Universiti Putra Malaysia

Coppersmith's Method: Solutions to Modular Polynomials

Boon Chian Tea
Assoc. Prof. Dr. Muhammad Rezal Kamel Ariffin, UPM
Prof. Dr. Bazzanella Danilo, PoliTO

June 22, 2021

Politecnico di Torino

Outlines

1. Motivations
2. Introduction
3. Important Theorems & Backgrounds
4. Coppersmith's Method
5. Applications of Coppersmith's method
6. Solutions to Multivariate Polynomials (Optional)
7. Main Reference

Disclaimer

About this sharing session

- All the content presented in this session are solely based on personal study and understanding.
- Certainly, I believe there are many theories and updates that I overlooked and not deeply familiar with.
- If there are any facts or statements presented later are wrong, please correct me, so that I can understand better too.

Table of Contents

Why Coppersmith's Method ?

- It is a popular method in cryptanalyzing RSA cryptosystem.
- One of the powerful methods to deal with the small integer solution(s) in both integer and modular polynomials.
- It involves lattices, and frequently applied in analyzing multivariate cryptography and lattice-based cryptography.
- The method is elegant, but a bit confusing for beginners who are not familiar with it.

Table of Contents

Modular Polynomials and Modular Equations

Let

$$F(x) = a_d x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + a_0 \tag{1}$$

be a univariate polynomial over $\mathbb{Z}[x]$ with degree $d > 1$. Suppose we are interested to find solutions to the modular equation of $F(x) \equiv 0 \pmod{N}$.

- If the factorization of N is known, then solving $F(x) \equiv 0 \pmod{N}$ is easy.
- Otherwise, it could be difficult.
- Moreover, if $F(x) \equiv 0 \pmod{N}$ has "small" solution, then we are not sure whether it is necessarily hard, or not?

Introduction (cont.)

- Håstad in 1988 firstly addressed similar problem of solving $F(x) \equiv 0 \pmod{N}$ with $a_d = 1$ (monic), $x < \min(N)$ and N composed by k distinct primes. Håstad proved that if $k > \frac{d(d+1)}{2}$, then x_0 can be recovered in polynomial time.
- Coppersmith in 1996 devised a method to find such "small" solution in polynomial time of $(\log N, d)$, with the condition such that x_0 is the solution to $F(x_0) \equiv 0 \pmod{N}$ and

$$|x_0| \leq N^{\frac{1}{d}} \tag{2}$$

The Central Problem

Suppose we know there exists at least one solution x_0 to $F(x) \equiv 0 \pmod{N}$ and that $|x_0| \le N^{\frac{1}{d}}$. How could we find them?

We know that $\left|x_0^i\right| \le N$ for all $0 \le i \le d$. If the coefficients a_i is small enough, one might have $F(x_0) = 0$ over $\mathbb{Z}$, then numerical methods (such as Newton's method) can be used to find an approximation of x_0 and checks whether $F(x_0) \equiv 0 \pmod{N}$.

Introduction (cont.)

What if those coefficients a_i are NOT small?

Coppersmith's Idea

Build a polynomial $G(x)$ from $F(x)$ that still has the same solution x_0, but with smaller coefficients a_i.

In other words, build from $F(x_0) = 0$ over $\mathbb{Z}_N$ to $G(x_0) = 0$ over $\mathbb{Z}$.

Example 1

Let $F(x) = x^2 + 33x + 215$. Find x_0 such that $F(x_0) \equiv 0 \pmod{323}$.

Solution 1

Set

$$\begin{aligned} G(x) &= 9F(x) - 323(x+6) \\ &= 9x^2 - 26x - 3 \\ &= (9x+1)(x-3) \end{aligned}$$

Then, $x_0 = 3$ is the solution to $G(x) = 0$, which is also the solution to $F(x) \equiv 0 \pmod{323}$.

Table of Contents

Theorem 1

(Howgrave-Graham) [4]. Let $F(x) = \sum_{i=0}^{d} a_i x^i \in \mathbb{Z}[x]$. Suppose $x_0 \in \mathbb{Z}$ is a solution to $F(x) \equiv 0 \pmod{N}$ such that $|x_0| < X$ for $N, X \in \mathbb{N}$. The following defines the row vector associated with the polynomial $F(x)$,

$$b_F = \left(a_o, a_1 X, \ldots, a_{d-1} X^{d-1}, a_d X^d\right).$$

If $\| b_F \| < \frac{N}{\sqrt{d+1}}$, then $F(x_0) = 0$.

Definition 1

Let $G_i(x) = Nx^i$ for $0 \leq i \leq d$ be $d+1$ polynomials that has the root $x_0 \pmod N$. Then we define a basis B corresponds to these polynomials $G_i(x)$ together with $F(x)$ for a lattice L as follows:

$$B = \begin{pmatrix} N & 0 & \dots & 0 & 0 \\ 0 & NX & \dots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & \dots & NX^{d-1} & 0 \\ a_0 & a_1 X & \dots & a_{d-1}X^{d-1} & X^d \end{pmatrix}$$

Important Theorems (cont.)

Theorem 2

Suppose given a basis B as defined in **Definition 1**, and $G(x)$ be the polynomial corresponding to the first vector in the **LLL**-reduced basis for L. If

$$X < \frac{N^{\frac{2}{d(d+1)}}}{\sqrt{2}\,(d+1)^{\frac{1}{d}}},$$

then any root x_0 of $F(x) \pmod{N}$ such that $|x_0| \le X$ satisfies $G(x_0) = 0$ in $\mathbb{Z}$.

Remark 1

Small solutions x_0 may be found even when x_0 does not satisfy the condition of the theorem above.

Table of Contents

The Full Coppersmith's Method

Based on **Theorem 2**, the success of of finding small roots of modular polynomials is essentially

$$2^{\frac{d}{4}} N^{\frac{d}{(d+1)}} X^{\frac{d}{2}} < \frac{N}{\sqrt{d+1}}. \tag{3}$$

There are two strategies to allow larger value for X in (3):

1. Increase the dimension n by adding rows to L that contributes less than N to the determinant, i.e., "x-shift" the polynomials $xF(x), x^2F(x), \ldots, x^kF(x)$.
2. Increase the power of N on the right hand side using power of $F(x)$. Since if $F(x_0) \equiv 0 \pmod{N}$, then $F(x_0)^k \equiv 0 \pmod{N^k}$.

At first it is not so obvious why the 2^{nd} strategy is valid. In fact, since $F(x_0) \equiv 0 \pmod{N}$, then one can express $F(x)$ as

$$F(x) = (x - x_0)\, p(x) + Nq(x)$$

for $p(x), q(x) \in \mathbb{Z}[x]$. Then,

$$\begin{aligned} F(x)^k &= \left[(x - x_0)\, p(x) + Nq(x)\right]^k \\ &= (x - x_0)^k p^k(x) + \binom{k}{1} (x - x_0)^{k-1} p^{k-1}(x)\, Nq(x) + \\ &\quad \cdots + \binom{k}{k-1} (x - x_0)\, p(x)\, N^{k-1} q^{k-1}(x) + N^k q^k(x) \end{aligned}$$

Since x_0 is the root to $F(x_0) \equiv 0 \pmod{N}$, we have

$$\begin{aligned} F(x_0)^k &= (x_0 - x_0)^k p^k(x) + \binom{k}{1}(x_0 - x_0)^{k-1} p^{k-1}(x) Nq(x) + \\ &\cdots + \binom{k}{k-1}(x_0 - x_0) p(x) N^{k-1} q^{k-1}(x) + N^k q^k(x) \\ &= N^k q^k(x) \\ &\equiv 0 \pmod{N^k} \end{aligned}$$

Hence, if $F(x_0) \equiv 0 \pmod{N}$, then $F(x_0)^k \equiv 0 \pmod{N^k}$. □

Theorem 3

(Coppersmith) [1]. Let $0 < \epsilon < \min\left\{0.18, \frac{1}{d}\right\}$. Let $F(x)$ be a monic polynomial of degree d with at least one small root $x_0 \pmod{N}$ such that

$$|x_0| < \frac{1}{2} M^{\frac{1}{d} - \epsilon}.$$

Then x_0 can be found in polynomial time in $\left(d, \frac{1}{\epsilon}, \log(N)\right)$.

*Note: the M in this proof is the modular N in these entire presentation.

Proof: Let $h > 1$ be an integer that depends on d and ϵ and will be determined in equation (19.3) below. Consider the lattice L corresponding (via the construction of the previous section) to the polynomials $G_{i,j}(x) = M^{h-1-j}F(x)^j x^i$ for $0 \leq i < d$, $0 \leq j < h$. Note that $G_{i,j}(x_0) \equiv 0 \pmod{M^{h-1}}$. The dimension of L is dh. One can represent L by a lower triangular basis matrix with diagonal entries $M^{h-1-j}X^{jd+i}$. Hence the determinant of L is

$$\det(L) = M^{(h-1)hd/2}X^{(dh-1)dh/2}.$$

Running LLL on this basis outputs an LLL-reduced basis with first vector $\underline{b}_1$ satisfying

$$\|\underline{b}_1\| < 2^{(dh-1)/4}\det(L)^{1/dh} = 2^{(dh-1)/4}M^{(h-1)/2}X^{(dh-1)/2}.$$

This vector corresponds to a polynomial $G(x)$ of degree $dh-1$ such that $G(x_0) \equiv 0 \pmod{M^{h-1}}$. If $\|\underline{b}_1\| < M^{h-1}/\sqrt{dh}$ then Howgrave-Graham's result applies and we have $G(x_0) = 0$ over $\mathbb{Z}$.

Hence, it is sufficient that

$$\sqrt{dh}2^{(dh-1)/4}M^{(h-1)/2}X^{(dh-1)/2} < M^{h-1}.$$

Rearranging gives

$$\sqrt{dh}2^{(dh-1)/4}X^{(dh-1)/2} < M^{(h-1)/2},$$

which is equivalent to

$$c(d,h)X < M^{(h-1)/(dh-1)}$$

where $c(d,h) = (\sqrt{dh}2^{(dh-1)/4})^{2/(dh-1)} = \sqrt{2}(dh)^{1/(dh-1)}$.

Now

$$\frac{h-1}{dh-1} = \frac{1}{d} - \frac{d-1}{d(dh-1)}.$$

Equating $(d-1)/(d(dh-1)) = \epsilon$ gives

$$\underline{h = ((d-1)/(d\epsilon)+1)/d} \approx 1/(d\epsilon). \tag{19.3}$$

Note that $dh = 1 + (d-1)/(d\epsilon)$ and so $c(d,h) = \sqrt{2}(1 + (d-1)/(d\epsilon))^{d\epsilon/(d-1)}$, which converges to $\sqrt{2}$ as $\epsilon \to 0$. Since $X < \frac{1}{2}M^{1/d-\epsilon}$ we require $\frac{1}{2} \leq \frac{1}{c(d,h)}$. Writing $x = d\epsilon/(d-1)$ this is equivalent to $(1+1/x)^x \leq \sqrt{2}$, which holds for $0 \leq x \leq 0.18$. Therefore, assume $\epsilon \leq (d-1)/d$.

Rounding h up to the next integer gives a lattice such that if

$$|x_0| < \frac{1}{2}M^{1/d-\epsilon}$$

then the LLL algorithm and polynomial root finding leads to x_0.

Since the dimension of the lattice is $dh \approx 1/\epsilon$ and the coefficients of the polynomials $G_{i,j}$ are bounded by M^h it follows that the running time of LLL depends on $d, 1/\epsilon$ and $\log(M)$. $\square$

Example 2

Let $N = 4611686047418417197$. Consider the polynomial

$$F(x) = 1942528644709637042 + 1234567890123456789x + 987654321987654321x^2 + x^3$$

Find a root $x_0 \pmod{N}$ such that $|x_0| \leq 2^{15}$.

Solution 2

From the proof of Theorem 3, $x = \frac{d\epsilon}{d-1}$ and $0 \leq x \leq 0.18$. Thus we have

$$\frac{d\epsilon}{d-1} \leq 0.18 \quad \text{which implies} \quad \frac{d-1}{d\epsilon} \geq \frac{1}{0.18}$$

and that

$$h = \frac{\frac{d-1}{d\epsilon}+1}{d} \geq \frac{\frac{1}{0.18}+1}{3} \approx 2.2$$

Therefore, we choose $h = 3$ in this case.

Since $G_{ij} = N^{h-1-j} X^i F^j(x)$, with $0 \leq i < d = 3$ and $0 \leq j < h = 3$. Then,

$$\begin{array}{lll} G_{00} = N^2 & G_{01} = NF(x) & G_{02} = F^2(x) \\ G_{10} = N^2 X & G_{11} = NXF(x) & G_{12} = XF^2(x) \\ G_{20} = N^2 X^2 & G_{21} = NX^2 F(x) & G_{22} = X^2 F^2(x) \end{array}$$

Arranging all the above G_{ij} accordingly, it forms the basis lattice B of dimension of 9 as follows:

Coppersmith's Method (cont.)

We denote $a_0 = 1942528644709637042$, $a_1 = 1234567890123456789$, $a_2 = 987654321987654321$ here, and take $X = 2^{15}$.

$$B = \begin{pmatrix} M^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M^2X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M^2X^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ Ma_0 & MXa_1 & MX^2a_2 & MX^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & MXa_0 & MX^2a_1 & MX^3a_2 & MX^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & MX^2a_0 & MX^3a_1 & MX^4a_2 & MX^5 & 0 & 0 & 0 \\ a_0^2 & 2a_0a_1X & \left(2a_0a_2+a_1^2\right)X^2 & 2\left(a_0+a_1a_2\right)X^3 & \left(2a_1+a_2^2\right)X^4 & 2a_2X^5 & X^6 & 0 & 0 \\ 0 & a_0^2X & 2a_0a_1X^2 & \left(2a_0a_2+a_1^2\right)X^3 & 2\left(a_0+a_1a_2\right)X^4 & \left(2a_1+a_2^2\right)X^5 & 2a_2X^6 & X^7 & 0 \\ 0 & 0 & a_0^2X^2 & 2a_0a_1X^3 & \left(2a_0a_2+a_1^2\right)X^4 & 2\left(a_0+a_1a_2\right)X^5 & \left(2a_1+a_2^2\right)X^6 & 2a_2X^7 & X^8 \end{pmatrix}$$

Executing the LLL-algorithm, Maple outputs the solution $x_0 = 16384$ to the $F(x) \equiv 0 \pmod{N}$ above.

Solution 3

Notice that if we eliminate the last two rows and columns from the previous solution (that is we excluded the last two constructed $G(x)$) such that

$$B = \begin{pmatrix} M^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M^2X & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M^2X^2 & 0 & 0 & 0 & 0 \\ Ma_0 & MXa_1 & MX^2a_2 & MX^3 & 0 & 0 & 0 \\ 0 & MXa_0 & MX^2a_1 & MX^3a_2 & MX^4 & 0 & 0 \\ 0 & 0 & MX^2a_0 & MX^3a_1 & MX^4a_2 & MX^5 & 0 \\ a_0^2 & 2a_0a_1X & \left(2a_0a_2+a_1^2\right)X^2 & 2(a_0+a_1a_2)X^3 & \left(2a_1+a_2^2\right)X^4 & 2a_2X^5 & X^6 \end{pmatrix}$$

Maple also outputs the same solution $x_0 = 16384$ to the $F(x) \equiv 0 \pmod{N}$ above, but with smaller dimension of 7.

Some Interesting Self-Discovery

- Sometimes $X \leq |x_0|$ (just not too small) is possible and still works in finding the root x_0.
- The solution will occur at the first row post-LLL-reduced matrix – in fact with suitably chosen X, the solution does appear in every row (most of the time) of the LLL-reduced matrix.
- It is helpful to consider the polynomial up to $F(x)^2$, which sometimes helps in reducing the dimension of the basis formed. Sometimes even the original $F(x)$ suffices to form the basis.

Table of Contents

Some Related Applications

- The small exponent attacks on RSA variants.
  1. Small public exponent e.
  2. Small private exponent d.
  3. Partial secret key exposure – with certain bits of d, p, q or N, it is possible to recover all completely.
- Factoring $N = pq$ with partial knowledge of p.
- Factoring moduli in the form of $p^r q$.
- Lattice-based cryptography and Learning with Errors (LWE).
- Solving the Hidden Number Problem (HNP) in finite fields and its applications to bit security of Diffie-Hellman key exchange.

So far...

- The Coppersmith's method discussed previously is of univariate (single variable x) case.
- The method is very straight forward and can be easily implemented to search for the small solution to modular equations.
- What about the case of finding roots of multivariate (integer/modular) polynomials?

Table of Contents

The Problem of Modular Bivariate Polynomial

Suppose given $F(x, y) \in \mathbb{Z}[x, y]$, find at least one root (x_0, y_0) to

$$F(x, y) \equiv 0 \pmod{N}$$

such that $|x_0| < X$ and $|y_0| < Y$.

Of course, one can apply the same strategy of Coppersmith, hoping to find two polynomials $F_1(x, y), F_2(x, y) \in \mathbb{Z}[x, y]$ such that

$$F_1(x_0, y_0) = F_2(x_0, y_0) = 0$$

over $\mathbb{Z}$, and that both $F_1(x, y), F_2(x, y)$ are algebraically independent (its resultant is not zero).

Multivariate Polynomials (cont.)

Theorem 4

Let $F(x, y) \in \mathbb{Z}[x, y]$ be a polynomial of total degree d, and $X, Y, N \in \mathbb{N}$ such that $XY < N^{\frac{1}{d}-\epsilon}$. Then one can find polynomials $F_1(x, y), F_2(x, y) \in \mathbb{Z}[x, y]$ such that for all $(x_0, y_0) \in \mathbb{Z}^2$ with $|x_0| < X, |y_0| < Y$ and $F(x_0, y_0) \equiv 0 \pmod{N}$, one has

$$F_1(x_0, y_0) = F_2(x_0, y_0) = 0$$

over $\mathbb{Z}$.

As the above theorem considers the case of modular form, readers may consider the proof given by Jutla [6] and Nguyen & Stern [7] for details.

Multivariate Polynomials (cont.)

- I considered the work done by Jochemz-May, as their heuristic strategy generally covers in finding both the modular and integer roots of multivariate polynomials by modifying the strategy accordingly.
- There are many strategies that had been proposed. For instance by Boneh & Durfee and Blömer & May.
- I personally found that Jochemz-May's strategy for finding roots of modular multivariate polynomials is easier to understand (from the beginner point of view).

Jochemz-May's Basic Strategy [5]

Let $\epsilon > 0$ be anarbitrary small constant. Depending on $\frac{1}{\epsilon}$, fixed an integer h. For $j \in \{0, \ldots, h+1\}$, define the set M_j of monomials

$$M_j := \left\{ x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \mid x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \text{ is a monomial of } F_N^h \right.$$

$$\left. \text{and } \frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^j} \text{ is a monomial of } F_N^{h-j} \right\}$$

where l is the leading monomial of F_N with coefficient a_l. It is assumed that the monomials of $F_N, \ldots, F_N^{h-1}$ are all contained in the monomials of F_N^h.

The Shift Polynomials

The following defines the shift polynomial that has the similar strategy as in Coppersmith's method, i.e., $G_{ij} = N^{h-1-j}x^i F^j(x)$.

$$G_{i_1 \ldots i_n}(x_1, \ldots, x_n) := \frac{x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}}{l^j} F_N^j(x_1, \ldots, x_n) N^{h-j} \tag{4}$$

for $j = 0, \ldots, h$ and $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \in M_j \backslash M_{j+1}$.

Multivariate Polynomials (cont.)

Example 3

Suppose we consider a small example of modular bivariate polynomial $F_N(x, y) = 1 + xy^2 + x^2y$. Let's assume $l = x^2y$ be the leading monomial and let $h = 2$.

Then, $F_N^2(x, y) = 1 + 2xy^2 + 2x^2y + x^2y^4 + 2x^3y^3 + x^4y^2$ with 6 monomials of $\left\{1, xy^2, x^2y, x^2y^4, x^3y^3, x^4y^2\right\}$. Now, we want to build a lattice having all the above monomials in its column.

Now, following the shift polynomial described by Jochemz-May:

$$G_{i_1 i_2}(x,y) := \frac{x^{i_1} y^{i_2}}{(x^2 y)^j} F_N^j(x,y)\, N^{h-j}$$

with $h = 2$ and $j = \{0,1,2\}$. We can now define the set of M_j as follows:

$$\begin{aligned} j = 0; \quad & M_0 = \left\{1, xy^2, x^2y, x^2y^4, x^3y^3, x^4y^2\right\} \\ j = 1; \quad & M_1 = \left\{x^2y, x^2y^4, x^3y^3, x^4y^2\right\} \\ j = 2; \quad & M_2 = \left\{x^4y^2\right\} \end{aligned}$$

Notice that the set M_j contains the monomials in $F_N^2(x,y)$ that is divisible by $\left(x^2y\right)^j$ for $j = 0, 1, 2$.

To construct the polynomials $G_{i_1 i_2}(x, y)$, we sort out the sets such that $x^{i_1} y^{i_2} \in M_j \backslash M_{j+1}$:

$$\begin{aligned} M_0 \backslash M_1 &= \left\{1, xy^2\right\} \\ M_1 \backslash M_2 &= \left\{x^2 y, x^2 y^4, x^3 y^3\right\} \\ M_2 \backslash M_3 &= \left\{x^4 y^2\right\} \end{aligned}$$

Taking the first sorted set $M_0 \backslash M_1 = \left\{1, xy^2\right\}$, we can now construct the following shift polynomials for each element (monomial) in the set:

$$\begin{aligned} G_{00}(x, y) &= \frac{x^0 y^0}{(x^2 y)^0} F_N^0(x, y) N^{2-0} &= \quad N^2 \\ G_{12}(x, y) &= \frac{x^1 y^2}{(x^2 y)^0} F_N^0(x, y) N^{2-0} &= \quad xy^2 N^2 \end{aligned}$$

For the next sorted set $M_1 \backslash M_2 = \left\{x^2 y, x^2 y^4, x^3 y^3\right\}$, we repeat the similar process:

$$
\begin{aligned}
G_{21}(x, y) = \frac{x^2 y^1}{(x^2 y)^1} F_N^1(x, y)\, N^{2-1} &= F_N(x, y)\, N \\
G_{24}(x, y) = \frac{x^2 y^4}{(x^2 y)^1} F_N^1(x, y)\, N^{2-1} &= y^3 F_N(x, y)\, N \\
G_{33}(x, y) = \frac{x^3 y^3}{(x^2 y)^1} F_N^1(x, y)\, N^{2-1} &= x y^2 F_N(x, y)\, N
\end{aligned}
$$

And for the last sorted set $M_2 \backslash M_3 = \left\{x^4 y^2\right\}$:

$$
G_{42}(x, y) = \frac{x^4 y^2}{(x^2 y)^2} F_N^2(x, y)\, N^{2-2} = F_N(x, y)^2
$$

- Notice that all the constructed shift polynomials except $G_{24}(x, y)$ contain the monomial from the original set.
- Since y^3 is not part of the monomials, introducing it in the basis matrix will produce more new monomials of y^3 and xy^5 which are not in the $F_N(x, y)$.
- This will next enlarge the dimension of the basis formed, which contradict to the aim of having low-determinant matrix.

The solution?

Instead of putting monomial x^2y^4 into $M_1 \backslash M_2$, we remain it in the first set of $M_0 \backslash M_1$, and proceed to compute as above:

$$G_{24}(x, y) = \frac{x^2y^4}{(x^2y)^0} F_N^0(x, y) N^{2-0} \quad = \quad x^2y^4N^2$$

Next, we can arrange and form the basis lattice B accordingly, as follows:

$$
B = \begin{array}{c} \\ G_{00} \\ G_{12} \\ G_{24} \\ G_{21} \\ G_{33} \\ G_{42} \end{array}
\begin{pmatrix}
1 & xy^2 & x^2y & x^2y^4 & x^3y^3 & x^4y^2 \\
N^2 & 0 & 0 & 0 & 0 & 0 \\
0 & XY^2N^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & X^2Y^4N^2 & 0 & 0 \\
N & XY^2N & X^2YN & 0 & 0 & 0 \\
0 & XY^2N & 0 & X^2Y^4N & X^3Y^3N & 0 \\
1 & 2XY^2 & 2X^2Y & 2X^2Y^4 & 2X^3Y^3 & X^4Y^2
\end{pmatrix}
$$

Since the diagonal contains 0, this can be handled easily by swapping G_{24} and G_{21}:

$$B = \begin{array}{c} \\ G_{00} \\ G_{12} \\ G_{21} \\ G_{24} \\ G_{33} \\ G_{42} \end{array} \begin{pmatrix} 1 & xy^2 & x^2y & x^2y^4 & x^3y^3 & x^4y^2 \\ N^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & XY^2N^2 & 0 & 0 & 0 & 0 \\ N & XY^2N & X^2YN & 0 & 0 & 0 \\ 0 & 0 & 0 & X^2Y^4N^2 & 0 & 0 \\ 0 & XY^2N & 0 & X^2Y^4N & X^3Y^3N & 0 \\ 1 & 2XY^2 & 2X^2Y & 2X^2Y^4 & 2X^3Y^3 & X^4Y^2 \end{pmatrix}$$

By executing the **LLL**-algorithm, one can proceed **to find the resultant matrix** that reveals the root of the $F_N(x, y) = 0$

Related Applications

- Cryptanalysis on RSA-CRT with known difference, i.e., the difference of $d_p - d_q$ is known to the attacker.
- Cryptanalysis on Common Prime RSA.

Table of Contents

Main Reference

- The main reference used in preparing this sharing session.

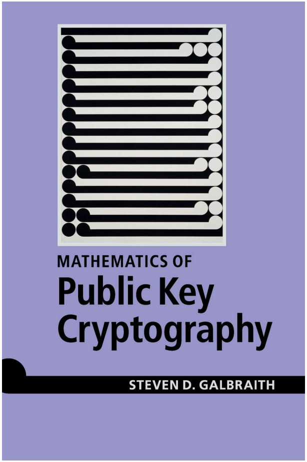

Figure 1: Mathematics of Public Key Cryptography by Steven D. Galbraith - **Chapter 19**.
E-book is available online for free.

Other References

Coppersmith, D.: Finding a Small Root of a Univariate Modular Equation. In: Maurer U. (eds) Advances in Cryptology - EUROCRYPT '96. EUROCRYPT 1996. Lecture Notes in Computer Science, **1070**. Springer, Berlin, Heidelberg, (1996).

Galbraith, S.D.: Mathematics of Public Key Cryptography (1st. ed.). Cambridge University Press, USA, (2012).

Håstad, J.: Solving simultaneous modular equations of low degree. SIAM J. Comput. **17**(2): 336–341. (1988).

Howgrave-Graham, N.: Finding Small Roots of Univariate Modular Equations Revisited. In: Darnell, M. (ed.) Cryptography and Coding 1997. LNCS, **1355**: 131–142. Springer, Heidelberg (1997).

Jochemsz, E., May, A.: A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants. In: Lai X., Chen K. (eds) Advances in Cryptology – ASIACRYPT 2006, LNCS, 4284. Springer, Berlin, Heidelberg, (2006).

Jutla, C.S.: On Finding Small Solutions of Modular Multivariate Polynomial Equations, EUROCRYPT 1998 (K. Nyberg, ed.), LNCS, **1403**: 158–170. Springer, (1998).

Nguyen, P.Q. and Stern, J.: The Two Faces of Lattices in Cryptology, Cryptography and Lattices (CaLC) (J. H. Silverman, ed.), LNCS, **2146**: 146–180. Springer, (2001).