The art of cryptography, summer 2013

Lattices and cryptography

Prof. Dr. Joachim von zur Gathen

Given a lattice $L \subseteq \mathbb{R}^n$ and $u \in \mathbb{R}^n$, the distance of u to L is

$$d(u, L) = \mathsf{min}\{\|u - x\|\colon x \in L\}.$$

An element $x \in L$ is a *closest vector* to u if $\|u - x\| = d(u, L)$. The *closest vector problem* CVP is to compute such an x, given u and a basis of L. In the approximate *close vector problem* α-CVP, we are also given some $\alpha \geq 1$ and have to compute $x \in L$ with $\|u - x\| \leq \alpha \cdot d(u, L)$.

The following heuristic method reduces 2α-CVP to α-SVP. The claim is unproven, but the algorithm seems to work in practice, at least in high dimensions.

We write $u = (u_1, \ldots, u_n)$ and place the $n \times n$ matrix A representing a basis of L into the top right part of the following matrix:

$$A' = \begin{pmatrix} 0 & & & \\ \vdots & & A & \\ 0 & & & \\ 1 & u_1 & \cdots & u_n \end{pmatrix}. \tag{1}$$

Using an algorithm for α-SVP, we compute a nonzero $y = (y_0, y_1, \ldots, y_n)$ in the lattice $M \subseteq \mathbb{R}^{n+1}$ generated by the rows of A', with $\|y\| \leq \alpha \cdot \lambda_1(M)$. If y_0 divides $y_1, \ldots, y_n$, we return $u - y_0^{-1} \cdot (y_1, \ldots, y_n)$, and otherwise "failure".

ALGORITHM 2. Nearest hyperplane.

Input: A reduced basis $B = (b_1, \ldots, b_\ell)$ of an ℓ-dimensional lattice L in $\mathbb{R}^n$, and $u \in \mathsf{span}_{\mathbb{R}}(L) \subseteq \mathbb{R}^n$.
Output: $x \in L$ with $\|u - x\| \le 2^{\ell/2} d(u, L)$.

1. Compute the GSO $(b_1^*, \ldots, b_\ell^*)$ of $(b_1, \ldots, b_\ell)$.
2. Compute $c = u \star b_\ell^* / (b_\ell^* \star b_\ell^*)$.
3. $c' \longleftarrow \lceil c \rfloor$,
   $v \longleftarrow u - (c - c') b_\ell^*$,
   $y \longleftarrow c' b_\ell$.
4. If $\ell = 1$, then return $x = y$. Else let M be the lattice generated by $b_1, \ldots, b_{\ell-1}$. Call the algorithm recursively to return $z \in M$ close to $v - y$.
5. Return $x = y + z$.

THEOREM 3. *The output* x *of the nearest hyperplane algorithm Algorithm 2 satisfies* $\|u - x\| < 2^{\ell/2} d(u, L)$. *It runs in polynomial time.*

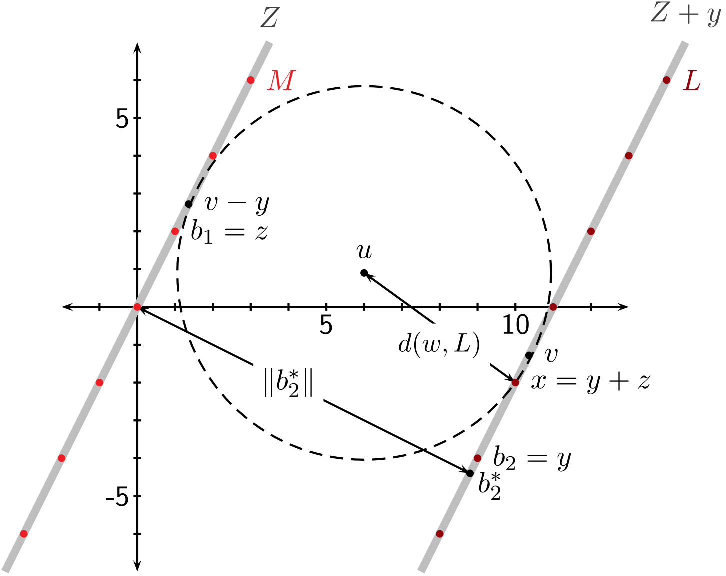

Figure: Trace of Algorithm 2 on the reduced basis $(b_1, b_2) = ((1,2),(9,-4))$ and the target vector $u = (6, 0.9)$.

EXAMPLE 4. We take the reduced basis $(b_1, b_2) = ((1,2),(9,-4))$ of the lattice L with $\ell = n = 2$, and $u = (6, 0.9)$. In the figure, one sees four candidates in L that look close to u. Which one is the closest? We have

$$\begin{aligned} b_1^* = b_1, b_2^* &= (44/5, -22/5), \\ u &= \frac{39}{25} b_1^* + \frac{111}{220} b_2^*, \\ c' &= \left\lceil \frac{111}{220} \right\rfloor = 1, \\ v &= u - (\frac{111}{220} - 1) \cdot b_2^* = (10.36, -1.28), \\ y &= (9, -4), \\ v - y &= (10.36, -1.28) - (9, -4) = (1.36, 2.72). \end{aligned}$$