Eigenes „Gesamtbild" CRYPTO1 – Gitterbasierte Verf. (II)

27.8.22 & 31.11.

Öffentl. schlechte Basis → Gemeinsame HNF ⇒ Geheime gute Basis → löst CVP m. Babai (= fast orthogonal! Hadam. ×1) (relativ kurz!)

⇒ „Gibt tatsächlich so ein Verfahren, aber schwierig & n. so durchschlagend."

Sei $(b_1, \dots, b_n)$ die Basis eines Gitters (mit ganzzahligen Koord.) $L \subseteq \mathbb{Z}^n$. (↳ muss nicht sein!)

Orthonormalisierung = wenn zusätzlich normiert (würde)

[1] Gram-Schmidt Orthogonalisierung(sverfahren):

... baut effektiv aus Basis $(b_1, \dots, b_n)$ eine Orthogonalbasis $(b_1^*, \dots, b_n^*)$, nur leider wahrsch. n. mehr im Gitter:

$b_1^* = b_1$

$b_2^* = b_2 - \nu_{2,1} b_1^*$ mit $\nu_{2,1} = \frac{\langle b_2, b_1^* \rangle}{\langle b_1^*, b_1^* \rangle}$ $\Rightarrow \langle b_1^*, b_2^* \rangle = 0$ (orthogonal)

$b_3^* = b_3 - \nu_{3,2} b_2^* - \nu_{3,1} b_1^*$ mit $\nu_{3,2} = \frac{\langle b_3, b_2^* \rangle}{\langle b_2^*, b_2^* \rangle}, \dots$

$b_k^* = b_k - \nu_{k,k-1} b_{k-1}^* - \dots - \nu_{k,2} b_2^* - \nu_{k,1} b_1^*$ mit $\nu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$, $j < i$, $\nu_{j,j} = 1$

$b_n^* = \dots$

↳ Aber was nun, wenn ich $\lceil \nu_{k,j} \rfloor$ runde, um im Gitter zu verbleiben?!:

(merk. VL!) ↳ Bsp.: $b_1 = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$; $b_2 = \begin{pmatrix} 8 \\ 11 \end{pmatrix}$; $\langle b_1, b_2 \rangle = 5 \cdot 8 + 7 \cdot 11 = 117 \neq 0$, d.h. n. orthogonal

$b_1^* = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$; $b_2^* = \begin{pmatrix} 8 \\ 11 \end{pmatrix} - \frac{117}{74} \begin{pmatrix} 5 \\ 7 \end{pmatrix} = \begin{pmatrix} 8 - 585/74 \\ 11 - 819/74 \end{pmatrix} = \begin{pmatrix} 7/74 \\ -5/74 \end{pmatrix}$; $\langle b_1^*, b_2^* \rangle = 5 \cdot \frac{7}{74} + 7 \cdot \frac{-5}{74} = 0$ ☑

Man sieht allerdings leicht, dass i. A. nun $z_1 \cdot b_1^* + z_2 \cdot b_2^* \notin L$ ↯!

(met. Code im Skript) [2] Selbes Verfahren, nur runde $\lceil \nu_{k,j} \rfloor$, um im Gitter zu verbleiben. („Reduktion")

↳ Bsp.: $\lceil -117/74 \rfloor = \lceil -1.58\dots \rfloor = -2$: $b_1 = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$; $b_2 = \begin{pmatrix} 8 \\ 11 \end{pmatrix} - 2 \begin{pmatrix} 5 \\ 7 \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \end{pmatrix}$;

$b_1 = \begin{pmatrix} 5 \\ 7 \end{pmatrix}$ u. $b_2 = \begin{pmatrix} -2 \\ -3 \end{pmatrix}$ nun zwar Basis v. L ☑, aber $\langle b_1, b_2 \rangle = -31 \neq 0$ ↯!

[3]/[4] Nicht nur Reduzieren, sondern auch Tauschen:

↳ denn (Zitat Skript): „Sometimes the reduction algorithm produces shorter vectors when we switch two basis vectors b_i and b_{i+1}."

„lattice base reduction"

LLL-Algorithmus (A. Lenstra, H. Lenstra, L. Lovász):

(de.wiki): ↳ = erster effizienter Gitterreduktionsalgorithmus (bezogen auf eukl. Norm)

(de.wiki): ↳ berechnet für ein Gitter eine Basis aus möglichst kurzen Vektoren

```
def LLL-reduction(matrix = (b_1, ..., b_n), δ ∈ [1/4, 1] = 3/4):        # zeilenw.
    n = len(matrix.rows())
    for r in range(1, ∞):                                               # (entspricht [1])
        (b_1, ..., b_n), {b_i*}, {ν_{i,j}} = reduction(matrix = (b_1, ..., b_n))   # (entspricht [2])
        for i in range(0, n-1):
            if not Lovacz-condition(b_i*, b_{i+1}*, ν_{i+1,i}, δ):       # L.-Bed. verletzt
                b_i, b_{i+1} = b_{i+1}, b_i                             # vertausche
                continue outer loop
        break
    return (b_1, ..., b_n), {b_i*}, {ν_{i,j}}
```

(„irre") (1.) Dieser Algorithmus terminiert ▽ (aber nur für $\delta \in (\frac{1}{4}, 1)$!)

(2.) Er hat sogar polynomielle Laufzeit: $n^6 \cdot (\ln(\max \|b_i\|))^3$ (! Dimension Gitter !)

```
def Lovacz-condition(b_i*, b_{i+1}*, ν_{i+1,i}, δ):   # für alle 1 ≤ i < n
    return δ · ||b_i*||^2 ≤ ||b_{i+1}*||^2 + ν_{i+1,i} · ||b_i*||^2          (en.wiki)
⇔   return (δ − ν_{i+1,i}^2) ||b_i*||^2 ≤ ||b_{i+1}*||^2                    (Skript)
```

(mit $\{b_i^*\}$ den zugehörigen GSO-Daten)

(3.) Eine δ-LLL-reduzierte Basis $(b_1, \dots, b_n)$ hat folgende Eigenschaften:

(en.wiki): (a) Der erste Vektor b_1 kann nicht viel größer sein als der kürz. Vektor $\neq 0$:

(en.wiki): für $\delta = \frac{3}{4}$ im Speziellen: $\|b_1\| \leq 2^{(n-1)/2} \cdot \lambda_1(L)$

(en.wiki): (b) Außerdem ist b_1 durch die Determinante des Gitters beschränkt:

(en.wiki): für $\delta = \frac{3}{4}$ im Speziellen: $\|b_1\| \leq 2^{(n-1)/4} \cdot (\det(L))^{1/n}$

(en.wiki): (c) Das Produkt der Normen der Basisv. kann n. viel größ. sein als die Det. des Gitters:

(en.wiki): für $\delta = \frac{3}{4}$: $\prod_{i=1}^{n} \|b_i\| \leq 2^{n(n-1)/4} \cdot \det(L)$ (VL:) $\det(L) \leq$

(VL): (d) $\min \|b_i\| \leq \lambda_1(L)$ (e) $\|b_j\| \leq 2^{(i-1)/2} \|b_i^*\|$, $1 \leq j \leq i \leq n$

(f) $\|b_i^*\|^2 \leq \|b_i\|^2 \leq (\frac{1}{2} + 2^{i-2}) \cdot \|b_i^*\|^2$, $1 \leq i \leq n$ (g) $\|b_i^*\|^2 \leq 2 \cdot \|b_{i+1}^*\|^2$, $1 \leq i < n$

(VL:) (m.M.n. anderswum!) Notation: $\lambda_i(L) := \max_{S} \left(\min_{\text{bes}} \|b\| \right)$, $S \subseteq L$, $|S| = i$; $\lambda_1^{(L)} =$ kürz. Vektor $\neq 0$ in L (= Länge)