most known public key cryptosystems

- The RSA cryptosystem, 1978: based on factorization.
- The Diffie-Hellman key exchange protocol, 1976: based on the discrete logarithm problem.
- The El Gamal Cryptosystem, 1985: based on the discrete logarithm problem.
- The elliptic curve cryptosystems and protocols, 1985: based on elliptic curves.
- The NTRU cryptosystem, 1996: based on lattice hard problems.
- The Learner with error cryptosystem, 2005: based on lattice hard problems.

Most known public key cryptosystems



Vulnerability to quantum computers

- The RSA cryptosystem: vulnerable.
- The Diffie-Hellman key exchange protocol: vulnerable.
- The El Gamal Cryptosystem: vulnerable.
- The elliptic curve cryptosystems and protocols: vulnerable.
- NTRU and LWE cryptosystems: still resistant (post quantum cryptography).

Contents

Introduction to lattices

Definition

Let n and d be two positive integers. Let $b_1 \cdots, b_d \in \mathbb{R}^n$ be d linearly independent vectors. The lattice $\mathcal{L}$ generated by $(b_1 \cdots, b_d)$ is the set

$$\mathcal{L} = \sum_{i=1}^{d} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{d} x_i b_i \mid x_i \in \mathbb{Z} \right\}.$$

The vectors $b_1 \cdots, b_d$ are called a vector basis of $\mathcal{L}$. The lattice rank is n and the lattice dimension is d. If $n = d$ then $\mathcal{L}$ is called a full rank lattice.

Introduction to lattices

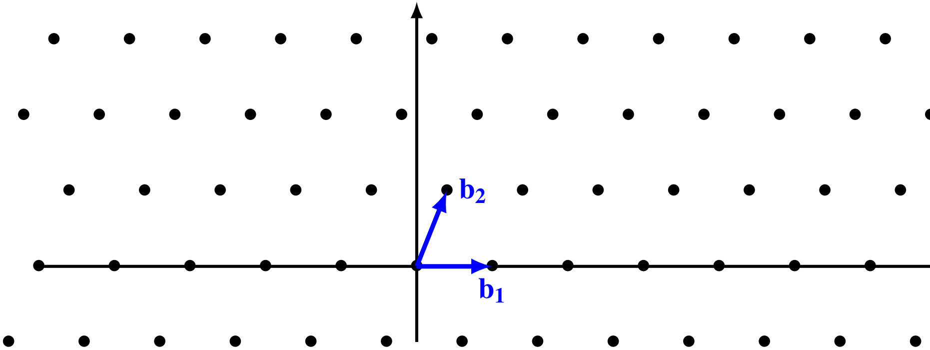


Figure: A lattice with the basis (b_1, b_2)

Introduction to lattices

Theorem

Let $\mathcal{L}$ be a lattice of dimension d and rank n. Then $\mathcal{L}$ can be written as the rows of an $n \times d$ matrix with real entries.

Let

$$b_i = \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{ni} \end{bmatrix}.$$

Let $v = \sum_{i=1}^{d} x_i b_i$ for $x_i \in \mathbb{Z}$. Then

$$v = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}.$$

Introduction to lattices

Theorem

Let $\mathcal{L} \subset \mathbb{R}^n$ *be a lattice of dimension* d. *Let* $(b_1 \cdots, b_d)$ *and* $(b'_1 \cdots, b'_d)$ *be two bases of* $\mathcal{L}$. *Then there exists a* $d \times d$ *matrix* U *with entries in* $\mathbb{Z}$ *and* $\det(U) = \pm 1$ *such that*

$$\begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_d \end{bmatrix} = U \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}.$$

Introduction to lattices

Definition

Let $\mathcal{L}$ be a lattice with a basis $(b_1 \cdots, b_d)$. The volume or determinant of $\mathcal{L}$ is

$$\det(\mathcal{L}) = \sqrt{\det\left(BB^t\right)},$$

where B is the $d \times n$ matrix of formed by the rows of the basis.

Theorem

Let $\mathcal{L}$ be a lattice of dimension d. Then the $\det(\mathcal{L})$ is independent of the choice of the basis.

Lemma

Let $\mathcal{L}$ be a full-rank lattice ($n = d$) of dimension n. If $(b_1 \cdots, b_n)$ is a basis of $\mathcal{L}$ with matrix B, then

$$\det(L) = |\det(B)|.$$

Introduction to lattices

Definition

Let $\mathcal{L}$ be a lattice with a basis $(b_1 \cdots, b_d)$. The fundamental domain or parallelepipede for $\mathcal{L}$ is the set

$$\mathcal{P}(b_1 \cdots, b_d) = \left\{ \sum_{i=1}^{d} x_i b_i, \mid 0 \leq x_i < 1 \right\}.$$

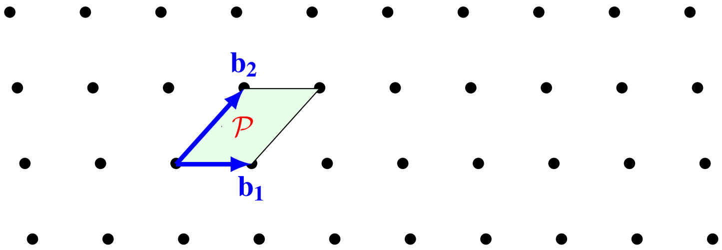


Figure: The fundamental domain for the basis (b_1, b_2)

Introduction to lattices

Theorem

Let $\mathcal{L}$ be a lattice with a basis $(b_1, \ldots, b_d)$. Then the volume $\mathcal{V}$ of the fundamental domain $\mathcal{P}(b_1, \ldots, b_d)$ satisfies

$$\mathcal{V}(\mathcal{P}(b_1, \ldots, b_d)) = \det(\mathcal{L}).$$

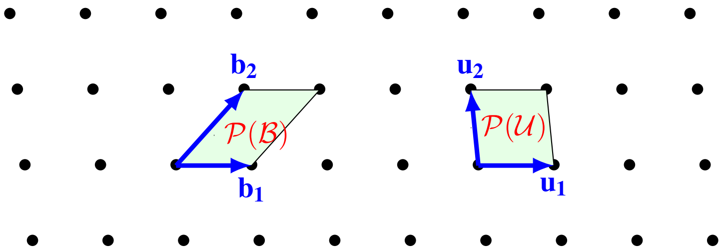


Figure: The fundamental domain for the bases (b_1, b_2) and (u_1, u_2)

Introduction to lattices

Definition

Let $u = (u_1, \cdots, u_n)$ and $v = (v_1 \cdots, v_n)$ be two vectors of $\mathbb{R}^n$.

1. The inner product of u and v is
$$\langle u, v\rangle = u^T v = \sum_{i=1}^{n} u_i v_i.$$
2. The Euclidean norm of u is
$$\|u\| = (\langle u, u\rangle)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} u_i^2\right)^{\frac{1}{2}}.$$

Introduction to lattices

Definition

Let L be a lattice. The minimal distance λ_1 of $\mathcal{L}$ is the length of the shortest nonzero vector of $\mathcal{L}$:

$$\lambda_1 = \inf\{\|v\| \in \mathcal{L} \mid v \in \mathcal{L}\backslash\{0\}\} = \inf\{\|v-u\| \in \mathcal{L} \mid v, u \in \mathcal{L},\ v \neq u\}.$$

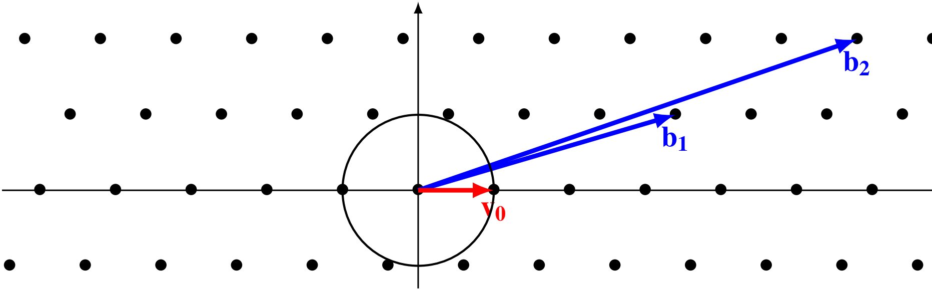


Figure: The shortest vectors are v_0 and $-v_0$

Introduction to lattices

Example

Let $\mathcal{L}$ be a lattice with a basis (b_1, b_2) with

$$b_1 = \begin{bmatrix} 19239 \\ 2971 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 22961 \\ 3546 \end{bmatrix}.$$

Find the shortest vector.

The shortest vector is in the form

$$v_0 = x_1 b_1 + x_2 b_2 = \begin{bmatrix} 19239x_1 + 22961x_2 \\ 2971x_1 + 3546x_2 \end{bmatrix},$$

for some integers $(x_1, x_2) \neq (0, 0)$.
One can show that $v_0 = 37b_1 - 31b_2$ is the shortest vector in the lattice $\mathcal{L}$.

Introduction to lattices

Example

Let $\mathcal{L}$ be a lattice with a basis (b_1, b_2, b_3) with

$$b_1 = \begin{bmatrix} 124797 \\ 2971 \\ 4781 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 95874 \\ 3546 \\ 7895 \end{bmatrix}, \quad b_3 = \begin{bmatrix} 56871 \\ 35462 \\ 16539 \end{bmatrix}.$$

Find the shortest vector in the lattice

The shortest vector is in the form

$$v_0 = x_1 b_1 + x_2 b_2 + x_3 b_3 = \begin{bmatrix} 124797x_1 + 95874x_2 + 56871x_3 \\ 2971x_1 + 3546x_2 + 35462x_3 \\ 4781x_1 + 7895x_2 + 16539x_3 \end{bmatrix},$$

for some integers $(x_1, x_2, x_3) \neq (0, 0, 0)$ for which the norm $\|v_0\|$ is as small as possible. Using the LLL algorithm, we can find that the shortest vector is $v_0 = -3b_1 + 4b_2$.

Introduction to lattices

Definition

Let L be a lattice of dimension n. For $i = 1, \ldots n$, the ith successive minimum of the lattice is

$$\lambda_i = \min\{\max\{\|v_1\|, \ldots, \|v_i\|\} \mid v_1, \ldots, v_i \in \mathcal{L} \text{ are linearly independent}\}.$$

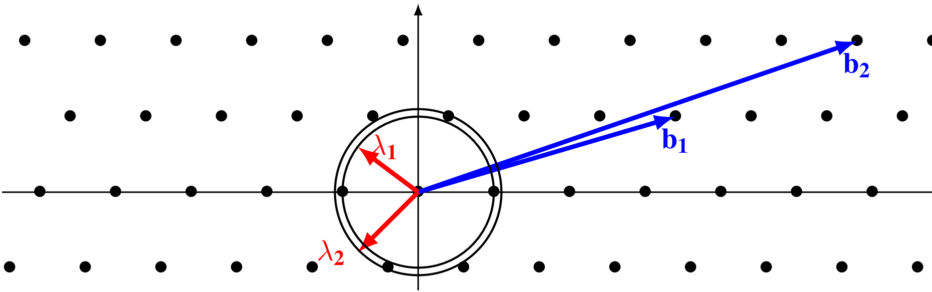


Figure: The first minima λ_1 and the second minima λ_2

Introduction to lattices

Definition

Let $\mathcal{L}$ be a full rank lattice of dimension n in $\mathbb{Z}^n$.

1. **The Shortest Vector Problem (SVP):** Given a basis matrix B for $\mathcal{L}$, compute a non-zero vector $v \in \mathcal{L}$ such that $\|v\|$ is minimal, that is $\|v\| = \lambda_1(\mathcal{L})$.
2. **The Closest Vector Problem (CVP):** Given a basis matrix B for $\mathcal{L}$ and a vector $v \notin \mathcal{L}$, find a vector $u \in \mathcal{L}$ such that $\|v - u\|$ is minimal, that is $\|v - u\| = \mathrm{d}(v, \mathcal{L})$ where $\mathrm{d}(v, \mathcal{L}) = \min_{u \in \mathcal{L}} \|v - u\|$.

Introduction to lattices

Definition

Let $\mathcal{L}$ be a full rank lattice of dimension n in $\mathbb{Z}^n$.

1. **The Shortest Independent Vectors Problem (SIVP):** Given a basis matrix B for $\mathcal{L}$, find n linearly independent lattice vectors $v_1, v_2, \ldots, v_n$ such that $\max_i \|v_i\| \leq \lambda_n$, where λ_n is the nth successive minima of $\mathcal{L}$.
2. **The approximate SVP problem (γSVP):** Fix $\gamma > 1$. Given a basis matrix B for $\mathcal{L}$, compute a non-zero vector $v \in \mathcal{L}$ such that $\|v\| \leq \gamma\lambda_1(\mathcal{L})$ where $\lambda_1(\mathcal{L})$ is the minimal Euclidean norm in $\mathcal{L}$.
3. **The approximate CVP problem (γSVP):** Fix $\gamma > 1$. Given a basis matrix B for $\mathcal{L}$ and a vector $v \notin \mathcal{L}$, find a vector $u \in \mathcal{L}$ such that $\|v - u\| \leq \gamma\lambda_1 \mathrm{d}(v, \mathcal{L})$ where $\mathrm{d}(v, \mathcal{L}) = \min_{u \in \mathcal{L}} \|v - u\|$.

Introduction to lattices

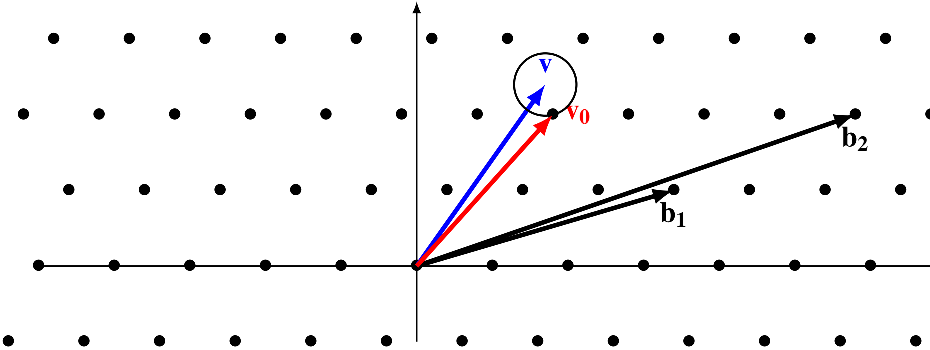


Figure: The closest vector to v is v_0

Introduction to lattices

Theorem (Minkowski)

Let $\mathcal{L}$ be a lattice with dimension n. Then there exists a nonzero vector $v \in \mathcal{L}$ satisfying

$$\|v\| \leq \sqrt{n}\,\det(L)^{\frac{1}{n}}.$$

The Gaussian Heuristic implies that the expected shortest non-zero vector in a lattice $\mathcal{L}$ is approximately $\sigma(\mathcal{L})$ where

$$\sigma(\mathcal{L}) = \sqrt{\frac{n}{2\pi e}}\det(\mathcal{L})^{\frac{1}{n}}.$$

Contents

The LLL algorithm

- Invented in 1982 by Lenstra, Lenstra and Lovász.
- Given an arbitrary basis B of a lattice $\mathcal{L}$, finds a “good” basis.
- Polynomial time algorithm.
- Various applications:
  1. Formulae for π, $\log 2$, ...
  2. Implemented in Mathematica, Maple, Magma, Pari/GP, ...
  3. Solving diophantine equations.
  4. Solving SVP and CVP problems in low dimensions.
  5. Cryptanalysis of Knapsack cryptosystems.
  6. Attacks on RSA and NTRU.

The LLL algorithm

Gram-Schmidt orthogonalization method

Theorem

Let V *be a vector space of dimension* n *and* $(b_1 \cdots, b_n)$ *a basis of* V*. Let* $(b_1^* \cdots, b_n^*)$ *be* n *vectors such that*

$$b_1^* = b_1, \quad b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*,$$

where, for $j < i$

$$\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}.$$

Then $(b_1^* \cdots, b_n^*)$ *is an orthogonal basis of* V*.*

The LLL algorithm

Gram-Schmidt orthogonalization method: $n = 2$

$$b_1^* = b_1, \quad b_2^* = b_2 - \frac{\langle b_2, b_1\rangle}{\langle b_1, b_1\rangle} b_1,$$

$$\Rightarrow \langle b_1^*, b_2^*\rangle = \langle b_1, b_2\rangle - \frac{\langle b_2, b_1\rangle}{\langle b_1, b_1\rangle}\langle b_1, b_1\rangle = 0.$$

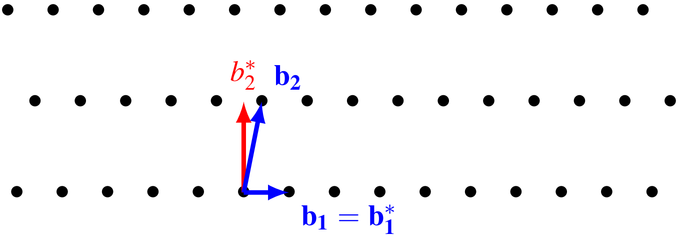


Figure: An orthogonal basis

The LLL algorithm

Gram-Schmidt orthogonalization method: the determinant

Corollary (Hadamard)

Let $B = \{b_1, \ldots, b_n\}$ *be a basis of a lattice* $\mathcal{L}$ *and let* $B^* = \{b_1^*, \ldots, b_n^*\}$ *be the associated Gram-Schmidt basis. Then*

$$\det(\mathcal{L}) = \prod_{i=1}^{n} \|b_i^*\| \leq \prod_{i=1}^{n} \|b_i\|.$$

The LLL algorithm

LLL-reduced basis

Definition

Let $\mathcal{L}$ be a lattice. A basis $(b_1 \cdots, b_n)$ of $\mathcal{L}$ is LLL-reduced if the orthogonal Gram-Schmidt basis $(b_1^* \cdots, b_n^*)$ satisfies

$$|\mu_{i,j}| \leq \frac{1}{2}, \quad \text{pour} \quad 1 \leq j < i \leq n, \tag{1}$$

$$\frac{3}{4}\|b_{i-1}^*\|^2 \leq \|b_i^* + \mu_{i,i-1}b_{i-1}^*\|^2, \quad \text{pour} \quad 1 < i \leq n, \tag{2}$$

where, for $j < i$

$$\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}.$$

The LLL algorithm

LLL-reduced basis: dimension 2

$$|\mu_{2,1}| = \left|\frac{\langle b_2, b_1^*\rangle}{\langle b_1^*, b_1^*\rangle}\right| \leq \frac{1}{2},$$
$$\frac{3}{4}\|b_1\|^2 \leq \|b_2\|^2.$$

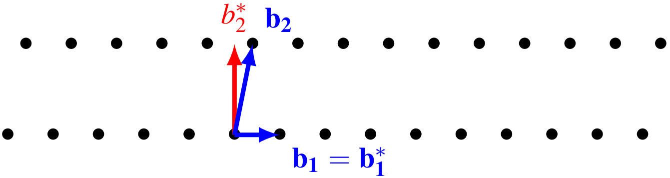


Figure: A 2-dimension reduced basis

The LLL algorithm

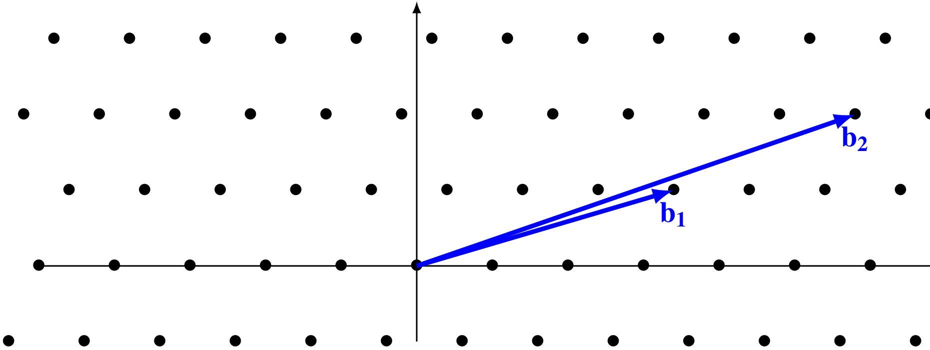


Figure: A lattice with *a bad* basis (b_1, b_2)

The LLL algorithm

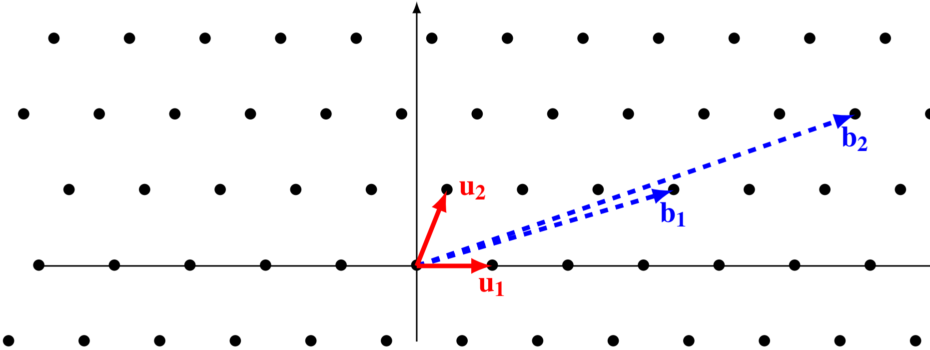


Figure: The same lattice with *a good* basis (u_1, u_2)

The LLL algorithm

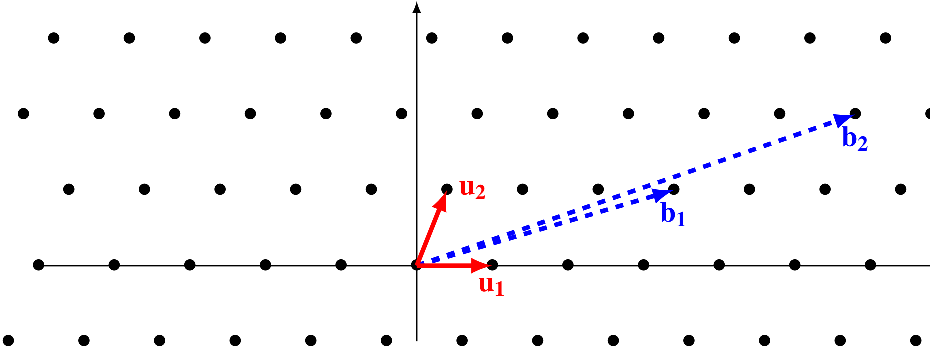


Figure: The same lattice with *a good* basis (u_1, u_2)

The LLL algorithm

LLL-reduced basis: properties

Theorem

Let $(b_1 \cdots, b_n)$ *be an LLL-reduced basis and* $(b_1^*, \cdots, b_n^*)$ *be the Gram-Schmidt orthogonal associated basis. We have*

1. $\|b_j^*\|^2 \leq 2^{i-j}\|b_i^*\|^2$ *for* $1 \leq j \leq i \leq n$.
2. $\prod_{i=1}^n \|b_i\| \leq 2^{\frac{n(n-1)}{4}} \det(L)$.
3. $\|b_j\| \leq 2^{\frac{i-1}{2}} \|b_i^*\|$ *for* $1 \leq j \leq i \leq n$.
4. $\|b_1\| \leq 2^{\frac{n-1}{4}} \det(L)^{\frac{1}{n}}$.
5. *For any nonzero vector* $v \in L$, $\|b_1\| \leq 2^{\frac{n-1}{2}} \|v\|$.

Comparison

- The LLL algorithm: $\|b_1\| \leq 2^{\frac{n-1}{4}} \det(L)^{\frac{1}{n}}$.
- Minkowski: $\|v\| \leq \sqrt{n} \det(L)^{\frac{1}{n}}$.

Contents

NTRU

NTRU

- Invented by Hoffstein, Pipher et Silverman in 1996.
- Security based on the Shortest Vector Problem (SVP).
- Various versions between 1996 and 2001.

Definition

The Shortest Vector Problem (SVP): Given a basis matrix B for $\mathcal{L}$, compute a non-zero vector $v \in \mathcal{L}$ such that $\|v\|$ is minimal, that is $\|v\| = \lambda_1(\mathcal{L})$.

NTRU: Ring of Convolution $\Pi = \mathbb{Z}[X]/(X^N - 1)$

Polynomials

$f = \sum_{i=0}^{N-1} f_i X^i, \qquad g = \sum_{i=0}^{N-1} g_i X^i,$

Sum

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Product

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ with

$$h_k = \sum_{i+j \equiv k \pmod{N}} f_i g_j.$$

NTRU: Ring of Convolution $\Pi = \mathbb{Z}[X]/(X^N - 1)$

Polynomials

$f = \sum_{i=0}^{N-1} f_i X^i, \qquad g = \sum_{i=0}^{N-1} g_i X^i,$

Sum

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Product

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ with

$$h_k = \sum_{i+j \equiv k \pmod N} f_i g_j.$$

NTRU: Ring of Convolution $\Pi = \mathbb{Z}[X]/(X^N - 1)$

Polynomials

$f = \sum_{i=0}^{N-1} f_i X^i, \qquad g = \sum_{i=0}^{N-1} g_i X^i,$

Sum

$f + g = (f_0 + g_0, f_1 + g_1, \cdots, f_{N-1} + g_{N-1}).$

Product

$f * g = h = (h_0, h_1, \cdots, h_{N-1})$ with

$$h_k = \sum_{i+j \equiv k \pmod{N}} f_i g_j.$$

NTRU: Ring of Convolution $\Pi = \mathbb{Z}[X]/(X^N - 1)$

Convolution

$$\underbrace{f = (f_0, f_1, \cdots, f_{N-1}), \qquad g = (g_0, g_1, \cdots, g_{N-1})}_{f * g = h = (h_0, h_1, \cdots, h_{N-1})}.$$

	1	X	$\cdots$	X^k	$\cdots$	X^{N-1}
	$f_0 g_0$	$f_0 g_1$	$\cdots$	$f_0 g_k$	$\cdots$	$f_0 g_{N-1}$
+	$f_1 g_{N-1}$	$f_1 g_0$	$\cdots$	$f_1 g_{k-1}$	$\cdots$	$f_1 g_{N-2}$
+	$f_2 g_{N-2}$	$f_2 g_{N-1}$	$\cdots$	$f_2 g_{k-2}$	$\cdots$	$f_2 g_{N-3}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\cdots$	$\vdots$	$\vdots$
+	$f_{N-2} g_2$	$f_{N-2} g_3$	$\cdots$	$f_{N-2} g_{k+2}$	$\cdots$	$f_{N-2} g_1$
+	$f_{N-1} g_1$	$f_{N-1} g_2$	$\cdots$	$f_{N-1} g_{k+1}$	$\cdots$	$f_{N-1} g_0$
$h =$	h_0	h_1	$\cdots$	h_k	$\cdots$	h_{N-1}

NTRU Parameters

- $N =$ a prime number (e.g. $N = 167,\ 251,\ 347,\ 503$).
- $q =$ a large modulus (e.g. $q = 128,\ 256$).
- $p =$ a small modulus (e.g. $p = 3$).

NTRU Algorithms

Key Generation:

- Randomly choose two private polynomials f and g.
- Compute the inverse of f modulo q: $f * f_q = 1 \pmod{q}$.
- Compute the inverse of f modulo p: $f * f_p = 1 \pmod{p}$.
- Compute the public key $h = f_q * g \pmod{q}$.

NTRU Algorithms

Encryption:

- m is a plaintext in the form of a polynomial mod q.
- Randomly choose a private polynomial r.
- Compute the encrypted message $e = m + pr * h \pmod{q}$.

Decryption:

- Compute $a = f * e = f * (m + pr * h) = f * m + pr * g \pmod{q}$.
- Compute $a * f_p = (f * m + pr * g) * f_p = m \pmod{p}$.

NTRU Algorithms

Encryption:

- m is a plaintext in the form of a polynomial mod q.
- Randomly choose a private polynomial r.
- Compute the encrypted message $e = m + pr * h \pmod{q}$.

Decryption:

- Compute $a = f * e = f * (m + pr * h) = f * m + pr * g \pmod{q}$.
- Compute $a * f_p = (f * m + pr * g) * f_p = m \pmod{p}$.

NTRU Algorithms

Encryption:

- m is a plaintext in the form of a polynomial mod q.
- Randomly choose a private polynomial r.
- Compute the encrypted message $e = m + pr * h \pmod{q}$.

Decryption:

- Compute $a = f * e = f * (m + pr * h) = f * m + pr * g \pmod{q}$.
- Compute $a * f_p = (f * m + pr * g) * f_p = m \pmod{p}$.

NTRU

Correctness of decryption

We have

$$\begin{aligned} a &\equiv f * e \pmod q \\ a &\equiv f * (p * r * h + m) \pmod q \\ a &\equiv f * r * (p * g * f_q) + f * m \pmod q \\ a &\equiv p * r * g * f * f_q + f * m \pmod q \\ a &\equiv p * r * g + f * m \pmod q. \end{aligned}$$

If $p * r * g + f * m \in \left[-\frac{q}{2}, \frac{q}{2}\right]$, then

$$m \equiv a * f_p \mod p.$$

Contents

Learning With Errors

LWE

- Invented by O. Regev in 2005.
- Security based on the GapSVP problem.
- Provable Security.

Definition

The GapSVP problem: Let $\mathcal{L}$ be a lattice with a basis B. Let $\lambda_1(\mathcal{L})$ be the length of the shortest nonzero vector of $\mathcal{L}$. Let $\gamma \in \mathbb{R}^+$. Decide whether $\lambda_1(\mathcal{L}) < 1$ or $\lambda_1(\mathcal{L}) > \gamma$.

Learning With Errors

LWE Key Generation

- **Input:** Integers n, m, l, q.
- **Output:** A private key S and a public key (A, P).

1. Choose $S \in \mathbb{Z}_q^{n\times l}$ at random.
2. Choose $A \in \mathbb{Z}_q^{m\times n}$ at random.
3. Choose $E \in \mathbb{Z}_q^{m\times l}$ according to a Gaussian character χ.
4. Compute $P = AS + E \pmod{q}$. Hence $P \in \mathbb{Z}_q^{m\times l}$.
5. The private key is S.
6. The public key is (A, P).

Learning With Errors

LWE Encryption

- **Input:** Integers n, m, l, t, r, q, a public key (A, P) and a plaintext $M \in \mathbb{Z}_t^{l\times 1}$.
- **Output:** A ciphertext (u, c).

1. Choose $a \in [-r, r]^{m\times 1}$ at random.
2. Compute $u = A^T a \pmod q \in \mathbb{Z}_q^{n\times 1}$.
3. Compute $c = P^T a + \left[\frac{Mq}{t}\right] \pmod q \in \mathbb{Z}_q^{l\times 1}$.
4. The ciphertext is (u, c).

Learning With Errors

LWE Decryption

- **Input:** Integers n, m, l, t, r, q, a private key S and a ciphertext (u, c).
- **Output:** A plaintext M.

1. Compute $v = c - S^T u$ and $M = \left\lfloor \frac{tv}{q} \right\rceil$.

Learning With Errors

Correctness of decryption

We have

$$\begin{aligned} v &= c - S^T u \\ &= (AS+E)^T a - S^T A^T a + \left[\frac{Mq}{t}\right] \\ &= E^T a + \left[\frac{Mq}{t}\right]. \end{aligned}$$

Hence

$$\left[\frac{tv}{q}\right] = \left[\frac{tE^T a}{q} + \frac{t}{q}\left[\frac{Mq}{t}\right]\right].$$

With suitable parameters, the term $\frac{tE^T a}{q}$ is negligible. Consequently $\left[\frac{tv}{q}\right] = M$.

Contents

GGH

GGH

- Invented by Goldreich, Goldwasser and Halevi in 1996.
- Security based on the Closest Vector Problem (CVP).
- Brocken by Nguyen in 1999.

Definition (The Closest Vector Problem (CVP))

Given a basis matrix B for $\mathcal{L}$ and a vector $v \notin \mathcal{L}$, compute a vector $v_0 \in \mathcal{L}$ such that $\|v - v_0\|$ is minimal.

Learning With Errors

GGH key generation

- **Input:** A lattice $\mathcal{L}$ of dimension n.
- **Output:** A public key B and a private key A.

1. Find a “good basis” A of $\mathcal{L}$.
2. Find a “bad basis” B of $\mathcal{L}$.
3. Publish B as the public key.
4. Keep A as the secret key.

Learning With Errors

GGH encryption

- **Input:** A lattice $\mathcal{L}$, a parameter $\rho > 0$, a public key B and a plaintext $m \in \mathbb{Z}^n$.
- **Output:** A ciphertext c.

1. Compute $v = mB \in \mathcal{L}$.
2. Choose a small vector $e \in [-\rho, \rho]^n$.
3. The ciphertext is $c = v + e$.

Learning With Errors

GGH decryption

- **Input:** A lattice $\mathcal{L}$, a private key A and a ciphertext c.
- **Output:** A plaintext $m \in \mathbb{Z}^n$.

1. Use an efficient reduction algorithm and the good basis A to find the closest vector $v \in \mathcal{L}$ of the ciphertext c.
2. Compute $m = vB^{-1}$.

Contents

Thank you

Terima kasih