Recovering ECDSA nonce with partial information

Federico Zanca

ECDSA (Elliptic Curve Digital Signature Algorithm)

Parameters: Elliptic curve E,generator point G on E of order n
Private Key: integer d
Public Key: Curve point $Q = dG$

Sign message m:

- Hash the message $h = H(m)$
- Randomly pick a nonce k
- Compute
  - $r = (kG)_x \bmod n$
  - $s = k^{-1}(h + dr) \bmod n$
- Signature is (r , s)

Nonce recovery and ECDSA security

k must be generated uniformly at random,
or we can use many signatures to compute the private key d

$$\left.\begin{aligned} k_1 - s_1^{-1} r_1 d - s_1^{-1} h_1 &\equiv 0 \bmod n \\ k_2 - s_2^{-1} r_2 d - s_2^{-1} h_2 &\equiv 0 \bmod n \\ &\vdots \\ k_m - s_m^{-1} r_m d - s_m^{-1} h_m &\equiv 0 \bmod n \end{aligned}\right\} \rightarrow \text{lattice attacks} \rightarrow d$$

If the k_i are *small*, system of equations likely has unique solution and lattice techniques can find d.

$$d = (sk - h)\, r^{-1} \bmod n$$

Nonce recovery and ECDSA security

Partial information on specific bits of the nonce can be recovered via side-channel attacks

- MOD not running in constant time (if < n the function return early)
- Scalar multiplication of points on EC not running in constant time
- Many more...

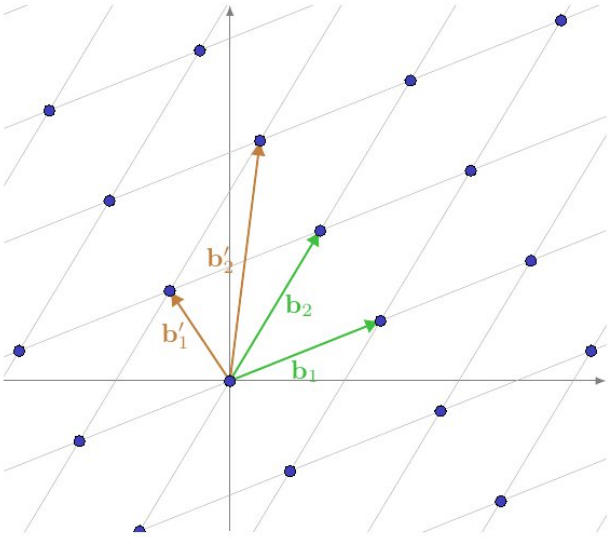
Mathematical Background

Lattices, lattice problems, lattice reduction

Lattices

Definition. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^m$ be a set of linearly independent vectors. The *lattice L generated by* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ is the set of linear combinations of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ with coefficients in $\mathbb{Z}$,

$$L = \{a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_n\boldsymbol{v}_n : a_1, a_2, \ldots, a_n \in \mathbb{Z}\}.$$



A 2-dimensional lattice and two different bases for it

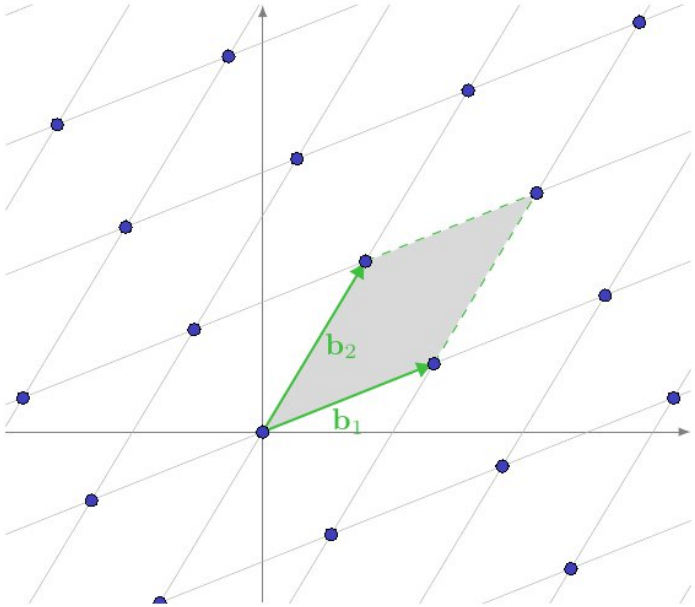
A basis for L is any set of independent vectors that generates L

Although a lattice can be represented by many different bases, some are particularly "good" for solving certain computational problems

Lattices

Definition (Fundamental Domain). Let $\mathbf{B} = \{\mathbf{b}_1,\ldots,\mathbf{b}_m\}$ be a basis. The fundamental domain (or parallelepiped) of $\mathbf{B}$ is defined as

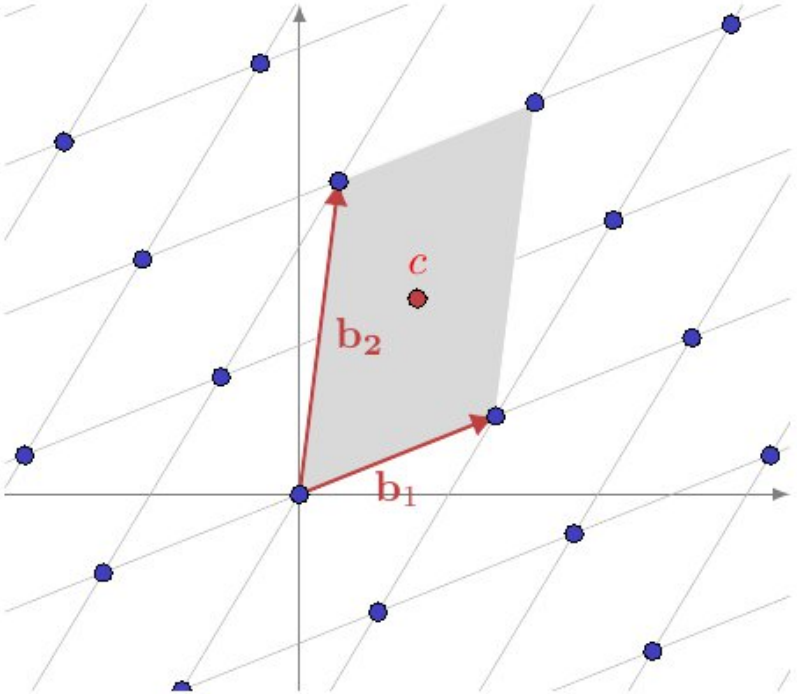
$$\mathcal{P}(\mathbf{B}) = \left\{ \sum_{i=1}^{m} a_i \mathbf{b}_i \mid a_i \in [0,1) \right\}$$



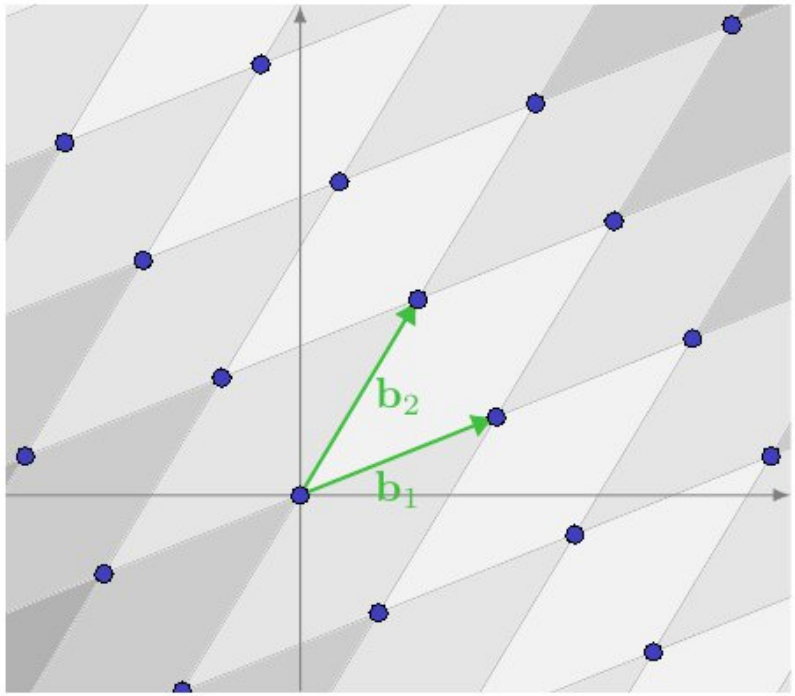
A basis $\mathbf{B}$ is a basis for the lattice L iff $\mathcal{P}(\mathbf{B}) \cap L = \{\mathbf{0}\}$

Lattices

A basis $\mathbf{B}$ is a basis for the lattice L iff $\mathcal{P}(\mathbf{B}) \cap L = \{\mathbf{0}\}$

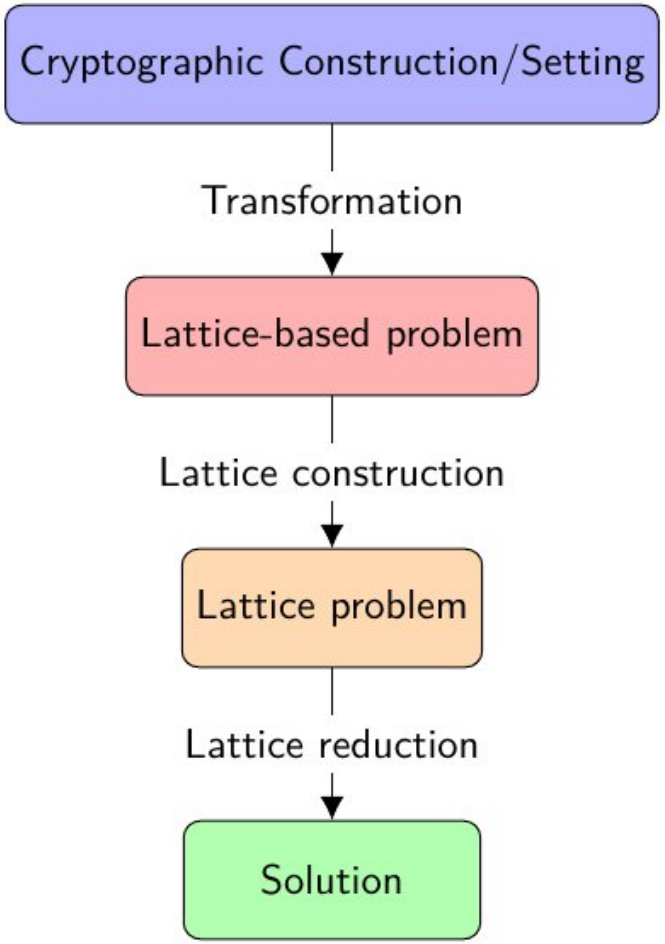


(a) $\{\mathbf{b}_1, \mathbf{b}_2\}$ do not form a basis, as $\mathcal{P}(\{\mathbf{b}_1, \mathbf{b}_2\})$ contains the non-zero lattice point c.



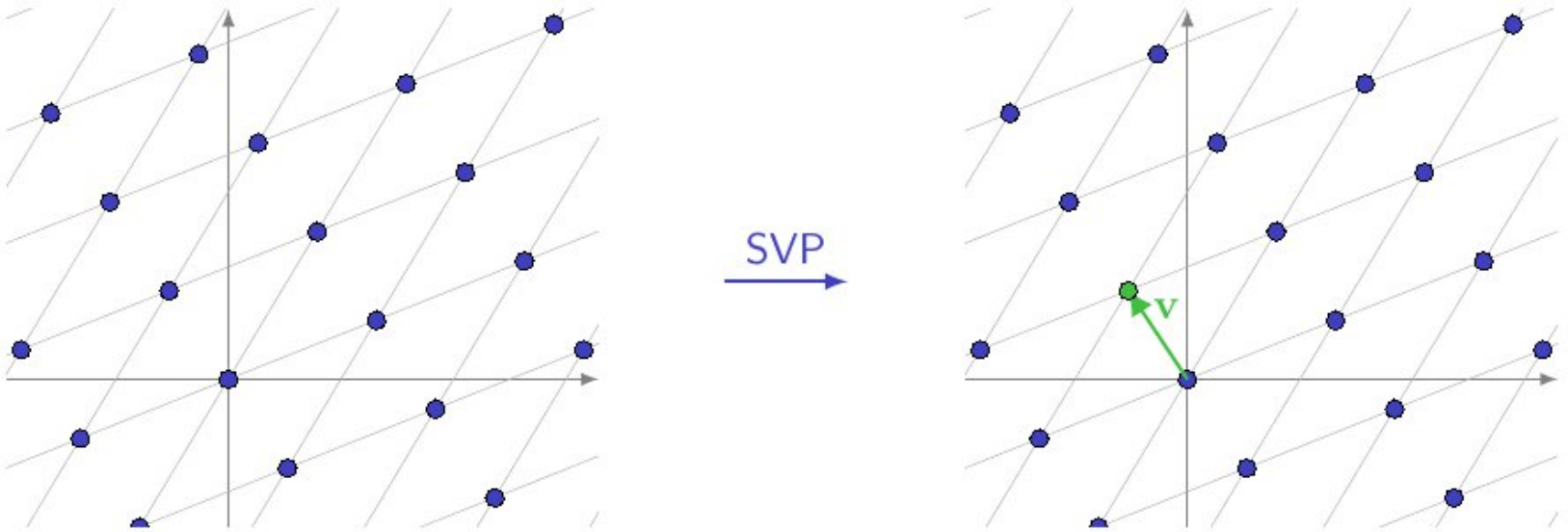
(b) $\{\mathbf{b}_1, \mathbf{b}_2\}$ form a basis, as $\mathcal{P}(\{\mathbf{b}_1, \mathbf{b}_2\})$ contains only the zero point, and tiles $\mathbb{R}^2$.

Lattices for cryptanalysis



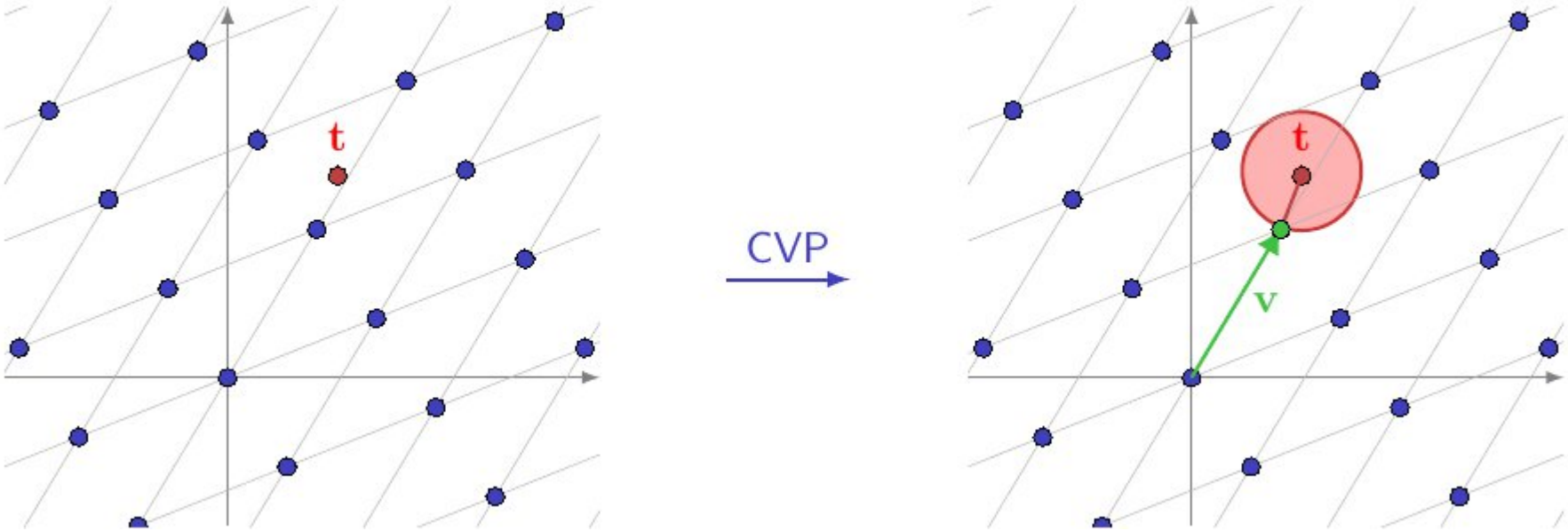
Shortest Vector Problem - SVP

Given a basis **B**, find a shortest non-zero vector $v \in L$ that minimizes $\|v\|$



Closest Vector Problem - CVP

Given a basis **B** of a lattice L, and a target vector **t** (not necessarily in L), find a lattice vector v that satisfies $\|v - \mathbf{t}\| = \min_{w \in L} \|w - t\|$



It is well-known that these problems are NP-Hard

apprSVP and apprCVP

Approximate Shortest Vector Problem ($\mathbf{SVP}_\gamma$)
Given a basis $\mathbf{B}$ of a lattice L find a non-zero lattice vector $\boldsymbol{v}$ that satisfies $\|\boldsymbol{v}\| \leq \gamma \boldsymbol{\lambda_1}(L)$

Approximate Closest Vector Problem ($\mathbf{CVP}_\gamma$)
Given a basis $\mathbf{B}$ of a lattice L and a target vector $\mathbf{t}$ and an approximation factor γ, find a lattice vector $\boldsymbol{v}$ that satisfies $\|\boldsymbol{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\boldsymbol{w} \in L} \|\boldsymbol{w} - \boldsymbol{t}\|$

Lattice Reduction is used to solve these problems.
Main idea: transform an arbitrary lattice basis into a "better" basis that contains shorter and more orthogonal vectors.

Lattice reduction algorithms

Main idea: transform an arbitrary lattice basis into a "better" basis that contains shorter and more orthogonal vectors.

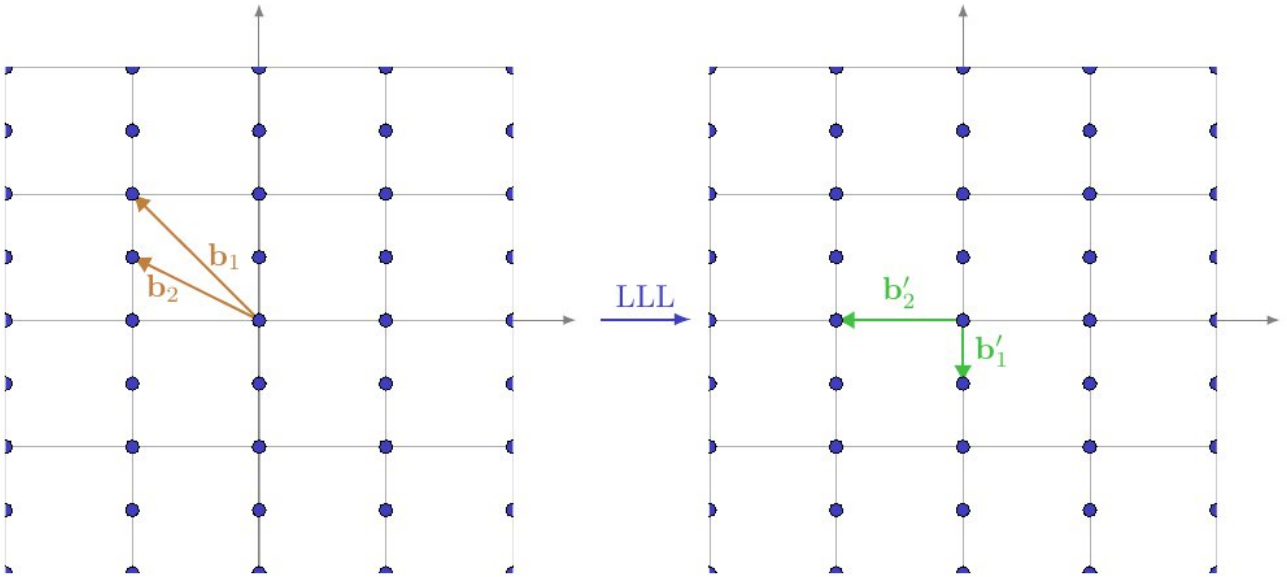
LLL and BKZ (LLL generalization) solve SVP

CVP can be solved using Babai's algorithm or Kannan's embedding method (transforms CVP in SVP with +1 dimensions)

LLL (Lenstra-Lenstra-Lovász algorithm)

LLL solves apprSVP/apprCVP within an approximation factor exponential in the dimension of the lattice to the shortest vector in polynomial time $\|v\| \leq C^n \|v\|_{\text{shortest}}$

In practice: $\|v\| \leq 1.02^n (\det L)^{1/n}$



(a) Lattice $\mathcal{L}(\mathbf{B})$ with original basis $\mathbf{b}_1 = (-2, 2)$, $\mathbf{b}_2 = (-2, 1)$.

(b) Lattice with LLL-reduced basis $\mathbf{b}'_1 = (0, -1)$, $\mathbf{b}'_2 = (-2, 0)$.

Reduction algorithms like LLL transform a basis **B** for a lattice L into a "better" basis: vectors should be as short as possible and as orthogonal as possible to one another

LLL the actual algorithm

Definition Let $\mathbf{B} = \{v_1,...,v_n\}$ be a basis for a lattice L and let $\mathbf{B}^* = \{v_1^*,...v_n^*\}$ be the associated Gram-Schmidt orthogonal basis. **B** is said to be *LLL reduced* if it satisfies the following two conditions:

$$\textbf{(Size Condition)} \quad |\mu_{i,j}| = \frac{|\boldsymbol{v}_i \cdot \boldsymbol{v}_j^*|}{\|\boldsymbol{v}_j^*\|^2} \leq \frac{1}{2} \qquad \text{for all } 1 \leq j < i \leq n.$$

$$\textbf{(Lovász Condition)} \quad \|\boldsymbol{v}_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|\boldsymbol{v}_{i-1}^*\|^2 \quad \text{for all } 1 < i \leq n.$$

2 crucial steps in the algorithm:

- Size reduction step
  - it aims to make the current basis vector $\boldsymbol{b_i}$ smaller in the direction of the previous basis vector $\boldsymbol{b_{i-1}}$
- Swap step
  - ensures that the basis vectors satisfy the Lovász condition, which is necessary to maintain progress towards finding a reduced basis

LLL the actual algorithm

Size reduction step: we do this by subtracting from v_k appropriate integer multiples ($\mu_{i,j}$ rounded to the closest integer) of the previous vectors $v_1,\ldots,v_{k-1}$ so as to make v_k smaller.

After size reduction check if Lovász condition is satisfied.

- If it is, we have a nearly optimal ordering of the vectors.
- If not, reorder the vectors and do further size reduction.

[1] Input a basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ for a lattice L
[2] Set $k = 2$
[3] Set $\boldsymbol{v}_1^* = \boldsymbol{v}_1$
[4] Loop while $k \leq n$
[5] Loop Down $j = k-1, k-2, \ldots, 2, 1$
[6] Set $\boldsymbol{v}_k = \boldsymbol{v}_k - \lfloor \mu_{k,j} \rceil \boldsymbol{v}_j$ [Size Reduction]
[7] End j Loop
[8] If $\|\boldsymbol{v}_k^*\|^2 \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right) \|\boldsymbol{v}_{k-1}^*\|^2$ [Lovász Condition]
[9] Set $k = k + 1$
[10] Else
[11] Swap $\boldsymbol{v}_{k-1}$ and $\boldsymbol{v}_k$ [Swap Step]
[12] Set $k = \max(k-1, 2)$
[13] End If
[14] End k Loop
[15] Return LLL reduced basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$

Note: at each step $v_1^*,\ldots,v_k^*$ is the orthogonal set of vectors obtained by applying Gram-Schmidt to the current values of $v_1,\ldots,v_k$ and $\mu_{i,j} = \frac{v_i \cdot v_j^*}{\|v_j^*\|^2}$

BKZ-LLL block Korkin-Zolotarev LLL

The block Korkin–Zolotarev variant of the LLL algorithm replaces the swap step in the standard LLL algorithm by a block reduction step (block of size β) to achieve a more global reduction across the entire basis.

LLL works with β=2 (performs reduction working on pairs of vectors).

BKZ-LLL works with a block of vectors of length β

$$v_k, v_{k+1}, \ldots, v_{k+\beta-1}$$

and replaces them with a KZ-reduced basis spanning the same sublattice.

Kannan's embedding method

Express a CVP problem as a SVP instance to solve it
Basis: $\mathbf{B} = \{\mathbf{b_1},...,\mathbf{b_n}\}$
Target vector $\mathbf{t} = (t_1,...,t_n)$
Solution to CVP problem: $c_1\mathbf{b_1} + c_2\mathbf{b_2} + ... + c_n\mathbf{b_n}$

Then we have $\mathbf{t} \approx \sum_{i=1}^{n} c_i\mathbf{b}_i \Rightarrow \mathbf{t} = \sum_{i=1}^{n} c_i\mathbf{b}_i + \mathbf{e}$ where $\|\mathbf{e}\|$ is small.

Hence, consider the $n + 1$ dimensional lattice with basis

$$\mathbf{B'} = \begin{bmatrix} \mathbf{B} & 0 \\ \mathbf{t} & q \end{bmatrix}$$

which contains the short vector $(\mathbf{e}, q)$ by the linear combination $(-c_1,..., -c_n, 1)$

The solution is given by subtracting $\mathbf{e}$ from $\mathbf{t}$

HNP Hidden Number Problem

Secret integer α mod p
Known m pairs of integers $\{(t_i, a_i)\}_{i=1}^{m}$ such that

$$t_i\alpha - a_i = b_i \mod p$$

with $|b_i| < B$ for some $B < p$

Reformulate as seeking a solution $x_i = b_i$, $y = \alpha$ to the unconstrained system

$$x_1 - t_1 y + a_1 \equiv 0 \mod p$$

$$\vdots$$

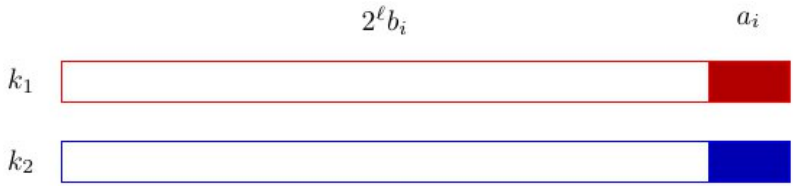
$$x_m - t_m y + a_m \equiv 0 \mod p$$

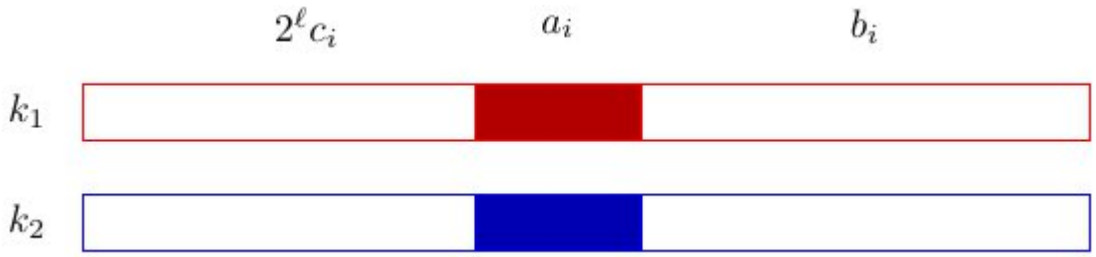
This is solvable by solving CVP with lattices
It's easier to solve SVP so express it as an SVP instance

ECDSA as HNP instance

Formulate the ECDSA key recovery problem as an instance of the Hidden Number Problem and compute the shortest vector of a specially constructed lattice to obtain the solution

Assumption: the attacker has access to multiple signatures and hashes of the messages





ECDSA as HNP instance

$$k_1 - t_1 d - a_1 \equiv 0 \bmod n$$
$$\vdots$$
$$k_m - t_m d - a_m \equiv 0 \bmod n$$

in unknowns k_i , d where $|k_i| < B$

$$t_i = s_i^{-1} r_i \qquad a_i = s_i^{-1} h_i$$

Construct the lattice basis

$$M = \begin{bmatrix} n & & & & \\ & n & & & \\ & & \ddots & & \\ & & & n & \\ t_1 & t_2 & \dots & t_m & \end{bmatrix}$$

Solve CVP with target vector $v_t = (a_1, a_2, \dots, a_m)$ using Kannan's embedding
$v_k = (k_1, k_2, \dots, k_m)$ will be the distance

$$M = \begin{bmatrix} n & & & & & \\ & n & & & & \\ & & \ddots & & & \\ & & & n & & \\ t_1 & t_2 & \dots & t_m & B/n & \\ a_1 & a_2 & \dots & a_m & & B \end{bmatrix}$$

$v_k = (k_1, k_2, \dots, k_m, \frac{Bd}{n}, B)$ is a short vector in this lattice

Assumption: k_i are small
(which means MSB = 0)

Known nonzero MSB

If MSB of k_i are nonzero and known, we can write $k_i = (x_i + y_i)$ where x_i is the leak (left shifted) and $|y_i| < \mathrm{K}$.

Then $k - td - a \equiv 0 \bmod n$ becomes

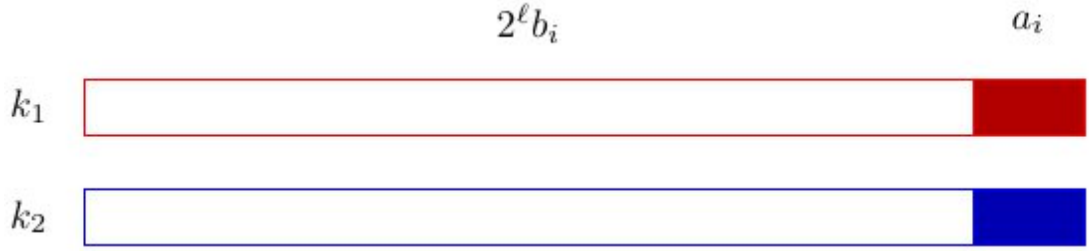
$$(x + y) - td - a \equiv 0 \bmod n$$

$$y - td - u \equiv 0 \bmod n$$

where $u = s^{-1}h - x$

This is again an instance of HNP and y is small

Known nonzero LSB



If LSB of k_i are nonzero and known, we can write $k_i = (x_i + 2^l y_i)$ where x_i is the leak, l is the size of the leak and $|y_i| < \mathrm{K}$.

Then $k - td - a \equiv 0 \bmod n$ becomes

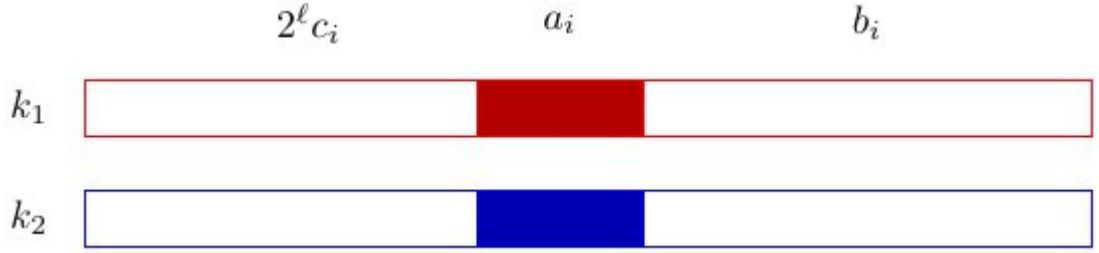
$$(x + 2^l y) - td - a \equiv 0 \bmod n$$

$$y - td - u \equiv 0 \bmod n$$

where $u = 2^{-l}(s^{-1}h - x)$

Once again, solve HNP as before and retrieve the key

Known middle bits



This is more complex, since we have two unknown chunks of the nonce to recover per signature. A generally larger leak is needed too.
Given two signatures generated with the same private key, the nonces satisfy

$$k_1 + tk_2 + u \equiv 0 \bmod n$$

where $t = -s_1^{-1}s_2r_1r_2^{-1}$ and $u = s_1^{-1}r_1h_2r_2^{-1} - s_1^{-1}h_1$

Since we know the middle bits (shifted) of k_1 and k_2 are a_1 and a_2 respectively, we can write

$$k_1 = a_1 + b_1 + 2^l c_1 \qquad \text{and} \qquad k_2 = a_2 + b_2 + 2^l c_2$$

where b_1, c_1, b_2, c_2 are unknown but small, less than some bound K

Known middle bits

$$k_1 + tk_2 + u \equiv 0 \bmod n$$
$$b_1 + 2^\ell c_1 + tb_2 + 2^\ell tc_2 + a_1 + ta_2 + u \equiv 0 \bmod n$$

Let $u' = a_1 + ta_2 + u$. We wish to find the small solution $x_1 = b_1$, $y_1 = c_1$, $x_2 = b_2$, $y_2 = c_2$ to the linear equation

$$f(x_1, y_2, x_2, y_2) = x_1 + 2^\ell y_1 + tx_2 + 2^\ell ty_2 + u' \equiv 0 \bmod n$$

$$B = \begin{bmatrix} K & K \cdot 2^{49} & Kt & Kt \cdot 2^{49} & u' \\ & Kn & & & \\ & & Kn & & \\ & & & Kn & \\ & & & & n \end{bmatrix}$$

Running BKZ on B we obtain a basis containing the vector
$v = (z_1\mathrm{K},\ z_2\mathrm{K},\ z_3\mathrm{K},\ z_4\mathrm{K},\ z_5)$
This corresponds to the linear equation

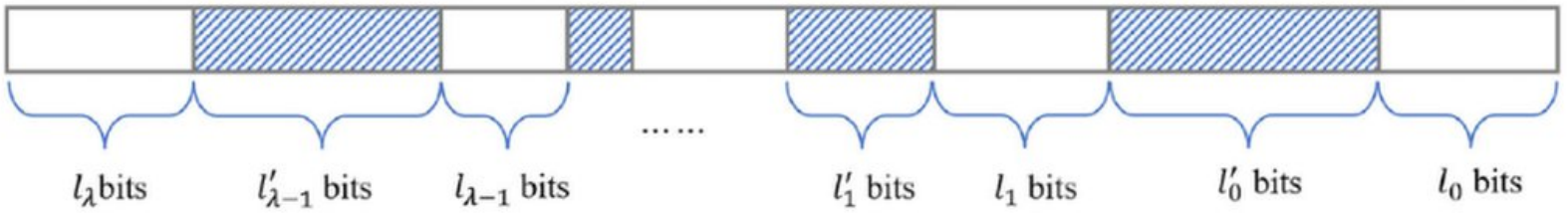
$$z_1x_1 + z_2y_1 + z_3x_2 + z_4y_2 + z_5 = 0$$

Do the same for the next three short vectors in the basis, solve the system and retrieve

$$x_1\ ,\ y_1\ ,\ x_2\ ,\ y_2$$

Generalization of the attack

It is possible to retrieve the key starting from chunks of bits in arbitrary positions of the nonce solving an instance of the Extended Hidden Number Problem



References

- Survey: Recovering cryptographic keys from partial information
- Biased Nonce Sense: Lattice Attacks against Weak ECDSA Signatures in Cryptocurrencies
- Return of the Hidden Number Problem
- Improved Attacks on (EC)DSA with Nonce Leakage by by Lattice Sieving
- A Gentle Tutorial for Lattice-Based Cryptanalysis
- An Introduction to Mathematical Cryptography