Applications of Lattice Reduction in Cryptography

Abderrahmane Nitaj

University of Caen Basse Normandie, France

Kuala Lumpur, Malaysia, June 27, 2014

ماليزيا

Contents

Contents

Linear diophantine equation

- We want to solve
$$x_1e_1 + \ldots + x_ne_n = S,$$
for small values with $\max(|x_1|, \cdots, |x_n|) < X$.
- We transform the equation $x_1e_1 + \ldots + x_ne_n - S = 0$ using the matrix
$$M(L) = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & e_1 \\ 0 & 1 & 0 & \cdots & 0 & e_2 \\ 0 & 0 & 1 & \cdots & 0 & e_3 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 & e_n \\ 0 & 0 & 0 & \vdots & 0 & -S \end{pmatrix}.$$

Linear diophantine equation

- From the matrix, we deduce the vectors

$$
\begin{array}{rcl}
b_1 & = & (1, 0, 0, \cdots, 0, e_1), \\
b_2 & = & (0, 1, 0, \cdots, 0, e_2), \\
b_3 & = & (0, 0, 1, \cdots, 0, e_3), \\
\vdots & \vdots & \qquad\qquad \vdots \\
b_n & = & (0, 0, 0, \cdots, 1, e_n), \\
b_{n+1} & = & (0, 0, 0, \cdots, 0, -S),
\end{array}
$$

Linear diophantine equation

- We have

$$v_0 = x_1 b_1 + \dots x_{n+1} b_{n+1} = (x_1, x_2, \dots, x_n, x_1 e_1 + \dots + x_n e_n - S).$$

- We define the lattice

$$\mathcal{L} := \{v = a_1 b_1 + \dots a_{n+1} b_{n+1} \mid a_i \in \mathbb{Z}\}.$$

- Then $\dim(\mathcal{L}) = n + 1$ and $\det(\mathcal{L}) = S$.
- We have

$$\|v_0\| = \sqrt{x_1^2 + \dots + x_n^2} \leq \sqrt{nX^2} = X\sqrt{n}.$$

Linear diophantine equation

- By applying the LLL algorithm to the lattice $\mathcal{L}$, it outputs a basis $(u_1, \ldots, u_{n+1})$ such that

$$\|u_1\| \leq 2^{\frac{n}{4}} \det(L)^{\frac{1}{n+1}}.$$

- To find v_0 among the vectors $(u_1, \ldots, u_{n+1})$, we set

$$\|v_0\| \leq X\sqrt{n} \leq 2^{\frac{n}{4}} \det(L)^{\frac{1}{n+1}}.$$

- Since $\det(L) = S$, we get

$$X \leq \frac{2^{\frac{n}{4}}}{\sqrt{n}} S^{\frac{1}{n+1}}.$$

Linear diophantine equation

Theorem

Let $x_1e_1 + \ldots + x_ne_n = S$ *be a linear equation with* $\max(|x_1|, \cdots, |x_n|) < X$. *If*

$$X \leq \frac{2^{\frac{n}{4}}}{\sqrt{n}} S^{\frac{1}{n+1}},$$

then one can solve the equation in polynomial time.

The LLL algorithm has been applied to break the knapsack cryptosystem.

Example

Let $340x_1 + 257x_2 + 378x_3 + 251x_4 = 9138$ be a linear equation with $\max(|x_1|, \cdots, |x_4|) < X$.

- We have $\frac{2^{\frac{n}{4}}}{\sqrt{n}} S^{\frac{1}{n+1}} \approx 6.196$,
- Set

$$b_1 := [1, 0, 0, 0, 340]; b_2 := [0, 1, 0, 0, 257]; b_3 := [0, 0, 1, 0, 378];$$

$$b_4 := [0, 0, 0, 1, 251]; b_5 := [0, 0, 0, 0, -9138].$$

- Applying the LLL algorithm, we get the vectors

$$[3, -2, -2, 1, 1]; [0, 1, -2, 2, 3]; [0, 0, -2, 3, -3];$$

$$[-4, -4, 3, 5, 1]; [9, 7, 8, 5, 0]$$

- The solution is $[9, 7, 8, 5, 0]$

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 $\mathcal{P} = \mathbb{Z}[X]/(X^N - 1)$

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 $\mathcal{P} = \mathbb{Z}[X]/(X^N - 1)$

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 $\mathcal{P} = \mathbb{Z}[X]/(X^N - 1)$

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 $\mathcal{P} = \mathbb{Z}[X]/(X^N - 1)$

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

The lattice

- Using $h \equiv f_q * g \pmod{q}$, we get $h * f \equiv g \pmod{q}$.
- Hence $f * h - q * u = g$ for a polynomial $u \in \mathcal{P}$.
- Consider the lattice

$$\mathcal{L} = \left\{(f, g) \in \mathcal{P}^2 \,|\exists u \in \mathcal{P},\ f * h - q * u = g\right\}.$$

- Using a matrix, we get

$$(f, -u) * \begin{bmatrix} 1 & h \\ 0 & q \end{bmatrix} = (f, g).$$

NTRU

Using $f = (f_0, f_1, \cdots, f_{N-1})$, $h = (h_0, h_1, \cdots, h_{N-1})$ and $u = (u_0, u_1, \cdots, u_{N-1})$, we get

$$\left[\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline\hline -u_1 \\ -u_2 \\ \vdots \\ -u_{N-1} \end{array}\right] * \left[\begin{array}{cccc||cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline\hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right] = \left[\begin{array}{c} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \\ \hline\hline g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{array}\right]$$

NTRU

Hence the matrix of the lattice is

$$\left[\begin{array}{cccc||cccc} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 \\ \hline\hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right].$$

with determinant $\dim(\mathcal{L})$ and $\det(\mathcal{L}) = q^N$.

NTRU

The lattice

- Since f and g have d_f and d_g coefficient that are equal to ± 1 and the the other coefficients are equal to 0, then

$$\|(f,g)\| = \left(\sum_{i=0}^{N-1} f_i^2 + \sum_{i=0}^{N-1} g_i^2\right)^{1/2} \approx \sqrt{d_f + d_g}.$$

- Applying the LLL algorithm, we get a basis with a vector v_1 such that

$$\|v_1\| \leq 2^{\frac{2N-1}{4}} \det(L)^{\frac{1}{2N}}.$$

- Hence, (f,g) will be among the vectors of the reduced basis if

$$\sqrt{d_f + d_g} \leq 2^{\frac{2N-1}{4}} \det(L)^{\frac{1}{2N}} \Longrightarrow d_f + d_g \leq 2^{\frac{2N-1}{2}} \det(L)^{\frac{1}{N}} = 2^{\frac{2N-1}{2}} \sqrt{q}.$$

NTRU

Theorem (Coppersmith, Shamir)

Let d_f be the number of ± 1 in f and d_g the number of ± 1 in g. If $d_f + d_g \leq 2^{\frac{2N-1}{2}}\sqrt{q}$, Then f and g can be found in polynomial time.

Contents

RSA

The Key Equation Problem

Given $N = pq$ and e satisfying $ed - k\phi(N) = 1$. Find d, k and $\phi(N)$.

Coppersmith's method

1. Define the norm of $f(x_1,\ldots,x_n)=\sum a_{i_1\ldots i_n}x_1^{i_1}\ldots x_n^{i_n}$ to be
$$\|f(x_1,\ldots,x_n)\|=\sqrt{\sum a_{i_1\ldots i_n}^2}.$$
2. Consider the congruence equation
$$f(x_1,\ldots,x_n)\equiv 0 \pmod{N}.$$
We want to find the solutions $(x_1^{(0)},\ldots,x_n^{(0)})$ such that $|x_i^{(0)}|<X_i$ for $i=1,\ldots,n$.
3. Find an expression of each X_i in terms of N so that $X_i=N^{\delta_i}$.

Coppersmith's method

1. Let l be the monomial in f with maximal weight. Usually, l, is called the leading monomial of f. If the coefficient of l is a_l, then we can assume that $\gcd(N, al) = 1$. Therefore, we can use $fa_l^{-1} \pmod N$ instead of f so that the coefficient of l becomes 1. Throughout this paper, we consider that the coefficient of l is 1.
2. Let m be a positive integer. Find the monomials $x_1^{i_1} \cdots x_n^{i_n}$ of f^m.
3. For $0 \leq k \leq m$, find the monomials $x_1^{i_1} \cdots x_n^{i_n}$ of f^{m-k}.
4. For $t \geq 0$, find the set

$$M_k = \bigcup_{0 \leq j \leq t} \left\{ x_1^{i_1+j} \cdots x_n^{i_n} \;\middle|\; x_1^{i_1} \cdots x_n^{i_n} \text{ is a monomial of } f^m \text{ and } \frac{x_1^{i_1} \cdots x_n^{i_n}}{l^k} \text{ is a monomial of } f^{m-k} \right\}.$$

Coppersmith's method

1. Similarly, for $t \geq 0$, find the set

$$M_{k+1} = \bigcup_{0\leq j\leq t} \left\{ x_1^{i_1+j}\cdots x_n^{i_n} \;\middle|\; x_1^{i_1}\cdots x_n^{i_n} \text{ monomial of } f^m \text{ and } \frac{x_1^{i_1}\cdots x_n^{i_n}}{l^{k+1}} \text{ monomial of } f^{m-(k+1)} \right\}.$$

2. For $0 \leq k \leq m$, find the set $M_k \backslash M_{k+1}$.
3. For $0 \leq k \leq m$, define the polynomials

$$g_{k,i_1,\ldots,i_n}(x_1,\ldots,x_n) = \frac{x_1^{i_1}\ldots x_n^{i_n}}{l^k} f(x_1,\ldots,x_n)^k N^{m-k}$$
$$\text{with} \quad x_1^{i_1}\ldots x_n^{i_n} \in M_k \backslash M_{k+1}.$$

4. Then $g_{k,i_1,\ldots,i_n}(x_1,\ldots,x_n) \equiv 0 \pmod{N^m}$.
5. For $0 \leq k \leq m$, define the polynomials $g_{k,i_1,\ldots,i_n}(x_1X_1,\ldots,x_nX_n)$.

Coppersmith's method

1. Let $\mathcal{L}$ denote the lattice spanned by the coefficient vectors of the polynomials $g_{k,i_1,\ldots,i_n}(x_1X_1,\ldots,x_nX_n)$. The ordering of the monomials is such that the matrix M is triangular.
2. Compute the dimension ω of the lattice $\mathcal{L}$ and its determinant
$$\det(\mathcal{L}) = N^{e_N}\cdot X_1^{e_1}\cdots X_n^{e_n}.$$
3. Compute e_N and the exponents e_i, $1\leq i\leq n$, in terms of m and t.
4. Let $t = m\tau$. Compute an approximation of e_N and the exponents e_i, $1\leq i\leq n$, in terms of m and τ by neglecting all terms of low degrees.
5. Apply the LLL algorithm to obtain a reduced basis $(b_1,\ldots,b_n)$ such that
$$\|b_1\|\leq 2^{\frac{\omega}{2}}\det(\mathcal{L})^{\frac{1}{\omega}}.$$

Coppermith's method

Theorem (Howgrave-Graham)

Let $h(x_1, \cdots, x_n) \in \mathbb{Z}[x_1, \cdots, x_n]$ *be a polynomial with at most* ω *monomials. Suppose that*

1. $h\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) \equiv 0 \pmod{N^m}$ *where* $|x_i^{(0)}| < X_i$ *for* $i = 1, \ldots, n$,
2. $h(x_1X_1, \cdots, x_nX_n) < \frac{N^m}{\sqrt{\omega}}$.

Then $h\left(x_1^{(0)}, \cdots, x_n^{(0)}\right) = 0$ *holds over the integers.*

Coppersmith's method

1. Combine Howgrave-Graham's bound $\|f(x_1X_1,\ldots,x_nX_n)\| < \frac{N^m}{\sqrt{\omega}}$ and LLL to form the inequation

$$2^{\frac{\omega}{2}}\det(\mathcal{L})^{\frac{1}{\omega}} < \frac{N^m}{\sqrt{\omega}}$$

2. Neglecting $2^{\frac{\omega}{2}}$ and $\sqrt{\omega}$, consider the condition

$$\det(\mathcal{L}) < N^{m\omega},$$

or equivalently,

$$N^{e_N}\cdot X_1^{e_1}\cdots X_n^{e_n} = N^{e_N}\cdot N^{e_1\delta_1}\cdots N^{e_n\delta_n} < N^{m\omega}.$$

3. Taking logarithms, we get the inequation

$$e_N + e_1\delta_1 + \ldots + e_n\delta_n < m\omega.$$

The next task is to solve this inequation.

Coppersmith's method: example

We want to solve the equation $ed - k\phi(N) = 1$.

1. We have $ed - k(N+1-p-q) = 1$ and then $-k(p+q) + (N+1)k + 1 \equiv 0 \pmod{e}$.
2. Set $k = -x$, $p+q = y$ and $N+1 = a$. Then the congruence becomes
$$f(x,y) = xy + ax + 1 \equiv 0 \pmod{e}.$$
3. Let $e = N^{\beta}$, $p+q = N^{1/2}$ and $d = N^{\delta}$. Then
$$k = \frac{ed-1}{\phi(N)} < \frac{ed}{\phi(N)} < d.$$
Hence $k = N^{\delta}$.
4. Let $l = xy$ be the leading monomial of $f(x,y) = xy + ax + 1$

Coppersmith's method: example

1. We have

$$
\begin{aligned}
f^m(x,y) &= (x(y+a)+1))^m \\
&= \sum_{i_1=0}^{m} \binom{m}{i_1} x^{i_1}(y+a)^{i_1} \\
&= \sum_{i_1=0}^{m} \binom{m}{i_1} x^{i_1} \sum_{i_2=0}^{i_1} \binom{i_1}{i_2} y^{i_2} a^{i_1-i_2} \\
&= \sum_{i_1=0}^{m} \sum_{i_1=0}^{i_1} \binom{m}{i_1} \binom{i_1}{i_2} x^{i_1} y^{i_2} a^{i_1-i_2}.
\end{aligned}
$$

2. Hence the monomials of f^m are in the form

$$
f^m = \left\{ x^{i_1} y^{i_2}, \quad i_1 = 0, \ldots, m, \quad i_2 = 0, \ldots, i_1 \right\}.
$$

Coppersmith's method: example

1. From the monomials of f^m, we easily deduce the monomials of f^{m-k}:
$$f^{m-k} = \{x^{i_1}y^{i_2}, \quad i_1 = 0, \ldots, m-k, \quad i_2 = 0, \ldots, i_1\}.$$
2. For $t \geq 0$ and $k \leq m$, let consider extra shifts in y, that is, we consider the set
$$M_k = \bigcup_{0 \leq j \leq t} \{x^{i_1}y^{i_2+j} \;\Big|\; x^{i_1}y^{i_2} \text{ monomial of } f^m \text{ and } \frac{x^{i_1}y^{i_2}}{l^k} \text{ monomial of } f^{m-k}\}.$$
3. Since $l = xy$, then
$$M_k = \bigcup_{0 \leq j \leq t} \{x^{i_1}y^{i_2+j} \;\Big|\; x^{i_1}y^{i_2} \text{ monomial of } f^m \text{ and } x^{i_1-k}y^{i_2-k} \text{ monomial of } f^{m-k}\}.$$

Coppersmith's method: example

1. Hence
$$x^{i_1}y^{i_2} \in M_k \Leftrightarrow i_1 = k, \ldots, m, \quad i_2 = k, \ldots, i_1 + t.$$
2. Similarly, for $t \geq 0$, we get
$$x^{i_1}y^{i_2} \in M_{k+1} \Leftrightarrow i_1 = k+1, \ldots, m, \quad i_2 = k+1, \ldots, i_1 + t.$$
3. For $0 \leq k \leq m$, we find that $x_1^{i_1}y^{i_2} \in M_k \setminus M_{k+1}$ if and only if
$$\begin{cases} i_1 = k, \ldots, m, \\ i_2 = k, \end{cases} \quad \text{or} \quad \begin{cases} i_1 = k, \\ i_2 = k+1, \ldots, i_1 + t. \end{cases}$$
4. For $0 \leq k \leq m$, we define the polynomials
$$g_{k,i_1,i_2}(x,y) = \frac{x^{i_1}y^{i_2}}{(xy)^k} f(x,y)^k e^{m-k}$$
$$\text{with} \quad x^{i_1}y^{i_2} \in M_k \setminus M_{k+1}.$$

Coppersmith's method: example

1. For $i_1 = k, \ldots, m$ and $i_2 = k$, the polynomials reduce to

$$g_{k,i_1,k}(x,y) = G_{k,i_1}(x,y) = x^{i_1-k} f(x,y)^k e^{m-k} \quad \text{for} \quad i_1 = k, \ldots, m.$$

For $i_1 = k$ and $i_2 = k+1, \ldots, i_1 + t = k + t$, the polynomials reduce to

$$g_{k,k,i_2}(x_1,y) = H_{k,i_2}(x,y) = y^{i_2-k} f(x,y)^k e^{m-k} \quad \text{for} \quad i_2 = k+1, \ldots, k+t.$$

2. For $0 \le k \le m$, we define the polynomials

$$G_{k,i_1}(xX,yY) = x^{i_1-k} X^{i_1-k} f(xX,yY)^k e^{m-k} \quad \text{for} \quad i_1 = k, \ldots, m,$$
$$H_{k,i_2}(xX,yY) = y^{i_2-k} Y^{i_2-k} f(xX,yY)^k e^{m-k} \quad \text{for} \quad i_2 = k+1, \ldots, k+t.$$

Coppersmith's method: example

	1	x	x^2	x^3	y	xy	x^2y	x^3y	xy^2	x^2y^2	x^3y^2	x^2y^3	x^3y^3	x^3y^4
$G_{0,0}$	e^3	0	0	0	0	0	0	0	0	0	0	0	0	0
$G_{0,1}$	0	Xe^3	0	0	0	0	0	0	0	0	0	0	0	0
$G_{0,2}$	0	0	X^2e^3	0	0	0	0	0	0	0	0	0	0	0
$G_{0,3}$	0	0	0	X^3e^3	0	0	0	0	0	0	0	0	0	0
$H_{0,1}$	0	0	0	0	Ye^3	0	0	0	0	0	0	0	0	0
$G_{1,1}$	$\circledast$	$\circledast$	0	0	0	XYe^2	0	0	0	0	0	0	0	0
$G_{1,2}$	0	$\circledast$	$\circledast$	0	0	0	X^2Ye^2	0	0	0	0	0	0	0
$G_{1,3}$	0	0	$\circledast$	$\circledast$	0	0	0	X^3Ye^2	0	0	0	0	0	0
$H_{1,2}$	0	0	0	0	$\circledast$	$\circledast$	0	0	XY^2e^2	0	0	0	0	0
$G_{2,2}$	$\circledast$	$\circledast$	$\circledast$	0	0	$\circledast$	$\circledast$	0	0	X^2Y^2	0	0	0	0
$G_{2,3}$	0	$\circledast$	$\circledast$	$\circledast$	0	0	$\circledast$	$\circledast$	0	0	X^3Y^2e	0	0	0
$H_{2,3}$	0	0	0	0	$\circledast$	$\circledast$	$\circledast$	0	$\circledast$	$\circledast$	0	X^2Y^3e	0	0
$G_{3,3}$	$\circledast$	$\circledast$	$\circledast$	$\circledast$	0	$\circledast$	$\circledast$	$\circledast$	0	$\circledast$	$\circledast$	0	X^3Y^3	0
$H_{3,4}$	0	0	0	0	$\circledast$	$\circledast$	$\circledast$	0	$\circledast$	$\circledast$	$\circledast$	$\circledast$	$\circledast$	X^3Y^4

Table: The coefficient matrix for the case $m = 3$, $t = 1$.

Coppersmith's method: example

1. The dimension of $\mathcal{L}$ is

$$\omega = \sum_{k=0}^{m} \sum_{i_1=k}^{m} 1 + \sum_{k=0}^{m} \sum_{i_2=k+1}^{k+t} 1 = \frac{1}{2}(m+1)(m+2t+2).$$

2. The determinant of $\mathcal{L}$ is

$$\det(\mathcal{L}) = e^{e_0} X^{e_1} Y^{e_2}.$$

3. From the construction of the polynomials $G_{k,i_1}(xX, yY)$ and $H_{k,i_2}(xX, yY)$, we get

$$e_0 = \sum_{k=0}^{m} \sum_{i_1=k}^{m} (m-k) + \sum_{k=0}^{m} \sum_{i_2=k+1}^{k+t} (m-k) = \frac{1}{6} m(m+1)(2m+3t+4).$$

Coppersmith's method: example

1. Similarly, we have

$$e_1 = \sum_{k=0}^{m} \sum_{i_1=k}^{m} i_1 + \sum_{k=0}^{m} \sum_{i_2=k+1}^{k+t} k = \frac{1}{6} m(m+1)(2m+3t+4),$$

and

$$e_2 = \sum_{k=0}^{m} \sum_{i_1=k}^{m} k + \sum_{k=0}^{m} \sum_{i_2=k+1}^{k+t} i_2 = \frac{1}{6}(m+1)(m^2+3mt+3t^2+2m+3t).$$

Coppersmith's method: example

1. Let $t = m\tau$. Then

$$\omega = \frac{1}{2}(m+1)(m+2m\tau+2) = \frac{1}{2}(1+2\tau)m^2 + o\left(m^2\right),$$

and

$$e_0 = \frac{1}{6}m(m+1)(2m+3m\tau+4) = \frac{1}{6}(2+3\tau)m^3 + o\left(m^3\right).$$

Similarly, we have

$$e_1 = \frac{1}{6}m(m+1)(2m+3m\tau+4) = \frac{1}{6}(2+3\tau)m^3 + o\left(m^3\right),$$

and

$$\begin{aligned} e_2 &= \frac{1}{6}(m+1)(m^2+3m^2\tau+3m^2\tau^2+2m+3m\tau) \\ &= \frac{1}{6}(1+3\tau+3\tau^2)m^3 + o\left(m^3\right). \end{aligned}$$

Coppersmith's method: example

1. Apply the LLL algorithm to obtain a reduced basis $(b_1, \ldots, b_n)$ such that

$$\|b_1\| \leq 2^{\frac{\omega}{2}} \det(\mathcal{L})^{\frac{1}{\omega}}.$$

2. Combining Howgrave-Graham's bound $\|f(x_1X_1, \ldots, x_nX_n)\| < \frac{e^m}{\sqrt{\omega}}$ and LLL, we get the inequation

$$2^{\frac{\omega}{2}} \det(\mathcal{L})^{\frac{1}{\omega}} < \frac{e^m}{\sqrt{\omega}}$$

3. Neglecting $2^{\frac{\omega}{2}}$ and $\sqrt{\omega}$, we get

$$\det(\mathcal{L}) < e^{m\omega},$$

or equivalently,

$$e^{e_0} X^{e_1} Y^{e_2} < e^{m\omega}.$$

Coppersmith's method: example

1. Since $e = N^{\beta}$, $Y = N^{1/2}$ and $X = N^{\delta}$. Then
$$N^{\beta e_0} \cdot N^{\delta e_1} \cdot N^{e_2/2} < N^{m\beta\omega}.$$
2. Taking logarithms, we get the inequation $\beta e_0 + \delta e_1 + \frac{e_2}{2} < m\beta\omega$.
3. Plugging the values of e_0, e_1 and ω, we get
$$\frac{1}{6}(2+3\tau)\beta + \frac{1}{6}(2+3\tau)\delta + \frac{1}{12}(1+3\tau+3\tau^2) - \frac{1}{2}(1+2\tau)\beta < 0.$$
4. The optimal value for τ in the left side is $\tau = \beta - \delta - \frac{1}{2}$, which leads to
$$-12\delta^2 + 4(1+6\beta)\delta - 12\beta^2 + 4\beta + 1 < 0.$$

Coppersmith's method: example

1. Solving for δ, we get

$$\delta < \beta + \frac{1}{6} - \frac{1}{3}\sqrt{6\beta + 1}.$$

2. A typical example is the case $e \approx N$, that is $\beta = 1$. Then, the former bound gives $d < \frac{7}{6} - \frac{1}{3}\sqrt{7} \approx 0.284$. This is a famous bound found by Boneh and Durfee.

Theorem (Boneh,Durfee, 1998)

Let (N, e) be an RSA public key with private decryption exponent d. If $d < \frac{7}{6} - \frac{1}{3}\sqrt{7}$, then one can factor the RSA modulus N.

Thank you

Terima kasih