Lattice

Santanu Sarkar

Indian Institute of Technology Madras, Chennai

Factorization Methods

"The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic."- Carl Friedrich Gauss

- Pollard's $p-1$ algorithm (1974)
- Quadratic Sieve (QS): Pomerance (1981)
- Elliptic Curve Method (ECM): H. W. Lenstra (1987)
- Number Field Sieve (NFS): A. K. Lenstra et al.(1993)

Time table

RSA Number	Decimal digits	Binary digits	When	Who
RSA-100	100	330	April 1, 1991	A. Lenstra
RSA-120	120	397	July 9, 1993	T. Denny et al.
RSA-130	130	430	April 10, 1996	A. Lenstra et al.
RSA-150	150	496	April 16, 2004	K. Aoki et al.
RSA-160	160	530	April 1, 2003	J. Franke et al.
RSA-576	174	576	December 3, 2003	J. Franke et al.
RSA-190	190	629	November 8, 2010	A. Timofeev et al.
RSA-200	200	663	May 9, 2005	J. Franke et al.
RSA-704	212	704	July 2, 2012	S. Bai et al.
RSA-768	232	768	December 12, 2009	T. Kleinjung et al.

RSA-768

123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3

Lattice

Lattice based Root Finding of Polynomials

Motivation

No efficient algorithm for factorization of $N = pq$ is known.

Motivation

No efficient algorithm for factorization of $N = pq$ is known.

First equation: $N = pq \qquad \longrightarrow \qquad f(x, y) = N - xy$

Motivation

No efficient algorithm for factorization of $N = pq$ is known.

First equation: $N = pq \longrightarrow f(x, y) = N - xy$

Second equation:

- $ed = 1 + k(N + 1 - p - q) \longrightarrow 1 + k\big(N + (1 - p - q)\big) \equiv 0 \bmod e$
- $f(x, y) = 1 + x(N + y) \in \mathbb{Z}_e[x, y]$ with root $(k, 1 - p - q)$

Finding roots of a polynomial

Univariate Integer Polynomial

- $f(x) \in \mathbb{Z}[x]$ with root $x_0 \in \mathbb{Z}$ efficient methods available

Multivariate Integer Polynomial

- $f(x, y) \in \mathbb{Z}[x, y]$ with root $(x_0, y_0) \in \mathbb{Z} \times \mathbb{Z}$ not efficient

Univariate Modular Polynomial

- $f(x) \in \mathbb{Z}_N[x]$ with root $x_0 \in \mathbb{Z}_N$ not efficient

Hilbert's tenth Problem: 1900

Finding roots of a polynomial

Univariate Integer Polynomial

- $f(x) \in \mathbb{Z}[x]$ with root $x_0 \in \mathbb{Z}$ efficient methods available

Multivariate Integer Polynomial

- $f(x, y) \in \mathbb{Z}[x, y]$ with root $(x_0, y_0) \in \mathbb{Z} \times \mathbb{Z}$ not efficient

Univariate Modular Polynomial

- $f(x) \in \mathbb{Z}_N[x]$ with root $x_0 \in \mathbb{Z}_N$ not efficient

Hilbert's tenth Problem: 1900

Lattice based techniques help in some cases.

Lattice

Definition (Lattice)

Let $\mathbf{v_1}, \ldots, \mathbf{v}_\omega \in \mathbb{Z}^\omega$ be ω linearly independent vectors. Lattice $L = \{\mathbf{v} \in \mathbb{Z}^\omega \mid \mathbf{v} = \sum_{i=1}^{\omega} a_i \mathbf{v_i} \text{ with } a_i \in \mathbb{Z}\}$.

$\det(L)$ is determinant $\begin{bmatrix} v_1 \\ \vdots \\ v_\omega \end{bmatrix}$.

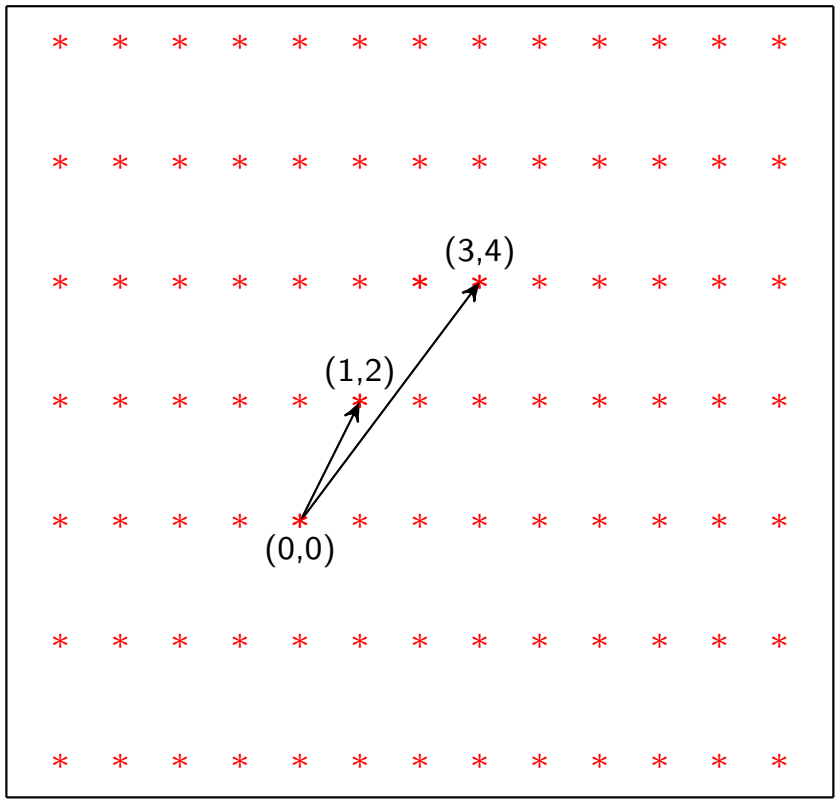
Example

Consider two vectors $\mathbf{v_1} = (1, 2), \mathbf{v_2} = (3, 4)$. The lattice L generated by $\mathbf{v_1}, \mathbf{v_2}$ is
$L = \{\mathbf{v} \in \mathbb{Z}^2 \mid \mathbf{v} = a_1\mathbf{v_1} + a_2\mathbf{v_2} \text{ with } a_1, a_2 \in \mathbb{Z}\}$.

$$B = \left\{ v_1 = (1, 2), v_2 = (3, 4) \right\}$$

$$B' = \left\{ r_1 = (1, 0), r_2 = (0, 2) \right\}$$

Lattice



Lattice

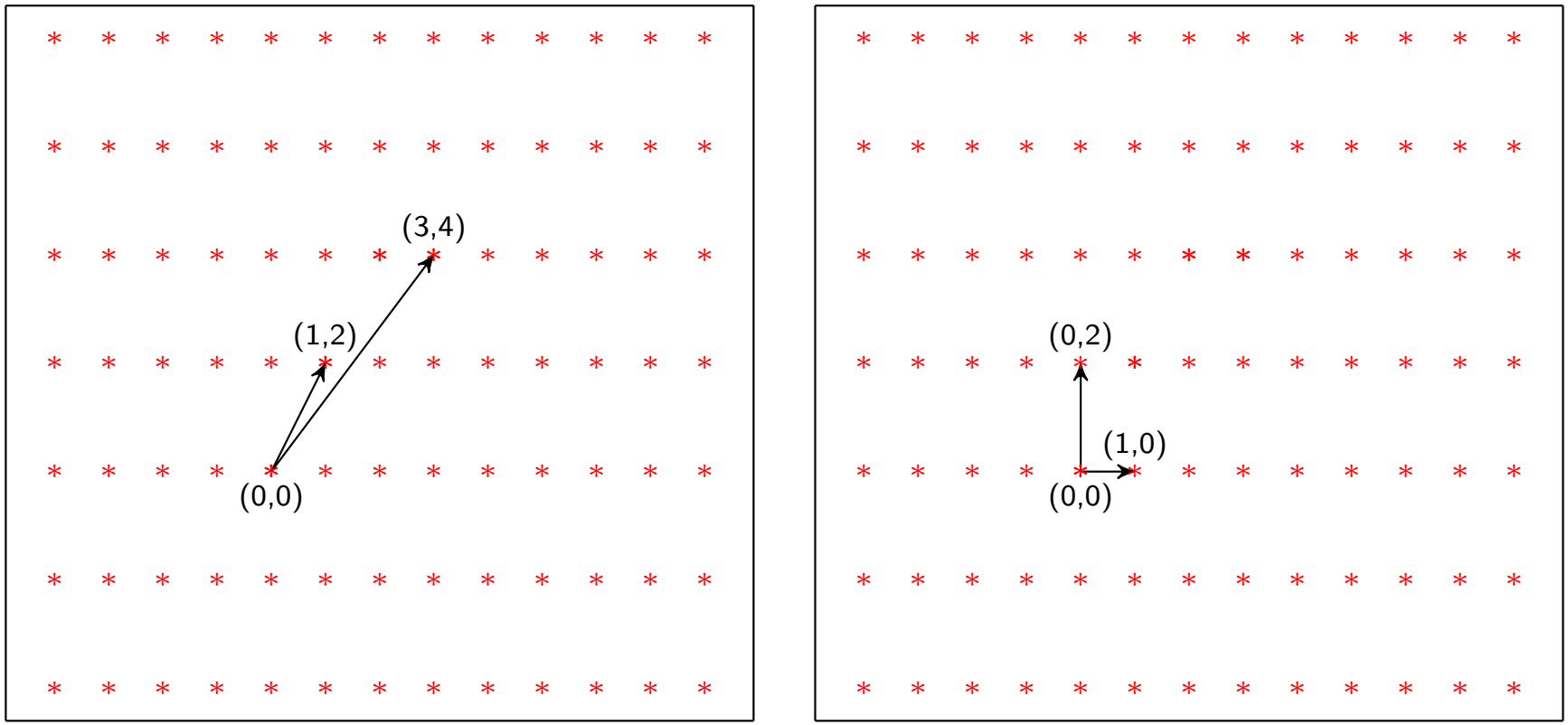


Figure: Lattice L with different basis.

LLL Algorithm



Devised by A. Lenstra, H. Lenstra and L. Lovász (Mathematische Annalen 1982)

Main goal: Produce lattice basis 'short (bounded)' and 'nearly orthogonal'

$$\left\{ v_1 = \{1, 2\}, v_2 = \{3, 4\} \right\} \Rightarrow \left\{ r_1 = \{1, 0\}, r_2 = \{0, 2\} \right\}$$

LLL Algorithm



Devised by A. Lenstra, H. Lenstra and L. Lovász (Mathematische Annalen 1982)

Main goal: Produce lattice basis 'short (bounded)' and 'nearly orthogonal'

$$\left\{v_1 = \{1, 2\}, v_2 = \{3, 4\}\right\} \Rightarrow \left\{r_1 = \{1, 0\}, r_2 = \{0, 2\}\right\}$$

Non zero $||r|| \leq \sqrt{\omega}\det(L)^{\frac{1}{\omega}}$: Minkowski

Connecting LLL to Root finding

The clue was provided by Nick Howgrave-Graham in 1997.

Connecting LLL to Root finding

The clue was provided by Nick Howgrave-Graham in 1997.

Theorem

Let $h(x) \in \mathbb{Z}[x]$ *be an integer polynomial with* ω *monomials. Let*

$$h(x_0) \equiv 0 \pmod{N} \text{ with } |x_0| < X \qquad \text{and} \qquad ||h(xX)|| < \frac{N}{\sqrt{\omega}}.$$

Then, $h(x_0) = 0$ *holds over integers.*

Connecting LLL to Root finding

The clue was provided by Nick Howgrave-Graham in 1997.

Theorem

Let $h(x) \in \mathbb{Z}[x]$ be an integer polynomial with ω monomials. Let

$$h(x_0) \equiv 0 \pmod{N} \text{ with } |x_0| < X \qquad \text{and} \qquad ||h(xX)|| < \frac{N}{\sqrt{\omega}}.$$

Then, $h(x_0) = 0$ holds over integers.

Proof.

$$|h(x_0)| = \left|\sum_i h_i x_0^i\right| \leq \sum_i \left|h_i x_0^i\right| \leq \sum_i |h_i| X^i \leq \sqrt{n} \cdot ||h(xX)|| < N.$$

Now since N divides $h(x_0)$ by the first condition, $h(x_0) = 0$. $\square$

Connecting LLL to Root finding

- $a_0 + a_1 x + \ldots + a_\omega x^\omega \leftrightarrow (a_0, a_1, \ldots, a_\omega)$
- $f(x) \in \mathbb{Z}_N[x]$ with $f(x_0) \equiv 0 \bmod N$ with $|x_0| < X$
- Generate $f_1(x), \ldots, f_\omega(x)$ with $f_1(x_0) \equiv \cdots \equiv f_\omega(x_0) \equiv 0 \bmod N$
- Construct L from $f_1(xX), \ldots, f_\omega(xX)$.
- LLL: $||r_1|| < \left(\det(L)\right)^{\frac{1}{\omega}}$
- $\det(L)^{\frac{1}{\omega}} < N \Rightarrow r_1'(x_0) = 0$ where $r_1'(x) = r_1(\frac{x}{X})$

Example with Modular Polynomial

PROBLEM: Find a root x_0 of $f(x) = x^2 + ax + b \equiv 0 \bmod N$.

- Consider two more polynomials $g(x) = xN$ and $h(x) = N$
- Construct lattice from coefficient vectors of $f(xX)$, $g(xX)$, $h(xX)$

$$L = \begin{bmatrix} X^2 & aX & b \\ 0 & NX & 0 \\ 0 & 0 & N \end{bmatrix}$$

- Use LLL algorithm to reduce this lattice
- $\det(L) = X^3 N^2$
- Need: $\left(\det(L)\right)^{\frac{1}{3}} < N$
- If the root is bounded by $|x_0| < N^{\frac{1}{3}}$, the reduction works!

Example with Modular Polynomial

Construct lattice from coefficient vectors of f^2, xfN, Nf, N^2x, N^2

$$L = \begin{bmatrix} X^4 & - & - & - & - \\ 0 & NX^3 & - & - & - \\ 0 & 0 & X^2N & - & - \\ 0 & 0 & 0 & XN^2 & - \\ 0 & 0 & 0 & 0 & N^2 \end{bmatrix}$$

- If the root is bounded by $|x_0| < N^{\frac{2}{5}}$, the reduction works!
- Can be improved up to $|x_0| < N^{\frac{1}{2}}$, for higher lattice dimension.

Resultant of two polynomials

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n, g(x) = b_0x^m + b_1x^{m-1} + \cdots + b_m$$

$$R(f,g) = \begin{vmatrix} a_0 & 0 & \ldots & 0 & b_0 & 0 & \ldots & 0 \\ a_1 & a_0 & \ldots & 0 & b_1 & b_0 & \ldots & 0 \\ a_2 & a_1 & \ldots & 0 & b_2 & b_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{n-1} & 0 & 0 & \ldots & b_{m-1} \\ 0 & 0 & \ldots & a_n & 0 & 0 & \ldots & b_m \end{vmatrix}_{(m+n,m+n)}$$

Property of resultant

- $f(x_1, \ldots, x_t)$ and $g(x_1, \ldots, x_t)$ share the common root $(x_1^{(0)}, \ldots, x_t^{(0)})$.
- Let $R(f, g)$ be the resultant of f, g with respect to x_t.
- Then, we have
  1. $R(f, g)$ is a polynomial in $t - 1$ variables $x_1, \ldots, x_{t-1}$,
  2. $R(f, g) \neq 0$ if $\gcd(f, g) = 1$, and
  3. $R(f, g)(x_1^{(0)}, \ldots, x_{t-1}^{(0)}) = 0$.

Complexity

- LLL Lattice Reduction: Nguyen and Stehlé is $O\left(\omega^5\left(\omega+A\right)A\right)$, where A is the maximum bitsize of an entry in the lattice
- Root Extraction: determinant of a matrix of size D in $O(D^3)$
- Almost impossible to reduce a lattice of large dimension (larger than 400, say)

RSA Equations

The obvious one

$$N = pq \qquad \longrightarrow \qquad f(x, y) = N - xy$$

The other one

$$d \equiv e^{-1} \bmod \phi(N) \longrightarrow f(x, y, z) = ex - 1 - y(N + 1 - z)$$

Problem: None can be solved directly by the Lattice method!

RSA Equations

The obvious one

$$N = pq \qquad \longrightarrow \qquad f(x, y) = N - xy$$

The other one

$$d \equiv e^{-1} \bmod \phi(N) \longrightarrow f(x, y, z) = ex - 1 - y(N + 1 - z)$$

Problem: None can be solved directly by the Lattice method!

Does any partial information help?

Existing Results: Howgrave-Graham

- Suppose an approximation p_0 of p is known
- Let $x_0 = p - p_0$
- Consider $f(x) = p_0 + x$
- Hence $f(x_0) \equiv 0 \bmod p$
- Let $|x_0| \leq X$
- Aim: find x_0 from $f(x)$ and N

Existing Results: Howgrave-Graham

Construct lattice from coefficient vectors of N^2, Nf, f^2, xf^2

$$L = \begin{bmatrix} N^2 & 0 & 0 & 0 \\ Np_0 & NX & 0 & 0 \\ p_0^2 & 2p_0X & X^2 & 0 \\ 0 & p_0^2X & 2p_0X^2 & X^3 \end{bmatrix}$$

- Need $\det(L) = X^6N^3 < (p^2)^4$
- $p \approx N^{\frac{1}{2}} \rightarrow X < N^{\frac{1}{6}}$

Existing Results: Howgrave-Graham 1997

$$\begin{aligned} p_i(x) &= N^{u-i} f^i \text{ for } 0 \leq i \leq u, \\ p'_i(x) &= f^u x^{i-u} \text{ for } u < i \leq h, \end{aligned}$$

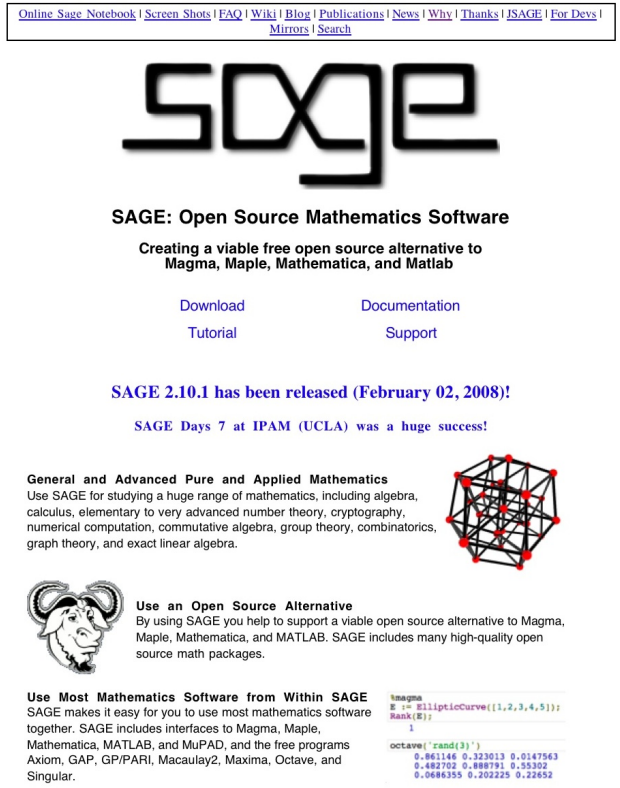
RESULT: If $|p - p_0| < N^{\frac{1}{4}}$, one can solve f in polynomial time.
[Coppersmith, 1996]

Existing Results

- Thus, $q_0 = \lfloor \frac{N}{p_0} \rfloor$ will be an approximation of p.
- Let us denote $p = p_0 + x_0$ and $q = q_0 + y_0$.
- Corresponding polynomial: $f(x, y) = N - (p_0 + x)(q_0 + y)$

RESULT: If $|p - p_0| < N^{\frac{1}{4}}$, one can solve f in polynomial time.

Sage: www.sagemath.org



- ./sage
- notebook()

Sage: www.sagemath.org

- $P = next\_prime(ZZ.random\_element(2^{511}, 2^{512}))$
- $d = e.inverse\_mod(N),\ a = gcd(a, b)$
- $M = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \Rightarrow M = matrix(ZZ, 2, 2, [1, 2, 3, 4])$

```
P=next_prime(ZZ.random_element(2^499,2^500))
Q=next_prime(ZZ.random_element(2^499,2^500))
N=P*Q
x0=P%2^230
P0=P-x0
X=ZZ.random_element(x0, 2*x0)
R.<x,y>=QQ[]
f=P0+x
print f(x0,0)%P
u=9
h=20
S=[]
for i in range (u+1):
    g=N^(u-i)*f^i
    S=union(S, g.monomials())
for i in range (u+1,h):
    g=f^u*x^(i-u)
    S=union(S, g.monomials())
```

```
dim=Set(S).cardinality()
M=Matrix(ZZ, dim, dim, range(dim*dim))

print 'Dimension of the Lattice', dim

for i in range (u+1):
    g=N^(u-i)*f^i
    g=g(x*X, 0)
    for j in range(dim):
        M[i,j]= g.coefficient(R(S[j]))(0,0)


for i in range (u+1,h):
    g=f^u*x^(i-u)
    g=g(x*X, 0)
    for j in range(dim):
        M[i,j]= g.coefficient(R(S[j]))(0,0)

tt=cputime()
```

```
M=M.LLL()
print cputime(tt)

M3=[]
f=0
for j in range(dim):
    f=f+(M[0][j]/S[j](X,X))*S[j]


R.<z>=QQ[]
f=f(z,0)
print f.roots() ,x0
```

```
0
Dimension of the Lattice 20
0.264
[(1035632721196309298413647286273275993437144247682563096913968041944541
, 1)]
1035632721196309298413647286273275993437144247682563096913968041944541
```

Reference

A. May. Using LLL-reduction for solving RSA and factorization problems. Technical report, LLL+25 Conference in honour of the 25th birthday of the LLL algorithm, Technische Universität Darmstadt, 2007. Available at `http://www.informatik.tu-darmstadt.de/KP/alex.html`.

```
#Congruential Crypto System
q=122430513841
f=231231
g=195698
f1=f.inverse_mod(q)
print f1
h=(f1*g).mod(q)
print h

m=123456
r=101010
e=(r*h+m).mod(q)
print e

a=(f*e).mod(q)
print a
print (f.inverse_mod(g)*a).mod(g)
```

```
49194372303
39245579300
18357558717
48314309316
123456
```

```
M=Matrix(ZZ,2,2,[q,0,
                 h,1])
M=M.LLL()
```

```
print M[0]
print g,f

(-195698, -231231)
195698 231231
```


Thank You!