Lattice Reduction Techniques To Attack RSA

David Wong

March 2015

University of Bordeaux

RSA

(e, N) is the **public key**

(d, N) is the **private key**

Encrypt a message m:

$$c = m^e \pmod{N}$$

Decrypt a ciphertext c:

$$m = c^d \pmod{N}$$

$$N = p \times q$$

$$(e, N) \quad (d, N)$$


ATTACKS

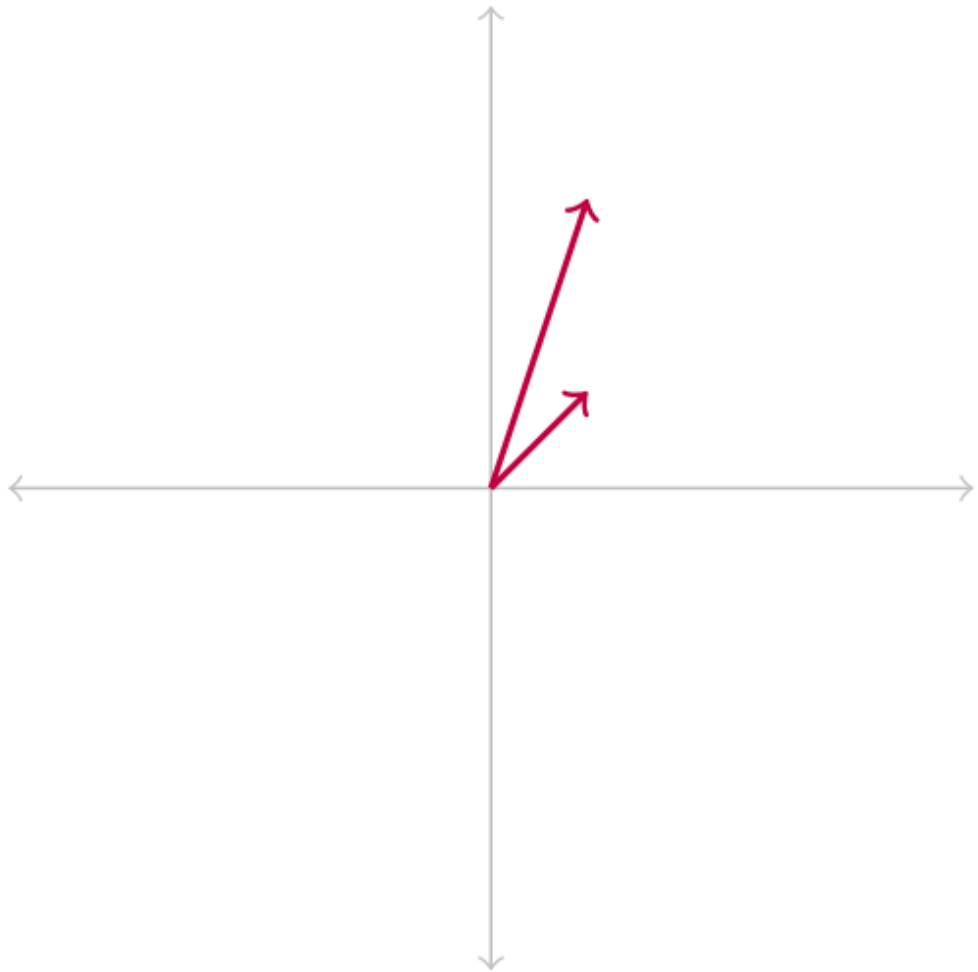
Attacks on the Implementation or the Mathematics.

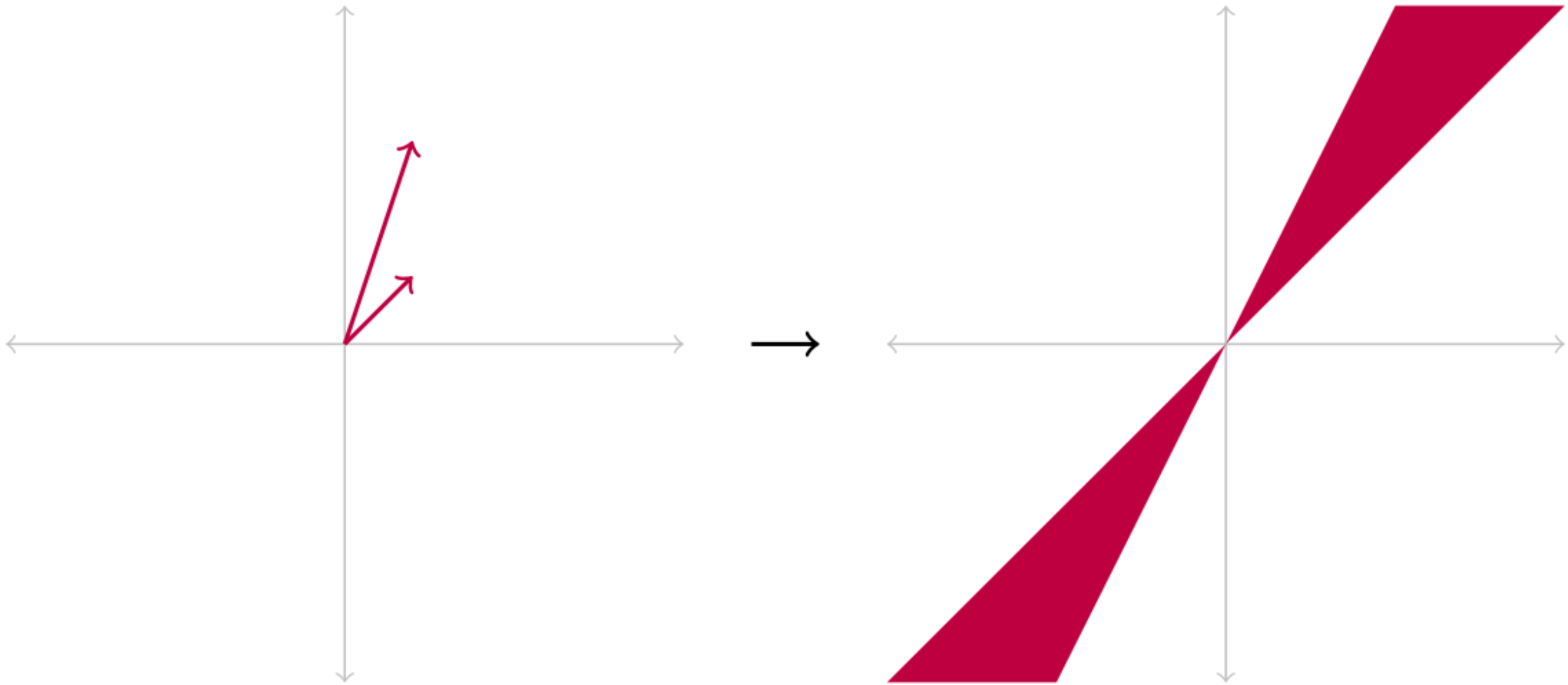
- Recover the plaintext
- Recover the private key

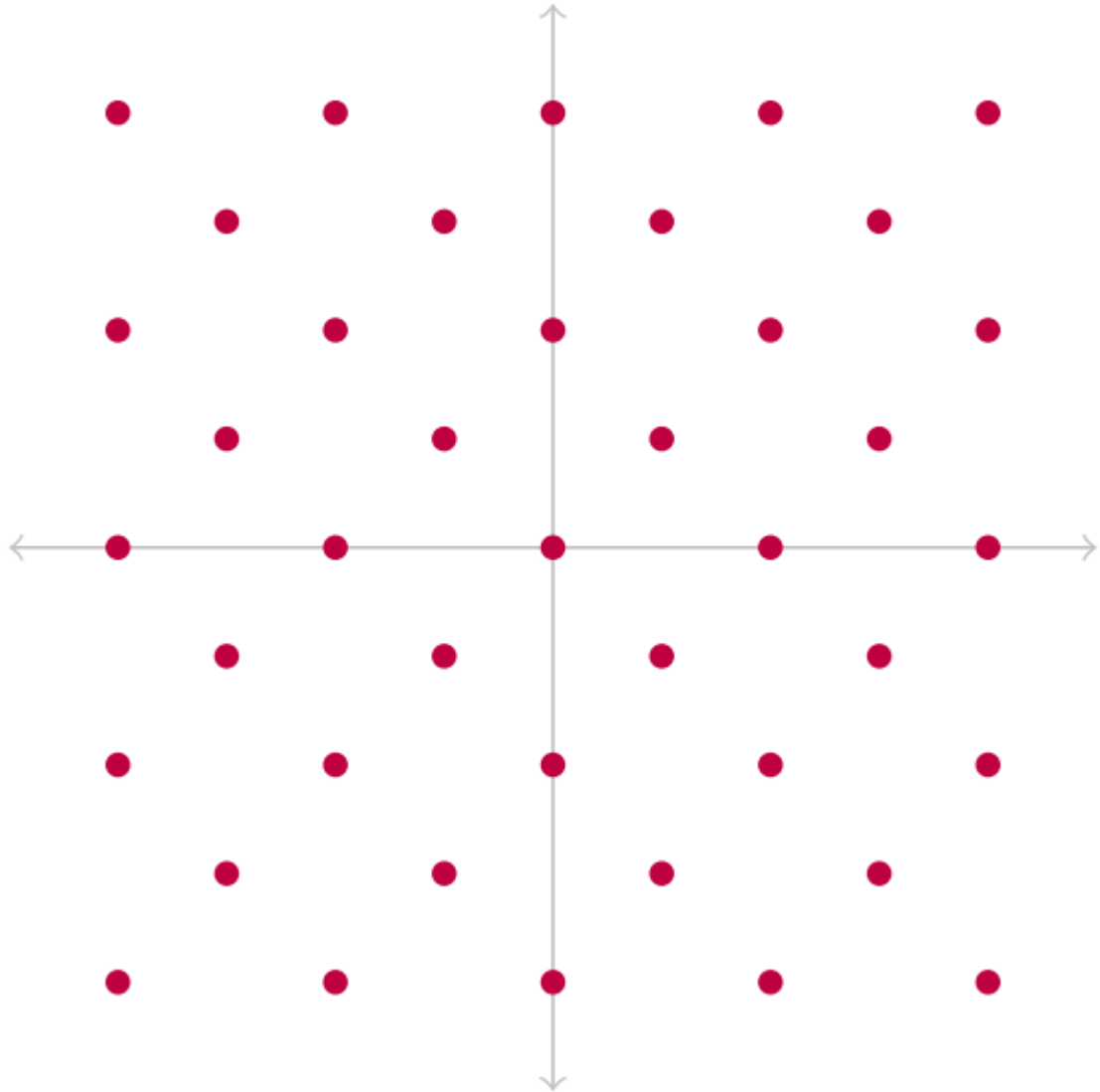
A Relaxed Model

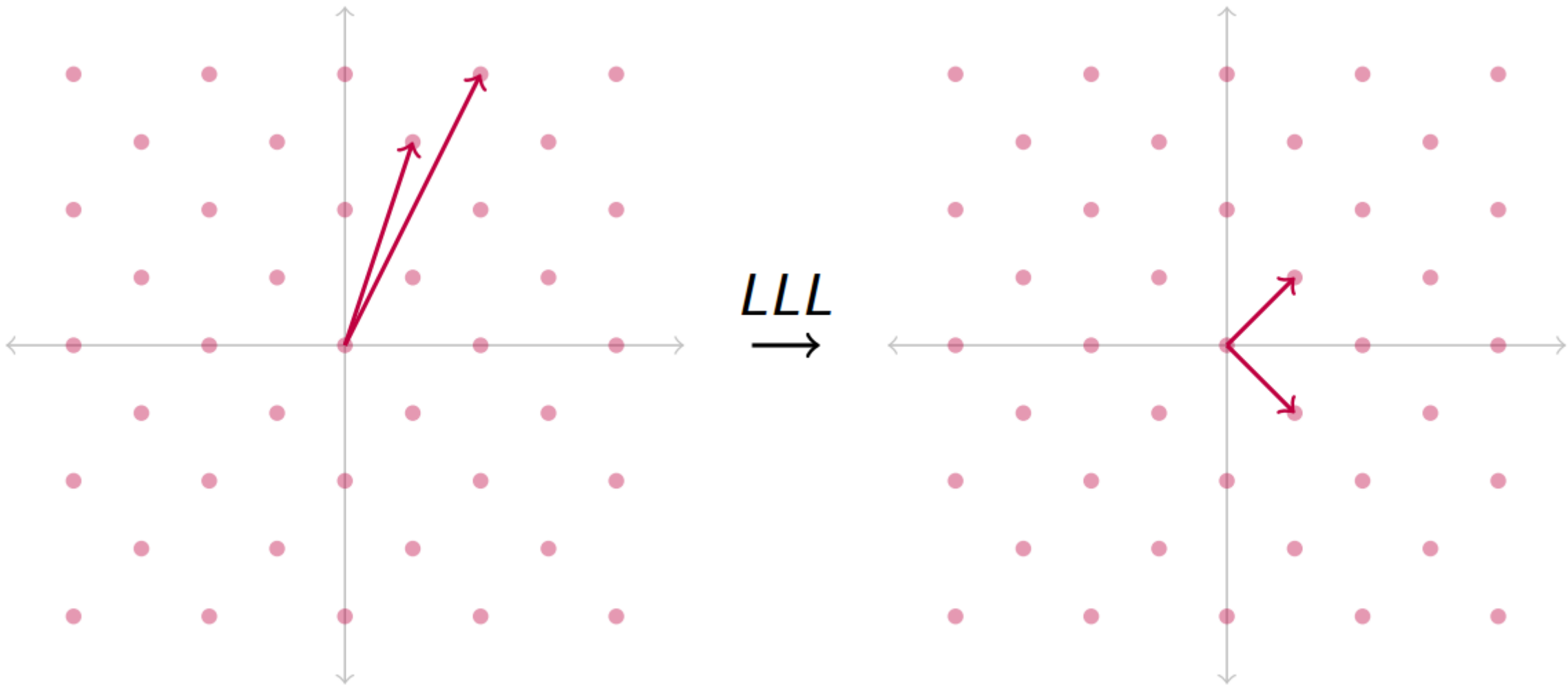
- We know a part of the message
- We know an approximation of one of the prime
- The private exponent is too small

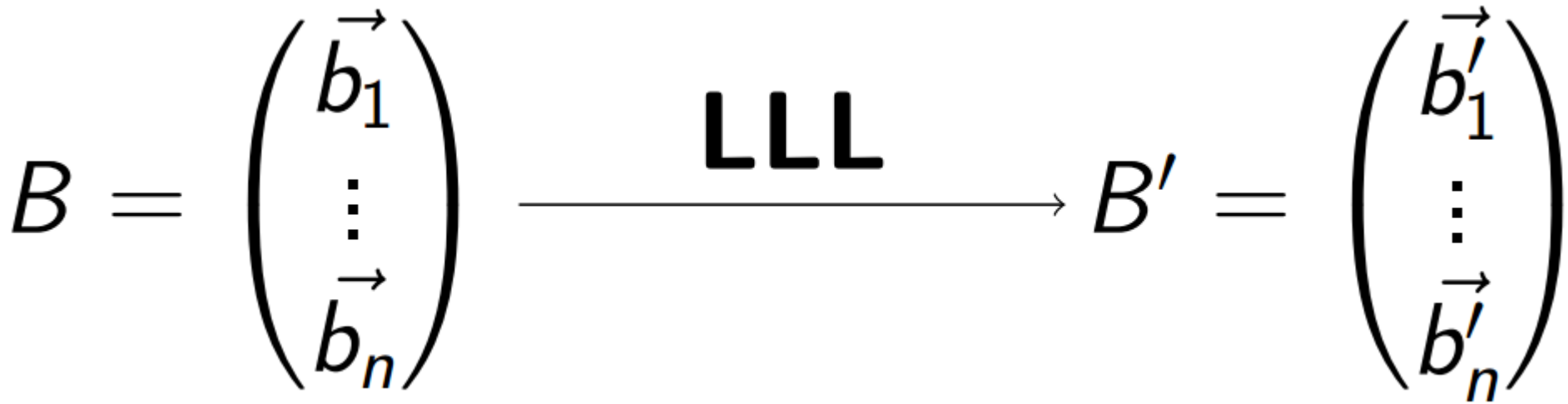
LATTICE








LLL


$B = \begin{pmatrix} \vec{b_1} \\ \vdots \\ \vec{b_n} \end{pmatrix}$
LLL
$B' = \begin{pmatrix} \vec{b'_1} \\ \vdots \\ \vec{b'_n} \end{pmatrix}$

$$B = \begin{pmatrix} \vec{b_1} \\ \vdots \\ \vec{b_n} \end{pmatrix} \xrightarrow{\textbf{LLL}} B' = \begin{pmatrix} \vec{b'_1} \\ \vdots \\ \vec{b'_n} \end{pmatrix}$$

$$\|b'_1\| \leq \|b'_2\| \leq \ldots \leq \|b'_i\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \cdot det(L)^{\frac{1}{n+1-i}}$$


COPPERSMITH

$$c = m^e \pmod{N}$$

$$c = m^e \pmod{N}$$

$$m = m_0 + x_0$$

« le password du jour : cupcake »

$$c = m^e \pmod{N}$$

$$m = m_0 + x_0$$

« le password du jour : cupcake »

$$f(x) = c - (m_0 + x)^e \pmod{N}$$

$$f(x) = 0 \pmod{N} \text{ with } |x| < X$$

$$g(x) = 0 \text{ over } \mathbb{Z}$$

HOWGRAVE-GRAHAM

Theorem *Let $g(x)$ be an univariate polynomial with n monomials. Further, let m be a positive integer. Suppose that*

$$g(x_0) = 0 \pmod{N} \quad \text{where} \quad |x_0| \leq X \tag{1}$$

$$\|g(xX)\| < \frac{N}{\sqrt{n}} \tag{2}$$

Then $g(x_0) = 0$ holds over the integers.

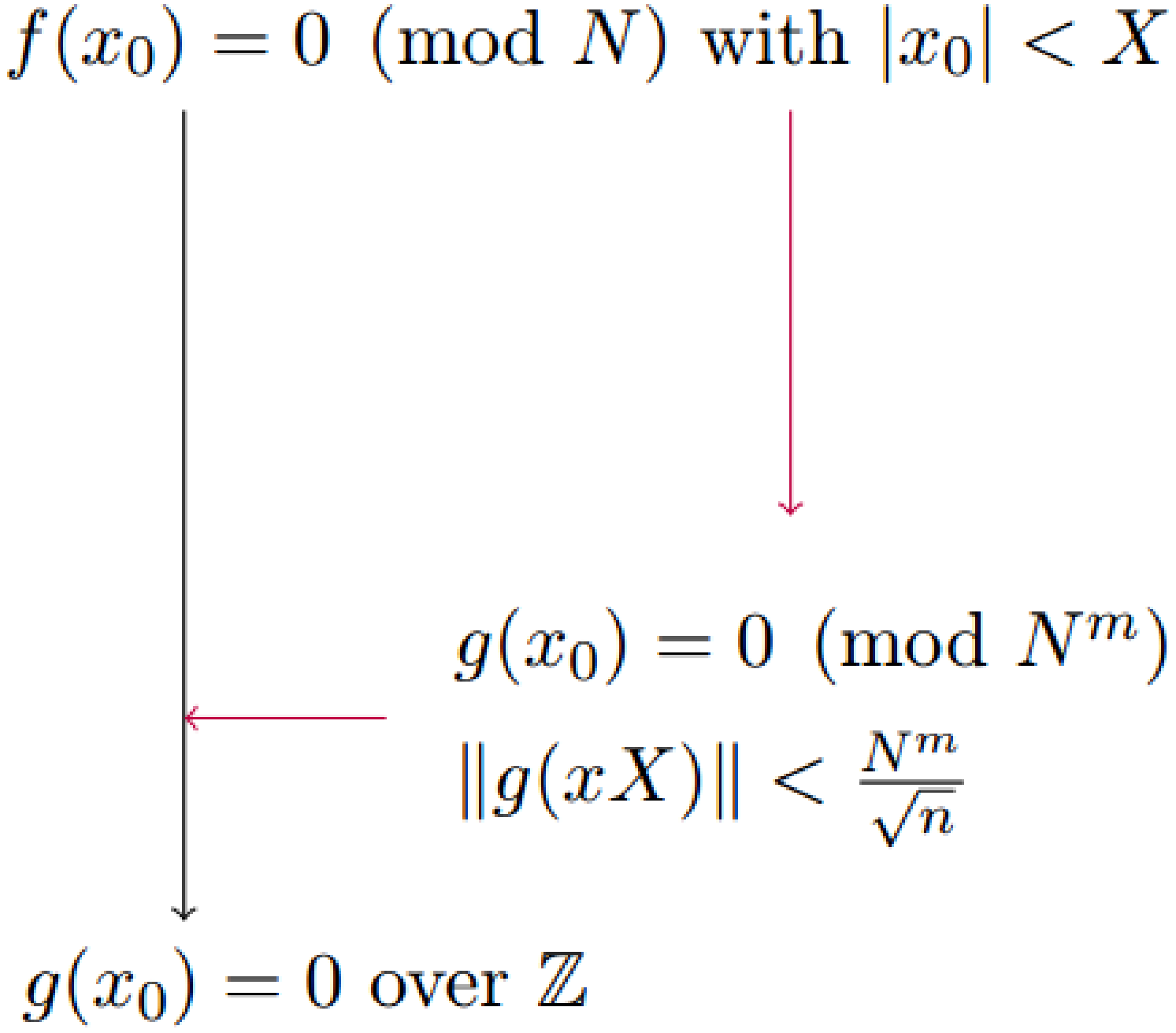
HOWGRAVE-GRAHAM

Theorem *Let $g(x)$ be an univariate polynomial with n monomials. Further, let m be a positive integer. Suppose that*

$$g(x_0) = 0 \pmod{N^m} \quad \text{where} \quad |x_0| \leq X \tag{1}$$

$$\|g(xX)\| < \frac{N^m}{\sqrt{n}} \tag{2}$$

Then $g(x_0) = 0$ holds over the integers.


$f(x_0) = 0 \pmod{N}$ with $|x_0| < X$
$g(x_0) = 0 \pmod{N^m}$
$\|g(xX)\| < \frac{N^m}{\sqrt{n}}$
$g(x_0) = 0$ over $\mathbb{Z}$

LLL reduction:

- It only does integer linear operations on the basis vectors
- The shortest vector of the output basis is bound

$$B = \begin{pmatrix} \vec{b_1} \\ \vdots \\ \vec{b_n} \end{pmatrix} \xrightarrow{\textbf{LLL}} B' = \begin{pmatrix} \vec{b'_1} \\ \vdots \\ \vec{b'_n} \end{pmatrix}$$

$$\|b'_1\| \leq \|b'_2\| \leq \ldots \leq \|b'_i\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \cdot det(L)^{\frac{1}{n+1-i}}$$

$$B = \begin{pmatrix} \vec{b_1} \\ \vdots \\ \vec{b_n} \end{pmatrix} \xrightarrow{\textbf{LLL}} B' = \begin{pmatrix} \vec{b'_1} \\ \vdots \\ \vec{b'_n} \end{pmatrix}$$

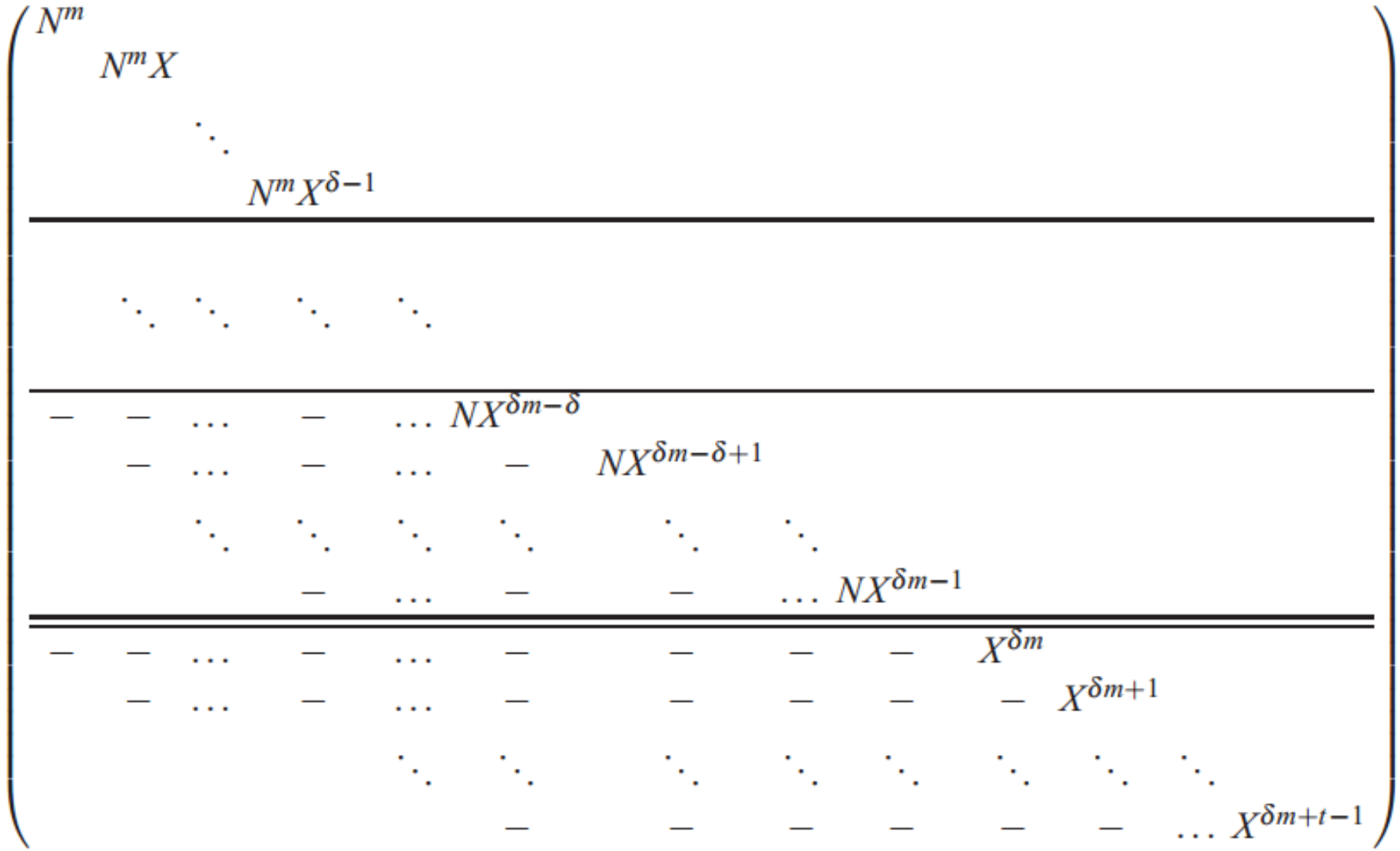
$$\|b'_1\| \leq \|b'_2\| \leq \ldots \leq \|b'_i\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \cdot det(L)^{\frac{1}{n+1-i}}$$

$$\|b'_1\| \leq 2^{\frac{n(n-1)}{4(n)}} \cdot det(L)^{\frac{1}{n}}$$

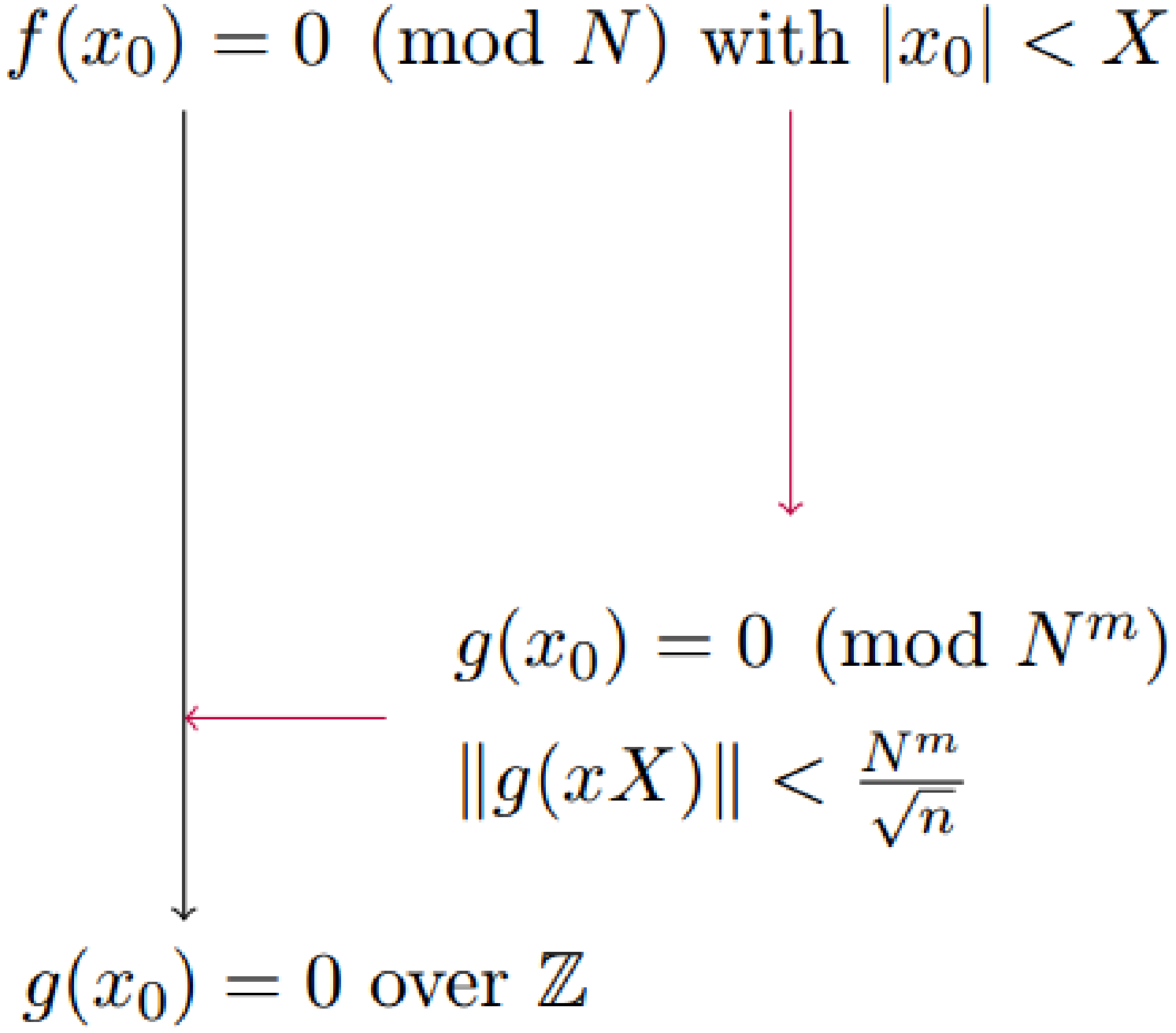
$$g_{i,j}(x) = x^j \cdot N^i \cdot f^{m-i}(x)$$
$$\text{for } i = 0, \ldots, m-1, \quad j = 0, \ldots, \delta - 1$$
$$h_i(x) = x^i \cdot f^m(x)$$
$$\text{for } i = 0, \ldots, t-1$$

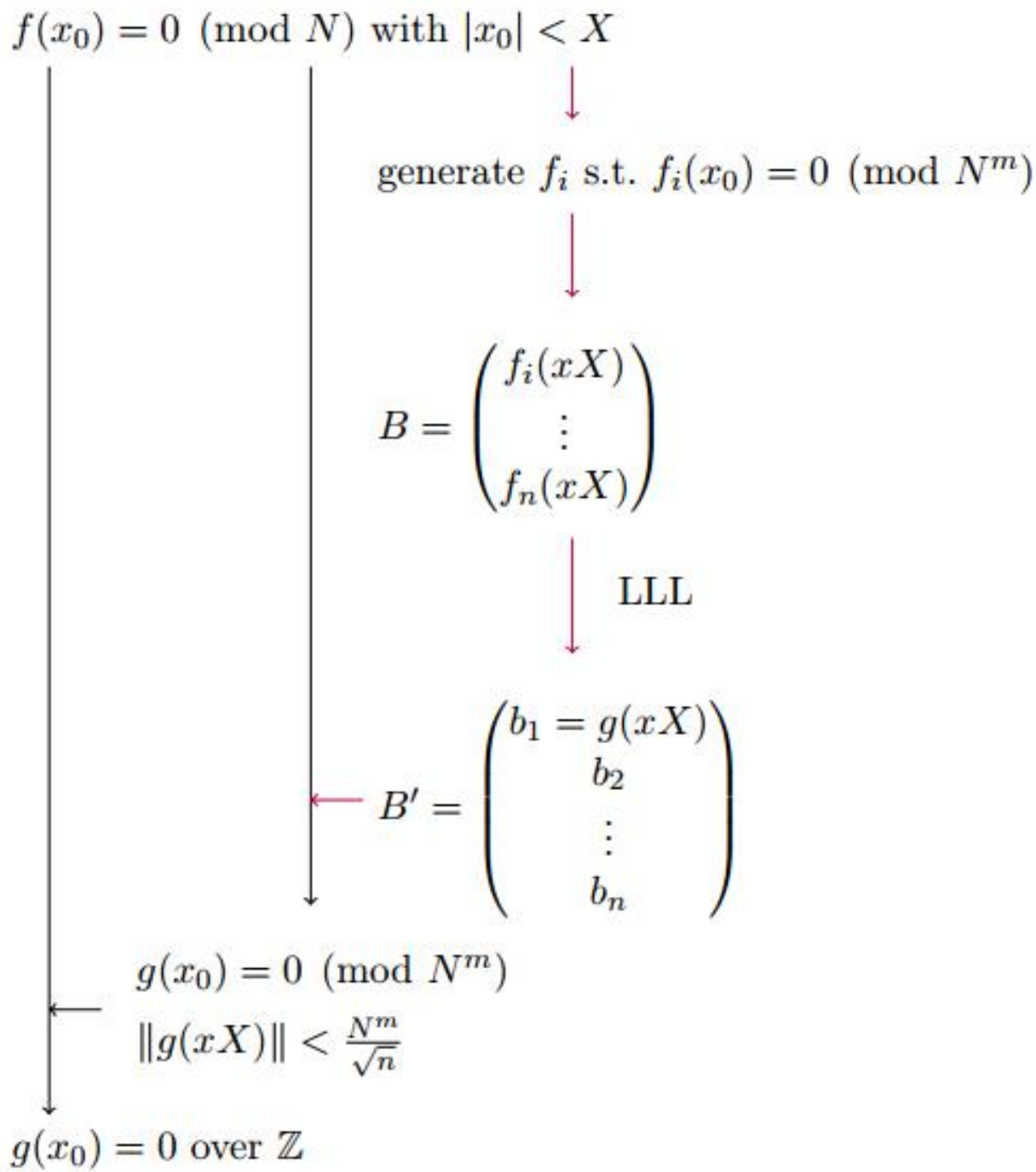
Those polynomials achieve two things:

- They have the same root x_0 but modulo N^m
- Each iteration introduce a new monomial



$$det(L) = N^{\frac{1}{2}\delta m(m+1)} X^{\frac{1}{2}n(n-1)}$$


$f(x_0) = 0 \pmod{N}$ with $|x_0| < X$
$g(x_0) = 0 \pmod{N^m}$
$\|g(xX)\| < \frac{N^m}{\sqrt{n}}$
$g(x_0) = 0$ over $\mathbb{Z}$


$f(x_0) = 0 \pmod N$ with $|x_0| < X$
generate f_i s.t. $f_i(x_0) = 0 \pmod{N^m}$
$B = \begin{pmatrix} f_i(xX) \\ \vdots \\ f_n(xX) \end{pmatrix}$
LLL
$B' = \begin{pmatrix} b_1 = g(xX) \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$
$g(x_0) = 0 \pmod{N^m}$
$\|g(xX)\| < \frac{N^m}{\sqrt{n}}$
$g(x_0) = 0$ over $\mathbb{Z}$

$$\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot det(L)^{\frac{1}{n}}$$

$$\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot det(L)^{\frac{1}{n}}$$

$$\|g(xX)\| < \frac{N^m}{\sqrt{n}}$$

$$\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot det(L)^{\frac{1}{n}}$$

$$\|b1\| = \|g(xX)\| < \frac{N^m}{\sqrt{n}}$$

$$\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot det(L)^{\frac{1}{n}}$$

$$\|b1\| = \|g(xX)\| < \frac{N^m}{\sqrt{n}}$$

$$det(L) < 2^{-\frac{n(n-1)}{4}} \cdot n^{-\frac{n}{2}} \cdot N^{n \cdot m}$$

$$\|b_1\| \leq 2^{\frac{n-1}{4}} \cdot det(L)^{\frac{1}{n}}$$

$$\|b1\| = \|g(xX)\| < \frac{N^m}{\sqrt{n}}$$

$$det(L) < 2^{-\frac{n(n-1)}{4}} \cdot n^{-\frac{n}{2}} \cdot N^{n \cdot m}$$

$$det(L) < N^{m \cdot n}$$

COPPERSMITH

Theorem *Let N be an integer of unknown factorization, which has a divisor $b \geq N^{\beta}$, $0 < \beta \leq 1$. Let $f(x)$ be a univariate monic polynomial of degree δ and let $c \geq 1$.*
Then we can find in time $\mathcal{O}(c\delta^5 log^9(N))$ all solutions x_0 of the equation

$$f(x) = 0 \pmod{b} \quad with \quad |x_0| \leq c \cdot N^{\frac{\beta^2}{\delta}}$$


BONEH-DURFEE

$$N = p \times q$$

$$(e, N) \quad (d, N)$$

$$e \cdot d = 1 \pmod{\varphi(N)}$$

$$e \cdot d = 1 \pmod{\varphi(N)}$$

$$\Longrightarrow e \cdot d = k \cdot \varphi(N) + 1$$

$$
\begin{aligned}
& e \cdot d = 1 \pmod{\varphi(N)} \\
\Longrightarrow\ & e \cdot d = k \cdot \varphi(N) + 1 \\
\Longrightarrow\ & k \cdot \varphi(N) + 1 = 0 \pmod{e}
\end{aligned}
$$

$$
\begin{aligned}
& e \cdot d = 1 \pmod{\varphi(N)} \\
\Longrightarrow\ & e \cdot d = k \cdot \varphi(N) + 1 \\
\Longrightarrow\ & k \cdot \varphi(N) + 1 = 0 \pmod{e} \\
\Longrightarrow\ & k \cdot (N + 1 - p - q) + 1 = 0 \pmod{e}
\end{aligned}
$$

$$\underbrace{k}_{x} \cdot (\underbrace{N+1}_{A} \underbrace{-p-q}_{y}) + 1 = 0 \pmod{e}$$

$$\underbrace{k}_{x} \cdot (\underbrace{N+1}_{A} \underbrace{-p-q}_{y}) + 1 = 0 \pmod{e}$$

$$f(x,y) = x(A+y)+1$$

HOWGRAVE-GRAHAM

Theorem *Let $g(x)$ be an bivariate polynomial with at most n monomials. Further, let m be a positive integer. Suppose that*

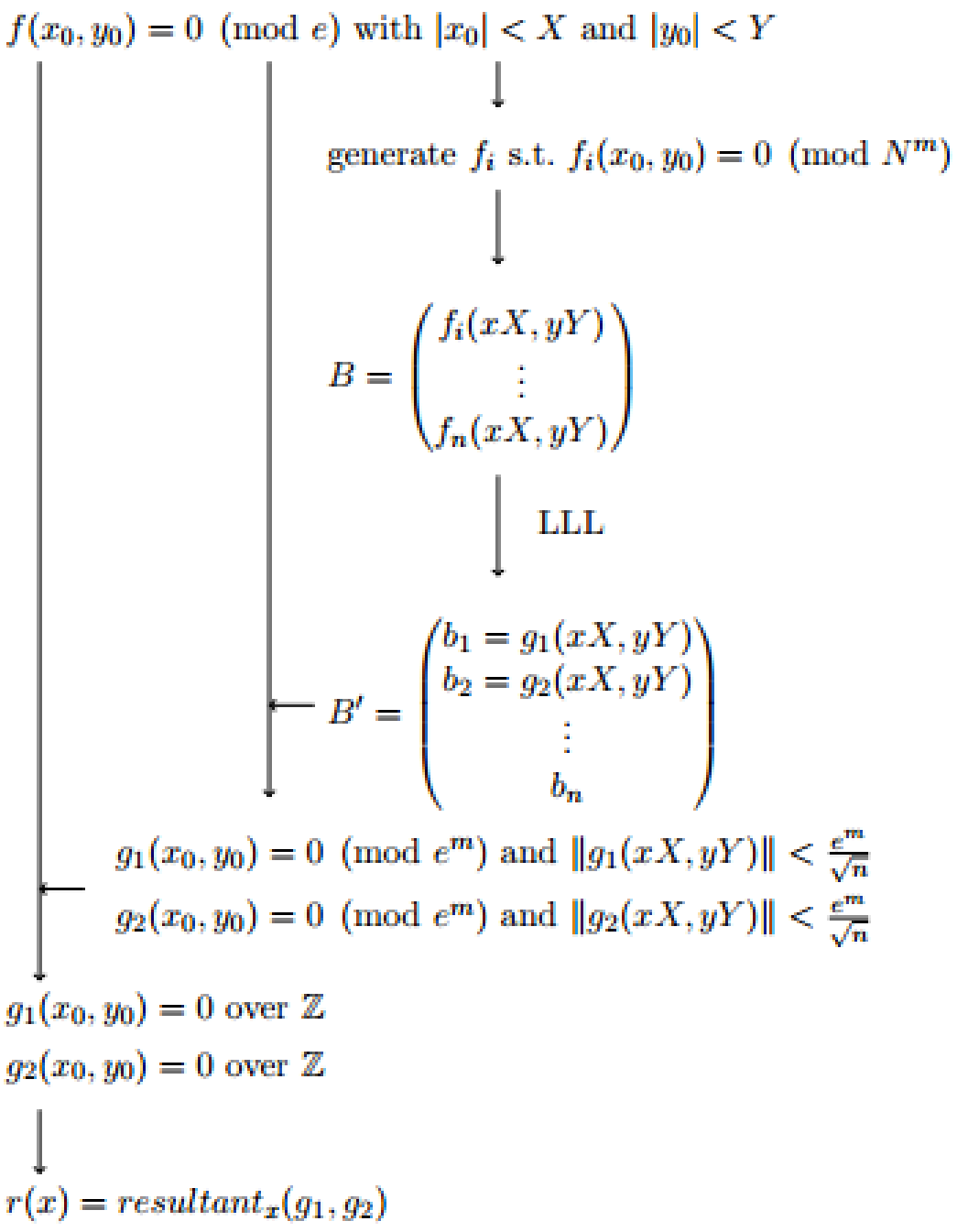
$$g(x_0, y_0) = 0 \pmod{e^m} \quad \text{where} \quad |x_0| \leq X \text{ and } |y_0| \leq Y \tag{1}$$

$$\|g(xX, yY)\| < \frac{e^m}{\sqrt{n}} \tag{2}$$

Then $g(x_0, y_0) = 0$ holds over the integers.

$$B = \begin{pmatrix} \vec{b_1} \\ \vdots \\ \vec{b_n} \end{pmatrix} \xrightarrow{\textbf{LLL}} B' = \begin{pmatrix} \vec{b'_1} \\ \vdots \\ \vec{b'_n} \end{pmatrix}$$

$$\|b'_1\| \leq \|b'_2\| \leq \ldots \leq \|b'_i\| \leq 2^{\frac{n(n-1)}{4(n+1-i)}} \cdot det(L)^{\frac{1}{n+1-i}}$$


$f(x_0, y_0) = 0 \pmod{e}$ with $|x_0| < X$ and $|y_0| < Y$
generate f_i s.t. $f_i(x_0, y_0) = 0 \pmod{N^m}$
$B = \begin{pmatrix} f_i(xX, yY) \\ \vdots \\ f_n(xX, yY) \end{pmatrix}$
LLL
$B' = \begin{pmatrix} b_1 = g_1(xX, yY) \\ b_2 = g_2(xX, yY) \\ \vdots \\ b_n \end{pmatrix}$
$g_1(x_0, y_0) = 0 \pmod{e^m}$ and $\|g_1(xX, yY)\| < \frac{e^m}{\sqrt{n}}$
$g_2(x_0, y_0) = 0 \pmod{e^m}$ and $\|g_2(xX, yY)\| < \frac{e^m}{\sqrt{n}}$
$g_1(x_0, y_0) = 0$ over $\mathbb{Z}$
$g_2(x_0, y_0) = 0$ over $\mathbb{Z}$
$r(x) = resultant_x(g_1, g_2)$

$$
\begin{array}{c|cccccccccc}
 & 1 & x & xy & x^2 & x^2y & x^2y^2 & y & xy^2 & x^2y^3 \\
e^2 & e^2 & & & & & & & & \\
xe^2 & & e^2X & & & & & & & \\
fe & e & eAX & eXY & & & & & & \\
x^2e^2 & & & & e^2X^2 & & & & & \\
xfe & & eX & & eAX^2 & eX^2Y & & & & \\
f^2 & 1 & 2AX & 2XY & A^2X^2 & 2AX^2Y & X^2Y^2 & & & \\
ye^2 & & & & & & & e^2Y & & \\
yfe & & & eAXY & & & & eY & eXY^2 & \\
yf^2 & & & 2AXY & & A^2X^2Y & 2AX^2Y^2 & Y & 2XY^2 & X^2Y^3
\end{array}
$$

Boneh-Durfee basis matrix for $m = 2$, $t = 1$

$$
\begin{array}{c}
 & \begin{array}{ccccccccc} 1 & x & xy & x^2 & x^2y & x^2y^2 & y & xy^2 & x^2y^3 \end{array} \\
\begin{array}{c} e^2 \\ xe^2 \\ fe \\ x^2e^2 \\ xfe \\ f^2 \\ ye^2 \\ yfe \\ yf^2 \end{array} &
\begin{pmatrix}
e^2 & & & & & & & & \\
 & e^2X & & & & & & & \\
e & eAX & eXY & & & & & & \\
 & & & e^2X^2 & & & & & \\
 & eX & & eAX^2 & eX^2Y & & & & \\
1 & 2AX & 2XY & A^2X^2 & 2AX^2Y & X^2Y^2 & & & \\
 & & & & & & e^2Y & & \\
 & & eAXY & & & & eY & eXY^2 & \\
 & & 2AXY & & A^2X^2Y & 2AX^2Y^2 & Y & 2XY^2 & X^2Y^3
\end{pmatrix}
\end{array}
$$

Boneh-Durfee basis matrix for $m = 2$, $t = 1$

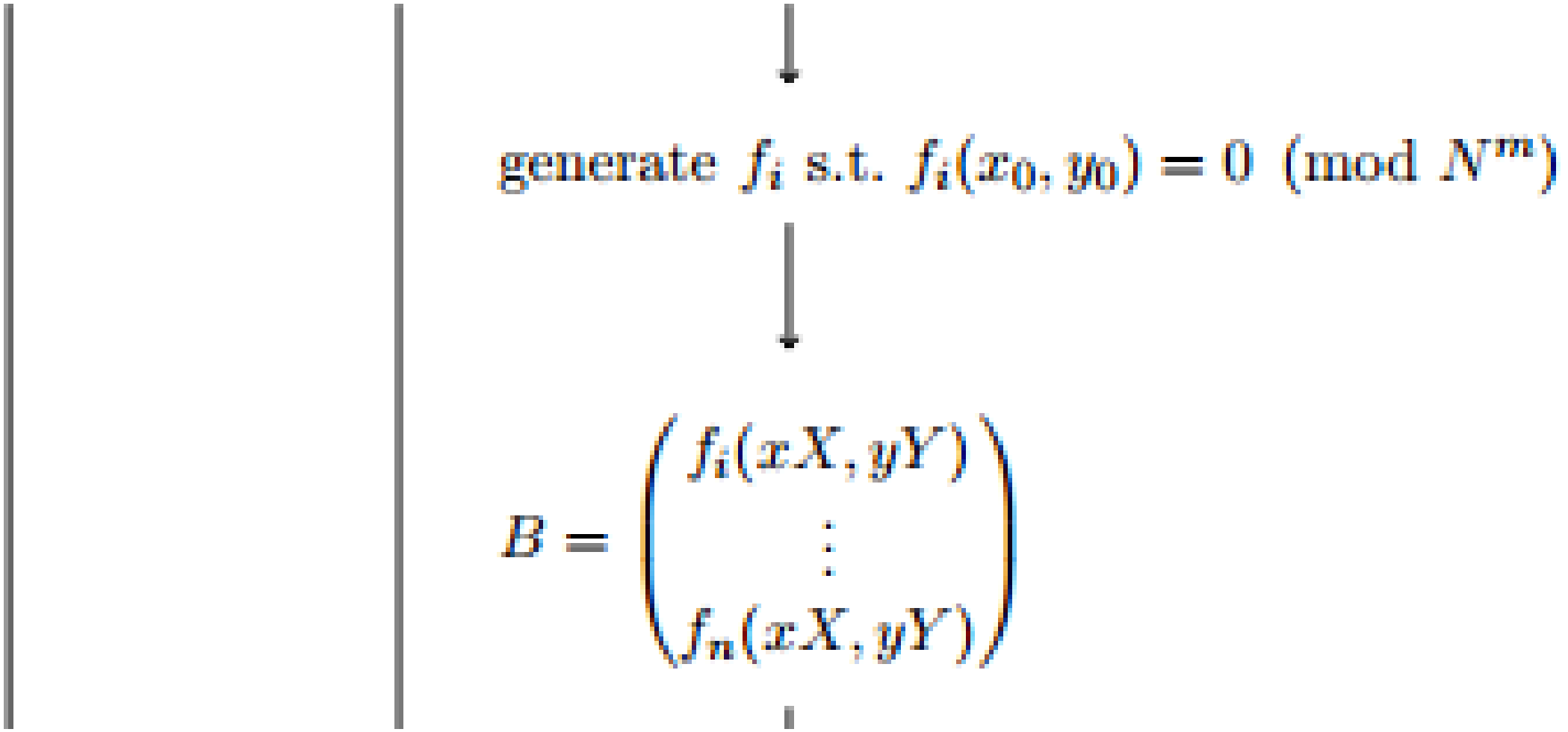
$$
\begin{array}{c}
 & \begin{array}{ccccccccc} 1 & x & xy & x^2 & x^2y & x^2y^2 & y & xy^2 & x^2y^3 \end{array} \\
\begin{array}{c} e^2 \\ xe^2 \\ fe \\ x^2e^2 \\ xfe \\ f^2 \\ yf^2 \end{array} &
\begin{pmatrix}
e^2 & & & & & & & & \\
 & e^2X & & & & & & & \\
e & eAX & eXY & & & & & & \\
 & & & e^2X^2 & & & & & \\
 & eX & & eAX^2 & eX^2Y & & & & \\
1 & 2AX & 2XY & A^2X^2 & 2AX^2Y & X^2Y^2 & & & \\
 & & 2AXY & & A^2X^2Y & 2AX^2Y^2 & Y & 2XY^2 & X^2Y^3
\end{pmatrix}
\end{array}
$$

After removing the damaging y-shifts' coefficient vectors

HERRMAN AND MAY:
UNRAVELLED LINEARIZATION

$$f(x,y) = \underbrace{1 + xy}_{u} + Ax \pmod{e}$$

$$f(x_0, y_0) = 0 \pmod{e} \text{ with } |x_0| < X \text{ and } |y_0| < Y$$



$$\text{generate } f_i \text{ s.t. } f_i(x_0, y_0) = 0 \pmod{N^m}$$

$$B = \begin{pmatrix} f_i(xX, yY) \\ \vdots \\ f_n(xX, yY) \end{pmatrix}$$

$$
\begin{array}{c}
 \\
e^2 \\
xe^2 \\
\bar{f}e \\
x^2e^2 \\
x\bar{f}e \\
\bar{f}^2 \\
y\bar{f}^2
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
1 & x & u & x^2 & ux & u^2 & u^2y
\end{array} \\
\begin{pmatrix}
e^2 & & & & & & \\
 & e^2X & & & & & \\
 & eAX & eU & & & & \\
 & & & e^2X^2 & & & \\
 & & & eAX^2 & eUX & & \\
 & & & A^2X^2 & 2AUX & U^2 & \\
 & -A^2X & -2AU & & A^2UX & 2AU^2 & U^2Y
\end{pmatrix}
\end{array}
$$

$$d < N^{0.292}$$

BONEH-DURFEE BOUND


CONCLUSIONS

coppersmith.sage

```sage
import time

debug = True

# display matrix picture with 0 and X
def matrix_overview(BB, bound):
    for ii in range(BB.dimensions()[0]):
        a = ('%02d ' % ii)
        for jj in range(BB.dimensions()[1]):
            a += '0' if BB[ii,jj] == 0 else 'X'
            a += ' '
        if BB[ii, ii] >= bound:
            a += '~'
        print a

def coppersmith_howgrave_univariate(pol, modulus, beta, mm, tt, XX):
    """
    Coppersmith revisited by Howgrave-Graham

    finds a solution if:
    * b|modulus, b >= modulus^beta , 0 < beta <= 1
    * |x| < XX
    """
    #
    # init
    #
    dd = pol.degree()
    nn = dd * mm + tt

    #
    # checks
    #
    if not 0 < beta <= 1:
        raise ValueError("beta should belongs in (0, 1]")

    if not pol.is_monic():
        raise ArithmeticError("Polynomial must be monic.")

    #
    # calculate bounds and display them
```

boneh_durfee.sage

```sage
import time

debug = True

# display stats on helpful vectors
def helpful_vectors(BB, modulus):
    nothelpful = 0
    for ii in range(BB.dimensions()[0]):
        if BB[ii,ii] >= modulus:
            nothelpful += 1

    print nothelpful, "/", BB.dimensions()[0], " vectors a
        not helpful"

# display matrix picture with 0 and X
def matrix_overview(BB, bound):
    for ii in range(BB.dimensions()[0]):
        a = ('%02d ' % ii)
        for jj in range(BB.dimensions()[1]):
            a += '0' if BB[ii,jj] == 0 else 'X'
            if BB.dimensions()[0] < 60:
                a += ' '
        if BB[ii, ii] >= bound:
            a += '~'
        print a

def boneh_durfee(pol, modulus, mm, tt, XX, YY):
    """
    Boneh and Durfee revisited by Herrmann and May

    finds a solution if:
    * d < N^delta
    * |x| < e^delta
    * |y| < e^0.5
    whenever delta < 1 - sqrt(2)/2 ~ 0.292
    """

    # substitution (Herrman and May)
    PR.<u, x, y> = PolynomialRing(ZZ)
    Q = PR.quotient(x*y + 1 - u) # u = x*y + 1
```

mimoo / RSA-and-LLL-attacks

Unwatch 7 | Unstar 71 | Fork 5

implementations of attacks on RSA through LLL reductions — Edit

151 commits | 1 branch | 0 releases | 1 contributor

branch: master | RSA-and-LLL-attacks / +

fixes

mimoo authored a day ago — latest commit b4df3bc27a

img	fixes	3 days ago
.gitignore	nom nom nom	a month ago
README.md	same fix as previous commit	6 days ago
boneh_durfee.pdf	for final survey	3 days ago
boneh_durfee.sage	no need	3 days ago
coppersmith.pdf	for final survey	3 days ago
coppersmith.sage	cleaning	6 days ago

Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings

HTTPS clone URL

https://github.com/r

You can clone with HTTPS, SSH

size of x_0	size of N	e	m	t	running time
100	512	3	3	0	0.02s
200	1024	3	3	0	0.05s

size of x_0	size of N	e	m	t	running time
100	512	3	3	0	0.02s
200	1024	3	3	0	0.05s

δ	size of N (bits)	size of d (bits)	m	t	dim(L)	running time
0.25	2048	512	3	1	11	0.5s
0.26	2048	532	3	1	11	1.9s
0.27	2048	553	6	2	33	2m 27s