NTNU
Norwegian University of
Science and Technology

Finding Small Roots of Polynomial Equations Using Lattice Basis Reduction

Ingeborg Sletta

Master of Science in Mathematics
Submission date: May 2009
Supervisor: Kristian Gjøsteen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

With this thesis I end my 6 year stay here at NTNU. It have been some exciting and challenging years where I have learned a lot. Socially I had some very nice years thanks to the class of LUR 2003.

Writing this thesis have been a challenging task that have required a lot of work alone. I have therefore appreciated my fellow students a lot for very meaningful breaks in the hall. I would specially like to thank Johan, Yvonne and Benedikte for LaTeX help and motivation.

I would like to thank my supervisor Kristian Gjøsteen for introducing me to this exciting field of mathematics, and for great guidance and help with this project.

I am grateful for my good friends who have been a great motivation and comfort during these years. I would like to thank my parents, grandparents, my sister and brother for being supportive and for giving me perspective and help. And I would like to thank Jørgen for great support and help during these last two years.

Trondheim, May 29, 2009

Ingeborg Sletta

Abstract

In 1996 Don Coppersmith used the Lenstra, Lenstra and Lovász (LLL) algorithm to find small integer roots of polynomials. It was used on univariate modular equations and bivariate polynomial equations, and he proved that the method would find small roots, if they existed, in polynomial time.

In this thesis we look at the LLL-algorithm and how this can be used to solve univariate modular equations and bivariate polynomial equations. We talk briefly about cryptography and public-key cryptosystems, and we will present some theory about lattices. We will look in detail at Coppersmith's method and show how it can be used to attack RSA encryption with a low exponent.

We will also look at improvements of this method done later by Howgrave-Graham which simplifies the univariate case. In the end we give a small univariate example where we apply both Coppersmith's and Howgrave-Graham's method.

Contents

Chapter 1

Introduction

The aim of this thesis is to understand the method described by Coppersmith on how to find small roots of polynomial equations using lattice basis reduction. We will use Coppersmith's article [2] and go in detail in the univariate and the bivariate case. Howgrave-Graham [5] has improved the univariate case and we will look into his method as well.

In 1982 Lenstra, Lenstra and Lovász developed the LLL-algorithm [6], which is an approximation to the shortest vector problem. This algorithm takes a lattice basis as input and converts it into a basis with short vectors that are nearly orthogonal and sorted by length. This has become an important algorithm with many applications. We will look at the algorithm and the properties of the basis returned by the algorithm.

In 1996 Coppersmith described how to use the LLL-algorithm on lattice bases to be able to find small roots of polynomials. Coppersmith makes a matrix that will span a lattice, and this matrix consists among other things of the coefficients to our polynomial. After the matrix get the appropriate form we use the LLL-algorithm on it to get a LLL-reduced basis. We use this basis to create a new polynomial, that we can solve and that have the same small roots as the original polynomial. This can be used to attack RSA cryptosystems with a low exponent.

Howgrave-Graham improved Coppersmiths univariate case by making a more direct approach. The matrix in this method is on a better form in the first place, so it demands less computations. We use the LLL-algorithm in this case too, but we look at the first vector in the LLL-reduced basis instead of the last one as in Coppersmiths case. Howgrave-Graham proved that his and Coppersmith's algorithm are equivalent, but Howgrave-Graham's may be preferred for computational efficiency.

We start in Chapter 2 with some general theory about lattices and then we go through the LLL-algorithm and its properties. In Chapter 3 we look at the RSA cryptosystem with the aim to motivate the reader on how Coppersmith's method can be used to attack such systems. In Chapter 4 we go through the method of finding small roots of polynomial equations as done by Coppersmith, and then Howgrave-Graham. In the last chapter we show an example where we apply both Coppersmith's and Howgrave-Graham's method.

Chapter 2

Theory of lattices

2.1 Lattices

A lattice in space is a set of points with a periodic structure. One example of a lattice is to take a 2-dimensional plane and cover it with squares, rectangels or parallelograms. We then place a point in every vertex, and all these points give us a lattice. We can reach every point in the lattice with linear combinations of the green vectors in Figure 2.1, so we call these vectors a basis for the lattice. Another basis is the two purple arrows, but presumably the green vectors are a better basis than the purple ones. A lattice will have several bases, and our aim in this chapter is to find a convenient basis for our lattice.

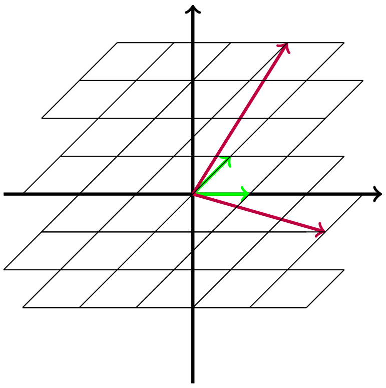

Figure 2.1: A 2-dimensional space covered with parallelograms, every vertex have a point and all these points give a lattice.

Definition 2.1. *Let n and m be positive integers. A subset $\mathcal{L}$ of the m-dimensional real vector space $\mathbb{R}^m$ is called a* lattice *if there exists a finite set of vectors $\{\mathbf{b}_i\} \subseteq \mathbb{R}^m$ such that*

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z}, 1 \leq i \leq n \right\}.$$

We say that such a set $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ *is a* basis *if it* spans $\mathcal{L}$ *and is* $\mathbb{Z}$*-linearly independent. We call* n *the* rank *of* $\mathcal{L}$*, and* m *the* dimension*.*

The lattice is a full-rank lattice if $n = m$. Given a $n \times m$ matrix B we can define

$$\mathcal{L} = \mathcal{L}(B) = \{xB | x \in \mathbb{Z}^n\}.$$

If we have a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for $\mathcal{L}$ then we have a $n \times m$ matrix for $\mathcal{L}(B) = \mathcal{L}$,

$$B = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix},$$

where the $\mathbf{b}_i$'s are written as row vectors. The rank of the lattice is n which also is the rank of the matrix B. The B matrix consists of the lattice basis as rows, and since the basis vectors are linearly independent the row rank is n and so is the rank of the matrix.

Theorem 2.1. *Let* $\mathcal{L}$ *be a finitely generated lattice over* $\mathbb{Z}$*. Then all bases of* $\mathcal{L}$ *have the same number of elements.*

Proof. Let $n < m$ and let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ and $\mathbf{b}'_1, \ldots, \mathbf{b}'_m$ be two bases for $\mathcal{L}$. We write

$$\mathbf{b}_i = a_{1i}\mathbf{b}'_1 + \cdots + a_{mi}\mathbf{b}'_m, \quad 1 \le i \le n$$

$$\mathbf{b}'_i = c_{1j}\mathbf{b}_1 + \cdots + c_{nj}\mathbf{b}_n, \quad 1 \le j \le m$$

Then

$$\begin{aligned}\mathbf{b}_i &= \sum_{k=1}^{n}\sum_{j=1}^{m} c_{kj}a_{ji}\mathbf{b}_k \\ &= \sum_{k=1}^{n}\mathbf{b}_k\sum_{j=1}^{m} c_{kj}a_{ji},\ 1 \le i \le n\end{aligned}$$

thus because the $\mathbf{b}_i$'s are linearly independent

$$\sum_{k=1}^{n}\mathbf{b}_k\left(\sum_{j=1}^{m} c_{kj}a_{ji} - \lambda_{ki}\right) = 0$$

where

$$\lambda_{ki} = \begin{cases} 0 & i \ne k \\ 1 & i = k \end{cases}.$$

Let $A = (a_{ji})$ be a $m \times n$ matrix and $C = (c_{kj})$ be a $n \times m$ matrix. This yields

$$CA = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & & \vdots \\ c_{n1} & \cdots & c_{nm} \end{pmatrix}\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = I_n$$

where I_n is a $n \times n$ identity matrix. Similarly $AC = I_m$. Let $A' = \begin{pmatrix} A & 0 \end{pmatrix}$ and $C' = \begin{pmatrix} C \\ 0 \end{pmatrix}$ be $n \times n$ augmented matrices, where each of the 0 blocks is a matrix of appropriate size. Then

$$A'C' = I_n, \qquad C'A' = \begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix}.$$

This implies $\det(A'C') = 1$ and $\det(C'A') = 0$. But A' and C' are $m \times m$ matrices over the commutative ring $\mathbb{Z}$, so $\det(A'C') = \det(C'A')$ which yields a contradiction. Hence $n \geq m$. By symmetry we get $m \geq n$. This proves that $m = n$. □

Proposition 2.1. *Let* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ *and* $\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_n$ *be two bases for the same lattice. Then*

$$\det\left((\mathbf{b}_1\mathbf{b}_2 \ldots \mathbf{b}_n)^T\right) = \pm \det\left((\mathbf{b}'_1\mathbf{b}'_2 \ldots \mathbf{b}'_n)^T\right)$$

Proof. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ and $\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_n$ be two bases for the same lattice. From the proof of Theorem 2.1 we have that

$$\mathbf{b}_i = \sum_{j=1}^{n} a_{ij}\mathbf{b}'_j$$

and

$$\mathbf{b}'_j = \sum_{i=1}^{n} c_{ji}\mathbf{b}_i$$

where $\{a_{ij}, c_{ji}\} \in \mathbb{Z}$. We write this as matrices:

$$\begin{aligned} \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{b}'_1 \\ \mathbf{b}'_2 \\ \vdots \\ \mathbf{b}'_n \end{pmatrix} \\ &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}. \end{aligned}$$

Both (a_{ij}) and (c_{ji}) are invertible matrices with integer elements, so their determinants are invertible integers as well. The only invertible integers are ±1, so $\det((c_{ji}))$ and $\det((a_{ij}))$ are ± 1. We have the desired equality

$$\begin{aligned} \det\left((\mathbf{b}_1\mathbf{b}_2 \ldots \mathbf{b}_n)^T\right) &= \det\left((a_{ij})\right) \det\left((\mathbf{b}'_1\mathbf{b}'_2 \ldots \mathbf{b}'_n)^T\right) \\ &= \pm \det\left((\mathbf{b}'_1\mathbf{b}'_2 \ldots \mathbf{b}'_n)^T\right). \end{aligned}$$

□

Let $|\cdot|$ denote the absolute value, $(,)$ define the ordinary inner product on $\mathbb{R}^n$ and let $\|\cdot\|$ define the ordinary Euclidean norm. We let the dot product of two vectors $\mathbf{x}$ and $\mathbf{y}$ be denoted by $\mathbf{x} \cdot \mathbf{y}$.

Definition 2.2. *The* determinant $\det(\mathcal{L})$ *of a full rank lattice* $\mathcal{L}(B)$ *with basis* $\mathbf{b}_1, \ldots, \mathbf{b}_n$ *is*

$$\det(\mathcal{L}) = |\det B|\,, \quad \textit{where} \quad B = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix},$$

and the $\mathbf{b}_i$*'s are written as row vectors.*

We can now conclude that the determinant of the lattice is a positive real number that does not depend on the choice of the basis. It is positive since we take the absolute value.

2.2 Shortest Vector Problem and Closest Vector Problem

There are two central problems when studying a lattice, that is how to find the shortest vector in the lattice and how to find the closest vector to a point not in the lattice. These problems are called the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) and are two fundamental hard lattice problems. By a short vector we mean a vector with small norm, so if a vector $\mathbf{x}$ is shorter than a vector $\mathbf{y}$, then $\|\mathbf{x}\| < \|\mathbf{y}\|$. The SVP and CVP are defined as follows, where $\mathbf{x} \in \mathcal{L}$ and $\mathbf{x}_0 \notin \mathcal{L}$:

$$SVP(\mathcal{L}) = \{\mathbf{x} \mid \mathbf{x} \neq 0, \forall \mathbf{y} \in \mathcal{L} : \|\mathbf{x}\| \leq \|\mathbf{y}\|\}$$

$$CVP(\mathcal{L}, \mathbf{x}_0) = \{\mathbf{x} \mid \forall \mathbf{y} \in \mathcal{L} : \|\mathbf{x} - \mathbf{x}_0\| \leq \|\mathbf{y} - \mathbf{x}_0\|\}.$$

To solve SVP and CVP we want a basis whose basis elements are short and almost orthogonal. The next section give an algorithm that can find such a basis.

2.3 The LLL-algorithm

The Lenstra, Lenstra and Lovász algorithm, usually named the LLL-algorithm, is an approximation to the shortest vector problem. This algorithm was developed by A. K. Lenstra, H.W. Lenstra and L. Lovász in 1982 [6].

The algorithm consists of a reduction step and a swap step. In the reduction step the basis is reduced by the basis vectors with lower basis index. After all the vectors have been reduced, they get swapped. We would like to have the shortest element first and the longest at the end, but we do not always need strict ordering. A δ decides how strict the ordering is going to be, and δ is in the interval $(\frac{1}{4}, 1)$, where 1 would have given the strict order.

The Gram-Schmidt orthogonalization process makes an inner product space basis orthogonal.

Definition 2.3. *Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$, the* Gram-Schmidt orthogonalization *of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is defined by*

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \tilde{\mathbf{b}}_j \text{ where } \mu_{i,j} = \frac{(\mathbf{b}_i, \tilde{\mathbf{b}}_j)}{(\tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j)}.$$

We denote the Gram-Schmidt basis of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ by $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$.

The Gram-Schmidt of a basis element $\mathbf{b}_i$ has shorter or equal length compared to $\mathbf{b}_i$, namely that $\|\tilde{\mathbf{b}}_i\| \leq \|\mathbf{b}_i\|$.

Definition 2.4. *A basis $B = \{\mathbf{b}_1, ..., \mathbf{b}_n\} \in \mathbb{R}^n$ is a δ-*LLL-Reduced Basis *if the following holds:*

1. *For all $1 \leq j < i \leq n$ we have $|\mu_{i,j}| \leq \frac{1}{2}$.*
2. *For all $1 \leq i < n$ we have $\delta \left\|\tilde{\mathbf{b}}_i\right\|^2 \leq \left\|\mu_{i+1,i}\tilde{\mathbf{b}}_i + \tilde{\mathbf{b}}_{i+1}\right\|^2$.*

The LLL-algorithm transforms an arbitrary basis into a δ-LLL-reduced one. The input to the algorithm is a lattice basis $\{\mathbf{b}_1, ..., \mathbf{b}_n\} \in \mathbb{Z}^n$. The output is a δ-LLL-reduced basis for the lattice. The algorithm is given in Figure 2.2, where $\lceil \cdot \rfloor$ means nearest integer.

```
Start: compute $\tilde{\mathbf{b}}_1, ..., \tilde{\mathbf{b}}_n$
Reduction step:
    for $i = 2$ to $n$ do
        for $j = i - 1$ to 1 do
            $\mathbf{b}_i \leftarrow \mathbf{b}_i - c_{i,j}\mathbf{b}_j$ where $c_{i,j} = \left\lceil \frac{(\mathbf{b}_i, \tilde{\mathbf{b}}_j)}{(\tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j)} \right\rfloor$
Swap step:
        if $\exists i$ s.t. $\delta \left\|\tilde{\mathbf{b}}_i\right\|^2 > \left\|\mu_{i+1,i}\tilde{\mathbf{b}}_i + \tilde{\mathbf{b}}_{i+1}\right\|^2$ then
        $\mathbf{b}_i \leftrightarrow \mathbf{b}_{i+1}$ for the smallest $i$
        go to start
Output $\mathbf{b}_1, ..., \mathbf{b}_n$
```

Figure 2.2: LLL-algorithm

We want to show that the algorithm gives us a reduced basis every time. We define the *potential* of the lattice basis, a function mapping a lattice basis to a positive number.

Definition 2.5. *Let $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a lattice basis. The* potential *of B, denoted by D_B is defined as*

$$D_B = \prod_{i=1}^{n} \left\|\tilde{\mathbf{b}}_i\right\|^{n-i+1} = \prod_{i=1}^{n} \left\|\tilde{\mathbf{b}}_1\right\| \left\|\tilde{\mathbf{b}}_2\right\| \cdots \left\|\tilde{\mathbf{b}}_i\right\| = \prod_{i=1}^{n} D_{B,i},$$

where $D_{B,i} = |\det(\mathcal{L}_i)|$ and $\mathcal{L}_i$ is defined as the lattice spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_i$.

If we can show that the initial value of D_B is not too large, and that it decays quickly, then the algorithm terminates and gives us a basis with shorter or equal basis elements than the initial basis. We can find the maximal value of D_B replacing all the basis elements $\mathbf{b}_i$ with the largest value among them. Thus D_B is bounded from above by $(\max_i \|\mathbf{b}_i\|)^{n(n-1)/2}$.

During the reduction step in the algorithm the Gram-Schmidt basis does not change and hence the D_B does not change since it only relies on the Gram-Schmidt basis. We now look at the swap step, and assume that $\mathbf{b}_k$ is swapped by $\mathbf{b}_{k+1}$. The Gram-Schmidt basis does not change up to k, it will change when $i = k$ and $i = k+1$. When $i > k+1$ the Gram-Schmidt basis remains the same. This means that the $D_{B,i}$ only changes when $i = k$, or $i = k+1$. Let $D'_{B,i}$ denote the new value of $D_{B,i}$ after the swap, and let $\mathcal{L}'_i$ be the new lattice. We look at the ratio between the two potentials

$$\begin{aligned}
\frac{D'_{B,i}}{D_{B,i}} &= \frac{\det(\mathcal{L}'_i)}{\det(\mathcal{L}_i)} \\
&= \frac{\det(\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}))}{\det(\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_i))} \\
&= \frac{\left(\prod_{j=1}^{i-1} \left\|\tilde{\mathbf{b}}_j\right\|\right) \left\|\mu_{i+1,i}\tilde{\mathbf{b}}_{i+1}\right\|}{\prod_{j=1}^{i} \tilde{\mathbf{b}}_j} \\
&= \frac{\left\|\mu_{i+1,i}\tilde{\mathbf{b}}_{i+1}\right\|}{\left\|\tilde{\mathbf{b}}_i\right\|} \\
&< \sqrt{\delta}
\end{aligned}$$

where the last inequality comes from the swap step. In each iteration the potential D_B decreases by a multiplicative factor $\sqrt{\delta}$. It is possible to show that D_B has a lower bound. The potential is then bounded from below, from above and it decreases for each iteration in the algorithm. The algorithm terminates every time, hence the number of iterations are finite.

2.4 Properties of the LLL-algorithm

A lattice $\mathcal{L}$ can have many different bases, but we want to find the basis with shortest elements. The problem is that there are no known efficient algorithms that can find such a basis in reasonable time, except for in dimension 2 and 3. The LLL-algorithm takes in an arbitrary basis for the lattice and makes it into a lattice basis with short vectors that are nearly orthogonal and sorted by length. This is a δ-LLL-reduced basis for the lattice and such bases have certain properties as stated in [6].

Proposition 2.2. *Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ be a δ-LLL-reduced basis for a lattice $\mathcal{L}$ in $\mathbb{R}^n$, and let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be the Gram-Schmidt basis. Then we have*

$$\|\mathbf{b}_j\|^2 \leq 2^{i-1} \|\tilde{\mathbf{b}_i}\|^2 \; for \; 1 \leq j \leq i \leq n$$

$$\|\tilde{\mathbf{b}_1}\| \leq \det(\mathcal{L})^{1/n} 2^{(n-1)/4}.$$

Observe that $\mathbf{b}_1$ is a short element in the lattice. The first equation in Proposition 2.2 give $\|\mathbf{b}_1\|^2 \leq 2^{n-1} \|\tilde{\mathbf{b}_n}\|^2$. From this and the second equation of Proposition 2.2 it follows that the last basis element $\mathbf{b}_n$ satisfies

$$\left\|\tilde{\mathbf{b}_n}\right\| \geq \det(\mathcal{L})^{1/n} 2^{(n-1)/4}. \tag{2.1}$$

We do not necessarily find the shortest vector in the lattice, but we find a short one, so it is an approximation to the shortest vector problem. To solve SVP and CVP for any dimension is still hard.

Definition 2.6. *A* hyperplane *in a n-dimensional space is the set of all* $\mathbf{y} = (y_0, \dots, y_n)$ *in the space which satisfies the equation*

$$a_0 y_0 + \cdots + a_n y_n = 0,$$

where not all a_i *are zero.*

A hyperplane in a vector space is a vector subspace whose dimension is one less than the dimension of the whole vector space. We now look at some lemmas from Coppersmith [2], where $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}$ is a δ-LLL-reduced basis.

Lemma 2.1. *If a lattice element* $\mathbf{s}$ *satisfies* $\|\mathbf{s}\| < \det(\mathcal{L})^{1/n} 2^{(n-1)/4}$ *then* $\mathbf{s}$ *lies in a hyperplane spanned by* $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}$.

Proof. A lattice element $\mathbf{s}$ can be expressed as $\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{b_i}$, where a_i are integers. The norm of $\mathbf{s}$ is,

$$\|\mathbf{s}\| = \|a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_n\mathbf{b}_n\| \geq \|a_n\mathbf{b}_n\| = |a_n| \|\mathbf{b}_n\| \geq |a_n| \|\tilde{\mathbf{b}_n}\|.$$

From (2.1) we have $\|\tilde{\mathbf{b}_n}\| \geq \det(\mathcal{L})^{1/n} 2^{(n-1)/4}$ and we have that $\|\mathbf{s}\| < \det(\mathcal{L})^{1/n} 2^{(n-1)/4}$ from Proposition 2.2. This gives us that $\mathbf{s}$ is shorter than the shortest orthogonal basis element, then $\|\mathbf{s}\| < \|\tilde{\mathbf{b}_n}\|$. Thus we get that a_n must be zero and $\mathbf{s}$ must have dimension $n-1$ and therefore $\mathbf{s}$ lies in the hyperplane spanned by $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}$. □

In the further discussion we are not necessarily looking for the shortest nonzero vector in the lattice but for a relatively short one, and Lemma 2.1 confines all such short vectors to a hyperplane. In Lemma 2.2 we generalize this from a hyperplane to a subspace of smaller dimension.

Lemma 2.2. *If a lattice element* $\mathbf{s}$ *satisfies* $\|\mathbf{s}\| < \|\tilde{\mathbf{b}_i}\|$ *for all i= k+1,...,n, then* $\mathbf{s}$ *lies in a hyperplane spanned by* $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$.

Chapter 3

Cryptography

Cryptography is about communication when there are possible enemies around. The goal with cryptography is to enable two persons, which we call Alice and Bob, to communicate over an insecure channel in such a way that an opponent, Oscar, cannot understand what is being said. We call the information that Alice wants to send Bob a *plaintext*. In advance Alice and Bob has secretly chosen a key K. This key then gives rise to an encryption rule e_K and a decryption rule d_K. Alice encrypts the plaintext before she sends it over the insecure channel, and when Bob receives it he decrypts the message and gets the plaintext and the information Alice wanted to give him.

We call cryptosystems where e_K and d_K are the same, or where d_K can easily be derived from e_K, *symmetric-key cryptosystems*. In *public-key cryptosystems* on the other hand, the encryption rule e_K is a public key, and it is computationally infeasible to determine d_K given e_K. The public-key cryptography was first put forward in the open literature by Diffie and Hellman in 1976. The theory was in reality developed years earlier. The security in public-key cryptography rests in different computational problems, but it can never provide unconditional security.

3.1 Knapsack cryptosystem

The knapsack encryption is a public key encryption scheme using a trap door function. The required trap door is obtained from the ancient knapsack puzzle: given the total weight of a knapsack and the weight of individual objects, which objects are in the bag? This problem can be quite difficult to solve.

Public key systems based on the knapsack problem are generally suspected to be weaker than RSA, and some versions have already been broken. The Merkle-Hellman knapsack encryption is based on the knapsack problem and was the first concrete realization of a public-key encryption scheme. It is therefore important for historical reasons. Many variations have been proposed, but most of them have been demonstrated to be insecure, including the original.

The ancient knapsack problem is a subset sum problem, which is given by the following: given a set of positive integers $\{a_1, a_2, \ldots, a_n\}$ and a positive integer s, determine whether or not there is a subset of the a_i that sum to s. It can also be asked to actually find a subset of the a_i which sum to s, provided that such a subset exists.

The subset sum problem can be reduced to the problem of finding a short vector in the lattice. The LLL-algorithm can then be used to find such a short vector. This is the most powerful general attack known on the knapsack encryption schemes.

3.2 Finding roots

We want to look at finding roots of different types of polynomials. Finding roots of a single variable polynomial with integer coefficients is easy. But finding roots to a polynomial in a single variable modulo a natural number which is not a prime

$$p(x) \equiv 0 \pmod{n},$$

is hard for large n. Finding such roots is as difficult as factoring a number n, which is believed to be very hard for high n. If n is a prime number then it is not difficult to find the roots of the modular polynomial.

Another hard problem is to find integer roots to a polynomial in several variables

$$p(x, y) = 0.$$

As we will see later Coppersmith is able to solve such problems, if we restrict these cases to the problems where there exist small enough solutions to the polynomials. These special cases can be solved using a lattice basis reduction technique as we will see in the following chapter.

3.3 RSA cryptosystem

The RSA cryptosystem was published by Rivest, Shamir and Adleman in 1977. For the first time in open literature. RSA is a public-key cryptosystem and its security is based on the difficulty of factoring large integers. In the RSA cryptosystem we use computations in $\mathbb{Z}^n$, where n is the product of two distinct primes p and q,

$$n = pq.$$

The Euler ϕ-function for n is $\phi(n) = (p-1)(q-1)$. The encryption and decryption are inverse operations since

$$ab \equiv 1 \pmod{\phi(n)}.$$

If we denote the plaintext x we get the following encryption and decryption formula

$$e_K(x) = x^b \pmod{n} = y$$

$$d_K(y) = y^a \pmod{n} = x.$$

If RSA is to be secure, it is necessary that $n = pq$ is large enough so that factoring it will be computationally infeasible.

Example 3.1. Example of stereotyped message in RSA

In this example most of the message is fixed or "stereotyped", we call this part of the message F. An example of this F can be: "Todays password is...". Let the plaintext m consist of two pieces, x and F. The first piece F is known and is the fixed part of the message. The second unknown piece x is the secret password, and the length of x is less than one-third the length of N. We use RSA encryption with an exponent of 3, so the ciphertext c is given by

$$c = m^3 = (F + x)^3 \pmod{N}.$$

We try to write this as a polynomial with x as the unknown, and we assume that we know F, c and N. We then get

$$p(x) = (F + x)^3 - c \pmod{N}.$$

We want to find x_0 so that

$$p(x_0) = (F + x)^3 - c = 0 \pmod{N}.$$

If we can solve this polynomial for x_0, then we will have recovered the secret message x.

It is obvious that we can recover x if $F = 0$. We will show how to recover x for $F \neq 0$ in the following chapter.

Chapter 4

Finding small roots of polynomial equations using lattice basis reduction

In [1] and [2] Coppersmith describes how lattice reduction can be used to find small roots of polynomial equations. It can be univariate modular equations, $p(x) \equiv 0 \pmod{N}$, or bivariate polynomials, $p(x, y) = 0$. In both these cases the outline is the same, but the bivariate case becomes more technical. Coppersmith makes a matrix that consists of the coefficients to the polynomial we want to find the roots of, the modulus N in the univariate case and some other factors. The rows of this matrix spans a lattice and we find a sublattice of this lattice. We use a lattice basis reduction technique on vectors in the sublattice, and find a hyperplane. This hyperplane contains all the short lattice elements. The equation of this hyperplane translates to a linear relation and then to a polynomial equation $c(x_0) = 0$ or to $c(x_0, y_0) = 0$ over $\mathbb{Z}$, where x_0 or (x_0, y_0) are a small root. In the univariate case we solve directly for x_0. In the bivariate case we combine $p(x_0, y_0)$ and $c(x_0, y_0)$ and solve. Coppersmith proved that at least one root can be found in polynomial time if we have a bound on the root we want to find.

Howgrave-Graham simplified Coppersmith's algorithm for the univariate case by making a more direct approach. In the entries of the matrix consisting of the coefficients of $p(x)$ in Coppersmith's case, Howgrave-Graham now uses entries that are multiples of $p(x)$ and N instead.

A simplification of the bivariate case was later proposed by Coron [3], but asymptotically its complexity was worse than Coppersmith's method. But in [4] a new algorithm for the bivariate integer case was put forward by Coron, this time with the same complexity as in Coppersmith's method. We will not go into Coron's algorithm here, but look at the other three methods.

We restrict the problem of finding roots of the two polynomial equations described for the cases where there exist a solution small enough. We can solve these special cases using a lattice basis reduction technique.

The idea is that we get a polynomial equation for example from RSA encryption, see

Example 3.1, and we want to find small roots of this polynomial. We make a matrix that spans a lattice, and this matrix will among other things consist of the coefficients of the polynomial we want to solve. We then make a vector that consists of x_0, which we do not know but is representing our small root. The vector and the matrix is constructed in such a way that multiplied together they will make a short vector that is contained in the lattice. Then we can use the LLL-algorithm to find short vectors, and if a small solution did exist in the first place, we will now be able to find it.

4.1 Coppersmith - the univariate case

We will go through Coppersmith's method for solving univariate polynomials modulo a composite integer as seen in [2]. We want to find small integer roots x_0 of a monic integer polynomial with degree δ in one variable

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{\delta-1} x^{\delta-1} + x^{\delta},$$

where $|x_0| < X$ and the a_i's are coefficients. We get the following modular equation

$$p(x_0) \equiv 0 \pmod{N},$$

where N is a large composite integer of unknown factorization. We select an integer

$$h \geq \max\left(\frac{\delta - 1 + \epsilon\delta}{\epsilon\delta^2}, \frac{7}{\delta}\right),$$

where $\epsilon > 0$. Using the first condition, $h \geq \left(\frac{\delta-1+\epsilon\delta}{\epsilon\delta^2}\right)$, we get that

$$\frac{h-1}{h\delta - 1} \geq \frac{1}{\delta} - \epsilon. \tag{4.1}$$

The second condition ensures that $h\delta \geq 7$. Both these conditions will be useful when evaluating the determinant in Subsection 4.1.2. Let $n = h\delta$. For each pair of integers (i, j) satisfying $0 \leq i < \delta$ and $1 \leq j < h$, we define the polynomial

$$q_{ij}(x) \equiv x^i p(x)^j.$$

For the solution x_0 we have that $p(x_0) = y_0 N$ for an integer y_0, so

$$\begin{aligned} q_{ij}(x_0) &= x_0^i p(x_0)^j \\ &= x_0^i y_0^j N^j \\ &\equiv 0 \pmod{N^j}. \end{aligned}$$

4.1.1 Building the matrix M in the univariate case

Coppersmith's method consists of making a matrix where the rows of this matrix spans a lattice. The matrix M of size $(2h\delta - \delta) \times (2h\delta - \delta)$ is upper triangular and is defined to be

$$M = \begin{pmatrix} A & C \\ 0 & D \end{pmatrix},$$

where we have divided M into 4 blocks. The upper left (hδ) × (hδ) block A is a diagonal matrix where the g'th entry on the diagonal is $\frac{X^{-g}}{h\delta}$ where g starts from zero and goes to $h\delta - 1$, $0 \leq g < h\delta$. The upper right block C is a matrix of size $(h\delta) \times (h\delta - \delta)$, where the rows are indexed by g, and the columns by $\gamma(i, j)$. The γ-function takes in values i and j where $0 \leq i < \delta$ and $1 \leq j < h$, and

$$\gamma(i, j) = h\delta + i + (j - 1)\delta.$$

The γ-function will give values in the following interval $h\delta \leq \gamma(i, j) < 2h\delta - \delta$. The element in C at the entry $(g, \gamma(i, j))$ is the coefficient of x^g in the polynomial $q_{ij}(x)$. The lower left $(h\delta - \delta) \times (h\delta - \delta)$ block is a zero matrix, and the lower right $(h\delta - \delta) \times (h\delta - \delta)$ block D is a diagonal matrix where the value on the diagonal in column number $\gamma(i, j)$ is N^j.

We construct a vector $\mathbf{r}$ whose elements consist of powers of the desired integer solution x_0 and the integer y_0,

$$\mathbf{r}(x_0) = \mathbf{r} = \left(1, x_0, x_0^2, \ldots, x_0^{h\delta-1}, -y_0, -x_0 y_0, \ldots, -x_0^{\delta-1} y_0, -y_0^2, x_0 y_0^2, \ldots, -x_0^{\delta-1} y_0^{h-1}\right)$$

where the first $h\delta$ entries are of the form

$$r_g = x_0^g$$

for $0 \leq g < h\delta$. The rest of the $h\delta - \delta$ entries can be written as

$$r_{\gamma(i,j)} = -x_0^i y_0^j$$

for $h\delta \leq \gamma(i, j) < 2h\delta - \delta$. We can write $\mathbf{r}$ as

$$\mathbf{r} = (r_g, r_{\gamma(i,j)}).$$

We emphasize that we still do not know x_0 or y_0, but we know that they exist by assumption.

This $\mathbf{r}$ vector is constructed in such a way that multiplied with the matrix M we get a vector $\mathbf{s}$ with 0's in the last entries,

$$\mathbf{s}(x_0) = \mathbf{s} = \left(\frac{1}{\sqrt{h\delta}}, \frac{(\frac{x_0}{X})}{\sqrt{h\delta}}, \frac{(\frac{x_0}{X})^2}{\sqrt{h\delta}}, \ldots, \frac{(\frac{x_0}{X})^{h\delta-1}}{\sqrt{h\delta}}, 0, \ldots, 0\right),$$

where

$$\mathbf{s} = \mathbf{r}M.$$

The first $h\delta$ elements of the $\mathbf{s}$ vector is given by

$$s_g = \frac{(\frac{x_0}{X})^g}{\sqrt{h\delta}},$$

and the right-hand elements become zero because

$$s_{\gamma(i,j)} = q_{ij}(x_0) - x_0^i y_0^j N^j = 0.$$

This $\mathbf{s}$ vector is an element in the lattice and spanned by the rows of M, which is a basis for the lattice.

We evaluate the norm of the $\mathbf{s}$ vector,

$$\|\mathbf{s}\| = \left[\sum_{g=0}^{h\delta-1} s_g^2 \right]^{1/2} < \left[\sum_{g=0}^{h\delta-1} \left(\frac{1}{\sqrt{h\delta}} \right)^2 \right]^{1/2} = 1, \tag{4.2}$$

and get that it is less than 1. Thus $\mathbf{s}$ is a short vector.

Since the last $h\delta - \delta$ elements of $\mathbf{s}$ is zero we can concentrate on the sublattice $\hat{M}$ of M consisting of vectors with 0 as right-hand elements. To find this $\hat{M}$ we observe that $p(x)$ and $q_{ij}(x)$ are monic polynomials, so certain rows in the upper right block of M will form an upper triangular matrix with 1 on the diagonal. For an example see Figure 5.1, where the 1's are highlighted in red. Thus we can multiply M with integer matrices with determinants of ±1 and end up with a matrix $\tilde{M}$ of the form

$$M \sim \tilde{M} = \begin{pmatrix} \hat{M} & 0 \\ A' & I \end{pmatrix}.$$

We can write this as

$$\tilde{M} = H_1 M,$$

or

$$M = H_1^{-1} \tilde{M},$$

where

$$H_1^{-1} = \begin{pmatrix} \mu_{11} & \cdots & \mu_{1,2h\delta-\delta} \\ \vdots & & \vdots \\ \mu_{2h\delta-\delta,1} & \cdots & \mu_{2h\delta-\delta,2h\delta-\delta} \end{pmatrix}.$$

For an example of $\tilde{M}$, see Figure 5.2 where the 1's in the identity block are highlighted in red again. We denote the rows of $\tilde{M}$ by $\mathbf{m}_i$,

$$\tilde{M} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \\ \vdots \\ \mathbf{m}_{2h\delta-\delta} \end{pmatrix}.$$

The upper-left $(h\delta) \times (h\delta)$ block of $\tilde{M}$ is $\hat{M}$ and this is the sublattice we want. It is $h\delta$-dimensional, and $\mathbf{s}$ is a relatively short element in this sublattice. We use the LLL-algorithm on the rows of $\hat{M}$ to find a LLL-reduced basis for $\mathcal{L}(\hat{M})$. We have a matrix B_2 where the rows are the LLL-reduced basis

$$B_2 = H_2\hat{M},$$

which is the same as

$$\hat{M} = H_2^{-1}B_2$$

where the entries in H_2^{-1} are h_{ij},

$$H_2^{-1} = \begin{pmatrix} h_{11} & \cdots & h_{1,h\delta} \\ \vdots & & \vdots \\ h_{h\delta,1} & \cdots & h_{h\delta,h\delta} \end{pmatrix}.$$

The matrix $\hat{M}$ can be written as

$$\hat{M} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{pmatrix},$$

where the rows are the first $h\delta$ rows of $\tilde{M}$, but shortened. We have removed the $h\delta - \delta$ zeros at the end of the row vectors, so the rows are now on the form $(m_{i1}, \ldots, m_{i,h\delta})$. We can write these row vectors as a linear combination of the LLL-reduced basis $\mathbf{b}_1, \ldots \mathbf{b}_n$,

$$\mathbf{m}_i = \sum_{j=1}^{h\delta} h_{ij}\mathbf{b}_j. \tag{4.3}$$

We know that $\mathbf{s}$ is short and lies in the lattice spanned by the rows of $\hat{M}$, so we can write $\mathbf{s}$ as an linear combination of the rows of $\hat{M}$,

$$\mathbf{s} = \sum_{i=1}^{h\delta} \alpha_i\mathbf{m}_i. \tag{4.4}$$

4.1.2 Analysis of the determinant of M

To be able to confine the $\mathbf{s}$ vector to a hyperplane we need to show that the norm of $\mathbf{s}$ is less than $\det(\mathcal{L})^{1/n}2^{(n-1)/4}$ as in Lemma 2.1 for a lattice $\mathcal{L}$. We need to combine the norm of the $\mathbf{s}$ vector with the determinant of the lattice. M is upper triangular and because of this, the determinant of M is a multiple of the entries on the diagonal

$$\begin{aligned}
\det(M) &= \frac{1}{\sqrt{h\delta}}\frac{1}{\sqrt{h\delta}X}\cdots\frac{1}{\sqrt{h\delta}X^{h\delta}}N\cdots N^{j} \\
&= \prod_{g=0}^{h\delta-1}\frac{1}{X^{g}\sqrt{h\delta}}\prod_{\gamma(i,j)=h\delta}^{2h\delta-\delta}N^{j} \\
&= \frac{N^{\frac{h\delta(h-1)}{2}}X^{\frac{-(h\delta)(h\delta-1)}{2}}}{\sqrt{h\delta}^{h\delta}} \\
&= [N^{\frac{h-1}{2}}X^{\frac{-(h\delta-1)}{2}}(h\delta)^{-\frac{1}{2}}]^{h\delta} \\
&= [N^{h-1}X^{-(h\delta-1)}(h\delta)^{-1}]^{\frac{h\delta}{2}}. \qquad (4.5)
\end{aligned}$$

The determinant of M equals the determinant of $\hat{M}$ in absolute value,

$$|\det(M)| = |\det(\hat{M})|\,|\det(I)| = |\det(\hat{M})|\,. \qquad (4.6)$$

Since $\mathbf{s}$ lies in the sublattice spanned by $\hat{M}$, we need to use Lemma 2.1 on the smaller matrix $\hat{M}$. First we need to show that

$$\|\mathbf{s}\| < \det(\hat{M})^{1/n}2^{-(n-1)/4}.$$

By (4.2) this will hold if

$$1 < \det(\hat{M})^{1/n}2^{-(n-1)/4}.$$

We know that $h\delta \geq 7$, which gives us $h\delta < 2^{\frac{h\delta-1}{2}}$. This we can use to get a new limit of the determinant of $\hat{M}$ combined with (4.5) and (4.6),

$$\det(\hat{M}) > [N^{h-1}X^{-(h\delta-1)}2^{\frac{-(h\delta-1)}{2}}]^{\frac{h\delta}{2}}.$$

We choose X to be $X \leq \frac{1}{2}N^{\frac{1}{\delta}-\epsilon}$, rewrite and get

$$\det(\hat{M}) > [N^{h-1-(h\delta-1)(\frac{1}{\delta}-\epsilon)}2^{\frac{(h\delta-1)}{2}}]^{\frac{h\delta}{2}}.$$

By using the condition $h-1 \geq (h\delta-1)(\frac{1}{\delta}-\epsilon)$ from (4.1) we get the expression

$$\det(\hat{M}) \geq 2^{\frac{(h\delta)(h\delta-1)}{4}},$$

which is the same as

$$\det(\hat{M})^{\frac{1}{h\delta}}2^{-\frac{h\delta-1}{4}} > 1.$$

Since $n = h\delta$ we get

$$\det(\hat{M})^{1/n}2^{-(n-1)/4} > 1. \qquad (4.7)$$

This shows that, by choosing

$$X \leq \frac{1}{2}N^{\frac{1}{\delta}-\epsilon}$$

we have

$$\|\mathbf{s}\| < 1 \leq \det(\hat{M})^{1/n}2^{-(n-1)/4}.$$

We can now use Lemma 2.1 on $\hat{M}$ and we get that the lattice element $\mathbf{s}$ lies in the hyperplane spanned by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n-1}$.

4.1.3 Finding the solution x_0

We have applied the LLL-algorithm on the rows of the matrix $\hat{M}$ of size $h\delta \times h\delta$, and got a basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ which satisfies (2.1)

$$\left\|\tilde{\mathbf{b}}_n\right\| \geq \det(\hat{M})^{1/n} 2^{(n-1)/4},$$

and together with (4.7) we get

$$\left\|\tilde{\mathbf{b}}_n\right\| \geq 1.$$

From Lemma 2.1 we have that any vector in the lattice generated by the rows of $\hat{M}$ with length less than 1 must lie in a hyperplane spanned by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{n-1}$.

We can write $\mathbf{s}$ as a linear combination of the LLL-reduced basis,

$$\mathbf{s} = \sum_{j=1}^{h\delta} \rho_j \mathbf{b}_j. \tag{4.8}$$

But we know that $\mathbf{s}$ is a short vector and by Lemma 2.1 we have that

$$\rho_{h\delta} = 0.$$

We can also write $\mathbf{s}$ as a linear combination of the rows of $\hat{M}$ as we did in (4.4) and we combine this with (4.3),

$$\begin{aligned}\mathbf{s} &= \sum_{i=1}^{h\delta} \alpha_i \mathbf{m}_i \\ &= \sum_{i=1}^{h\delta} \alpha_i \sum_{j=1}^{h\delta} h_{ij} \mathbf{b}_j \\ &= \sum_{j=1}^{h\delta} \mathbf{b}_j \sum_{i=1}^{h\delta} \alpha_i h_{ij}.\end{aligned}$$

From (4.8) we get that

$$\rho_j = \sum_{i=1}^{h\delta} \alpha_i h_{ij},$$

and

$$\rho_{h\delta} = \sum_{i=1}^{h\delta} \alpha_i h_{i,h\delta} = 0.$$

We know the value of $h_{i,h\delta}$ since it is the last column in H_2^{-1}.

We now look at the polynomial vectors $\mathbf{r}(x)$ and $\mathbf{s}(x)$, they are define as before but now we have the variable x instead of x_0. The $\mathbf{r}(x)$ vector is

$$\mathbf{r}(x) = \left(1, x, x^2, \ldots, x^{h\delta-1}, -y, -xy, \ldots, -x^{\delta-1}y, -y^2, xy^2, \ldots, -x^{\delta-1}y^{h-1}\right)$$

which is the same as

$$\mathbf{r}(x) = \left(1, x, \ldots, x^{h\delta-1}, -\frac{p(x)}{N}, -x\frac{p(x)}{N}, \ldots, -x^{\delta-1}\frac{p(x)}{N}, -\left(\frac{p(x)}{N}\right)^2, \ldots, -x^{\delta-1}\left(\frac{p(x)}{N}\right)^{h-1}\right).$$

Let $\mathbf{s}(x)$ be the $\mathbf{s}$ vector shortened, that is we have removed the zeros in the end and the variable is x instead of x_0,

$$\mathbf{s}(x) = \left(\frac{1}{\sqrt{h\delta}}, \frac{(\frac{x}{X})}{\sqrt{h\delta}}, \frac{(\frac{x}{X})^2}{\sqrt{h\delta}}, \ldots, \frac{(\frac{x}{X})^{h\delta-1}}{\sqrt{h\delta}}\right).$$

We define a vector of length $h\delta$

$$\sigma_l = \left(0, \ldots, 0, \frac{(\frac{1}{X})^{l-1}}{\sqrt{h\delta}}, 0, \ldots, 0\right),$$

where there are zeros except at the $(l-1)$th element in the vector. We can write σ_l as a linear combination of the rows of $\hat{M}$, since $\sigma_l \in \mathcal{L}(\hat{M})$

$$\sigma_l = \sum_{i=1}^{h\delta} \mu_{li}\mathbf{m}_i.$$

We can now write $\mathbf{s}(x)$ as

$$\begin{aligned}\mathbf{s}(x) &= \sum_{l=1}^{h\delta} \sigma_l x^{l-1} \\ &= \sum_{l=1}^{h\delta} x^{l-1} \sum_{i=1}^{h\delta} \mu_{li}\mathbf{m}_i \\ &= \sum_{i=1}^{h\delta} \mathbf{m}_i \sum_{l=1}^{h\delta} x^{l-1}\mu_{li},\end{aligned}$$

where

$$\alpha_i(x) = \sum_{l=1}^{h\delta} x^{l-1}\mu_{li}.$$

From before we have that

$$\begin{aligned}\rho_j &= \sum_{i=1}^{h\delta} \alpha_i(x) h_{ij} \\ &= \sum_{i=1}^{h\delta} \left(\sum_{l=1}^{h\delta} x^{l-1} \mu_{li} \right) h_{ij}.\end{aligned}$$

If $\mathbf{s}(x)$ is going to be a short vector, then $\rho_{h\delta}$ must be zero. We get the following equation,

$$\rho_{h\delta} = \sum_{i=1}^{h\delta} \left(\sum_{l=1}^{h\delta} x^{l-1} \mu_{li} \right) h_{i,h\delta} = 0.$$

We can solve this equation for x, since we know the other variables. We find $h_{i,h\delta}$ in the last column of H_2^{-1}, μ_{li} is in the matrix H_1^{-1}. We have the following equation we can solve for x_0,

$$c(x_0) = \sum_{i=1}^{h\delta} \left(\sum_{l=0}^{h\delta} x_0^{l-1} \mu_{li} \right) h_{i,h\delta} = 0.$$

This equation holds over the integers as well as modulo N and is easy to solve. This yields the desired solution x_0.

Written in terms of our matrices we can find our new polynomial in this way,

$$c(x) = \left[\mathbf{r}(x) H_1^{-1}\right]_{sh} \cdot ((H_2^{-1})_{h\delta})^T,$$

where $[\cdot]_{sh}$ denote the vector shortened, and $((H_2^{-1})_{h\delta})^T$ is the last column in H_2^{-1} transposed.

4.2 Coppersmith - the bivariate case

Finding small integer solutions to a polynomial equation in two variables are similar to the modular case with one variable. The basic outline is the same but this case is more technical. We will in the bivariate case get a matrix that is not square, which makes the approach harder.

We have a polynomial equation in two variables over the integers (not modulo N) with coefficients a_{ij}

$$p(x, y) = \sum_{j=0}^{\delta} \sum_{i=0}^{\delta} a_{ij} x^i y^j = 0$$

for which we want to find small integer solutions (x_0, y_0). The solution is bounded by X and Y, where $x_0 < |X|$ and $y_0 < |Y|$. We assume that $p(x, y)$ has δ as maximum degree in each variable separately and that $p(x, y)$ is irreducible over the integers, which means that the coefficients are relatively prime as a set.

We select an integer $k > \frac{2}{3\epsilon}$, for some $\epsilon > 0$. For all pairs of (i, j) where $0 \leq i < k$ and $0 \leq j < k$ we have the polynomial

$$q_{ij}(x, y) = x^i y^j p(x, y).$$

We see that if (x_0, y_0) is a root of $p(x, y)$ then it is also a root of $q_{ij}(x, y)$.

4.2.1 Building the matrix M_1 in the bivariate case

We have created several polynomials $q_{ij}(x, y) = x^i y^j p(x, y)$, which all have (x_0, y_0) as a root. We build a matrix M_1 which represent a lattice from the coefficients of these polynomials together with the bounds X and Y. This matrix is not square as before but rectangular, so we have a k-dimensional lattice in $\mathbb{Z}^n$ where $k < n$. This is not a problem for lattice basis reduction routines.

The matrix M_1 is a $(k+\delta)^2 \times ((k+\delta)^2 + k^2)$. It has $(k+\delta)^2$ rows which are indexed by

$$\gamma(g, h) = (k + \delta)g + h$$

with $0 \leq g, h < k + \delta$ and where $\gamma(g, h)$ gives values between 0 and $(k+\delta)^2 - 1$. M_1 has $(k + \delta)^2 + k^2$ columns, where the first $(k + \delta)^2$ columns are indexed by $\gamma(g, h)$ and the last k^2 columns are indexed by

$$\beta(i, j) = (k + \delta)^2 + ki + j$$

for $0 \leq i, j < k$. The β-function gives values from $(k + \delta)^2$ to $(k + \delta)^2 + k^2 - 1$.

M_1 consists of two blocks, D_1 and A_1,

$$M_1 = \begin{pmatrix} D_1 & A_1 \end{pmatrix}.$$

The block D_1 is diagonal of size $(k + \delta)^2 \times (k + \delta)^2$, where the $(\gamma(g, h), \gamma(g, h))$ entry on the diagonal of D_1 is $X^{-g}Y^{-h}$. The matrix is shown in Figure 4.1 where the blank spaces are zeros.

The $(\gamma(g, h), \beta(i, j))$ entry of A_1 is the coefficient of $x^g y^h$ in the polynomial $q_{ij}(x, y)$. The block A_1 has size $(k+\delta)^2 \times k^2$ and can be seen in Figure 4.2. The A_1 matrix consists again of k blocks that are similar. They are called E and are shown in Figure 4.3.

The columns of E, as seen in Figure 4.3, are all shifted versions of one vector $\mathbf{v}$. In the k columns in E the vectors gets shifted one row down for each column. The same E occurs again in A_1 in column $(k + 1)$, but now the block is shifted $\delta + 1$ rows down. The E block occurs k times in A_1, and for each time it gets moved $\delta + 1$ rows down.

We multiply M_1 with integer matrices with determinants ± 1 to produce a matrix M_2

$$M_1 \sim M_2 = \begin{pmatrix} \hat{M}_2 & 0 \\ C_2 & I \end{pmatrix}$$

where the identity matrix is $k^2 \times k^2$ and the zero matrix is $(2k\delta + \delta^2) \times k^2$. This can be done because the greatest common divisor of the coefficients of p is $1(p$ is irreducible).

$$D_1 = \begin{pmatrix}
1 &&&&&&&&&&& \\
& X^{-0}Y^{-1} &&&&&&&&&& \\
&& X^{-0}Y^{-2} &&&&&&&&& \\
&&& \ddots &&&&&&&& \\
&&&& X^{-0}Y^{-(k+\delta-1)} &&&&&&& \\
&&&&& X^{-1}Y^{-0} &&&&&& \\
&&&&&& X^{-1}Y^{-1} &&&&& \\
&&&&&&& \ddots &&&& \\
&&&&&&&& X^{-1}Y^{-(k+\delta-1)} &&& \\
&&&&&&&&& X^{-2}Y^{-0} && \\
&&&&&&&&&& \ddots & \\
&&&&&&&&&&& X^{-(k+\delta-1)}Y^{-(k+\delta-1)}
\end{pmatrix}$$

Figure 4.1: The diagonal matrix D_1.

The top $(2k\delta + \delta^2)$ rows of M_2 forms a sublattice of the original lattice and we call this sublattice $\hat{M_2}$. We do lattice basis reduction on these top $(2k\delta + \delta^2)$ rows of M_2, and we call the matrix consisting of these new rows M_3. We construct a vector $\mathbf{r}$ with length $(k+\delta)^2$ where the $\gamma(g,h)$ entry is $x_0^g y_0^h$,

$$\mathbf{r} = \left(x_0^0 y_0^0, x_0^0 y_0^1, \ldots, x_0^0 y_0^{k+\delta-1}, x_0^1 y_0^0, x_0^1 y_0^1, x_0^1 y_0^2, \ldots, x_0^{k+\delta-1} y_0^{k+\delta-1} \right).$$

The row vector $\mathbf{s}$ of length $(k+\delta)^2 + k^2$ is given by $\mathbf{s} = \mathbf{r}M_1$.

$$\mathbf{s} = \left(\left(\frac{x_0}{X}\right)^0 \left(\frac{y_0}{Y}\right)^1, \ldots, \left(\frac{x_0}{X}\right)^{k+\delta-1} \left(\frac{y_0}{Y}\right)^{k+\delta-1}, q_{01}(x_0,y_0), \ldots, q_{k-1,k-1}(x_0,y_0) \right)$$

Each of the first $(k+\delta)^2$ entries are given by

$$s_{\gamma(g,h)} = \left(\frac{x_0}{X}\right)^g \left(\frac{y_0}{Y}\right)^h$$

and the k^2 right-hand side entries are zero,

$$s_{\beta(i,j)} = q_{ij}(x_0,y_0) = 0.$$

Since $x_0 < |X|$ and $y_0 < |Y|$ the left-hand side elements satisfies

$$\left| s_{\gamma(g,h)} \right| < 1$$

and the length of the vector is

$$\|\mathbf{s}\| < k + \delta.$$

We know that $\mathbf{s}$ is generated by the rows of M_3 since $\mathbf{s}$ has zeros on its right hand side. To show that $\mathbf{s}$ is a "relatively short" vector in the lattice, we can confine it to a hyperplane by Lemma 2.1. To be able to do so we need to evaluate the determinant of the matrix M_1.

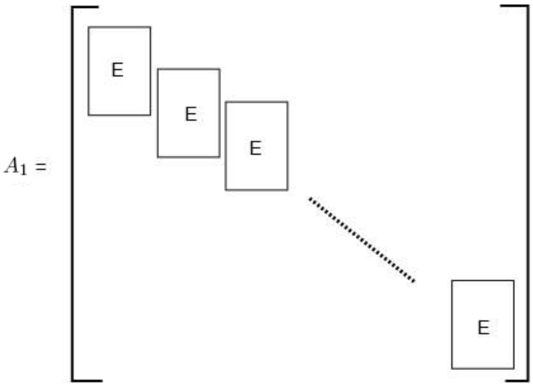

Figure 4.2: The matrix A_1 of size $(k+\delta)^2 \times k^2$, where the E block is given in Figure 4.3.

4.2.2 Analysis of the determinant

Let M_4 be the matrix obtained from M_1 by multiplying the $\gamma(g,h)$ row by X^gY^h and multiplying the $\beta(i,j)$ column by $X^{-i}Y^{-j}$. M_4 can then be written as

$$M_4 = \Delta_1 M_1 \Delta_2, \tag{4.9}$$

where Δ_1 is a $(k+\delta)^2 \times (k+\delta)^2$ diagonal matrix and Δ_2 is a $((k+\delta)^2+k^2) \times ((k+\delta)^2+k^2)$ diagonal matrix. We use the notation $diag(\cdot)$ to express a diagonal matrix where the diagonal elements are listed in the parenthesis. We have

$$\Delta_1 = \operatorname{diag}\Big(1, Y^1, Y^2, \dots, Y^{k-\delta-1}, X^1Y^0, X^1Y^1, X^1Y^2, \dots, X^1Y^{k+\delta-1}, \\ X^2Y^0, \dots, X^{k+\delta-1}Y^{k+\delta-1}\Big)$$

$$\Delta_2 = \operatorname{diag}\Big(1, 1, \dots, 1, X^0Y^0, X^0Y^{-1}, \dots, X^0Y^{-(k-1)}, X^{-1}Y^0, \dots, X^{-(k-1)}Y^{-(k-1)}\Big)$$

where the number of 1's in the beginning of Δ_2 are $(k+\delta)^2$. We get another diagonal matrix Δ_3 of size $(k+\delta)^2 \times (k+\delta)^2$ by removing k^2 1's on the diagonal of Δ_2.

$$\Delta_3 = \operatorname{diag}\Big(1, \dots, 1, X^0Y^0, X^0Y^{-1}, \dots, X^0Y^{-(k-1)}, X^{-1}Y^0, \dots, X^{-(k-1)}Y^{-(k-1)}\Big).$$

M_4 consists of two blocks, the left block is the identity matrix and the right-hand block is A_4 as shown in Figure 4.4 where the blocks are given in Figure 4.5.

$$M_4 = \begin{pmatrix} I & A_4 \end{pmatrix}$$

A_4 consists again of k blocks, they are identical as seen in Figure 4.5. The blocks are shifted $\delta + 1$ rows down from the previous block, in the same manner as for the A_1 matrix. Each column in block F represents the coefficients of a polynomial

$$x^iy^jp(xX, yY) = x^iy^j\breve{p}(x, y)$$

$$E = \begin{pmatrix}
a_{00} & 0 & \cdots & 0 \\
a_{01} & a_{00} & & \vdots \\
\vdots & a_{01} & \ddots & \\
 & \vdots & & a_{00} \\
a_{0\delta} & & & a_{01} \\
0 & a_{0\delta} & & \vdots \\
\vdots & 0 & & \\
0 & \vdots & & a_{0\delta} \\
a_{10} & 0 & & 0 \\
a_{11} & a_{10} & & \vdots \\
\vdots & a_{11} & & 0 \\
a_{1\delta} & \vdots & & a_{10} \\
0 & a_{1\delta} & & a_{11} \\
\vdots & 0 & & \vdots \\
a_{\delta\delta} & \vdots & & a_{1\delta} \\
0 & a_{\delta\delta} & & 0 \\
\vdots & 0 & & \vdots \\
 & \vdots & & a_{\delta\delta}
\end{pmatrix}$$

Figure 4.3: The block E of size $((\delta+1)(\delta+k)+1) \times k$ in the A_1 matrix.

where

$$\breve{p}(x,y) = p(xX, yY).$$

If we let $g = i + u$ and $h = j + v$, then the entries in M_1 and M_4 can be written in terms of the polynomials $p(x,y)$ and $\breve{p}(x,y)$,

$$(M_1)_{\gamma(g,h),\beta(i,j)} = a_{uv}$$

$$(M_4)_{\gamma(g,h),\beta(i,j)} = a_{uv}X^uY^v = \breve{p}(x,y).$$

Coppersmith [2] has a lemma that finds a $k^2 \times k^2$ matrix of a large determinant, where

$$W = \max_{ij} |a_{ij}| X^iY^j.$$

Lemma 4.1. *There is a $k^2 \times k^2$ submatrix of M_4, which we call S, with determinant at least $W^{k^2}2^{-6k^2\delta^2-2k^2}$ in absolute value,*

$$\det(S) \geq \left| W^{k^2}2^{-6k^2\delta^2-2k^2} \right|.$$

If the largest coefficient of $\breve{p}$ is $\breve{a}_{00}, \breve{a}_{0\delta}, \breve{a}_{\delta0}$or $\breve{a}_{\delta\delta}$, then the bound is W^{k^2}.

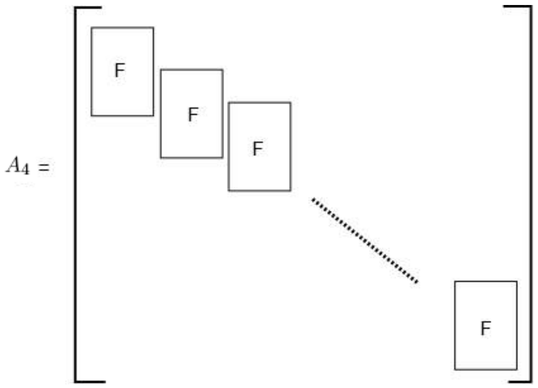

Figure 4.4: The matrix A_4 of size $(k+\delta)^2 \times k^2$, where the F block is given in Figure 4.5.

Lemma 4.1 finds a $k^2 \times k^2$ matrix from the right hand block of M_4, with large determinant. Select $2k\delta + \delta^2$ columns of the left hand block of M_4 to extend this to a $(k+\delta)^2 \times (k+\delta)^2$ matrix with the same determinant. To make this into a square matrix we need to fill in with 0's in the blank spaces. We can rearrange this new matrix and get a matrix with the $k^2 \times k^2$ submatrix of M_4 in the upper left corner with 0's beneath to fill up the rows, and with the columns with 1's to the right. These can be picked so that we have an identity block, or at least an invertible block. The determinant of this new matrix will be the determinant of the submatrix times the determinant of the invertible matrix. So the determinant will be greater than or equal to the determinant of the $k^2 \times k^2$ matrix we have in Lemma 4.1.

Let T be a $((k+\delta)^2 + k^2) \times (k+\delta)^2$ permutation matrix which selects the appropriate $(k+\delta)^2 = (2k\delta + \delta^2) + k^2$ columns of M_4, so that M_4T is a quadratic matrix. From the Lemma 4.1 and the discussion above we have

$$\det(M_4T) \geq \left| W^{k^2} 2^{-6k^2\delta^2 - 2k^2} \right|$$

and from (4.9) we get

$$\det(\Delta_1 M_1 \Delta_2 T) \geq \left| W^{k^2} 2^{-6k^2\delta^2 - 2k^2} \right|.$$

We choose T such that

$$\Delta_2 T = T \Delta_3$$

and get

$$\det(\Delta_1 M_1 T \Delta_3) \geq \left| W^{k^2} 2^{-6k^2\delta^2 - 2k^2} \right|. \tag{4.10}$$

We compute the determinants of Δ_1, Δ_2 and Δ_3,

$$\det(\Delta_1) = \prod_{h=0}^{k+\delta-1} \prod_{g=0}^{k+\delta-1} X^g Y^h = (XY)^{\frac{k+\delta)^2(k+\delta-1)}{2}}$$

$$F = \begin{pmatrix}
p_{00} & 0 & \cdots & 0 \\
p_{01}Y & p_{00} & & \vdots \\
\vdots & p_{01}Y & \ddots & \\
p_{0\delta}Y^{\delta} & \vdots & & p_{00} \\
0 & p_{0\delta}Y^{\delta} & & p_{01}Y \\
\vdots & 0 & & \vdots \\
0 & \vdots & & p_{0\delta}Y^{\delta} \\
p_{10}X & 0 & & 0 \\
p_{11}XY & p_{10}X & & \vdots \\
\vdots & p_{11}XY & & 0 \\
p_{1\delta}XY^{\delta} & \vdots & & p_{10}X \\
0 & p_{1\delta}XY^{\delta} & & p_{11}XY \\
\vdots & 0 & & \vdots \\
p_{\delta\delta}X^{\delta}Y^{\delta} & \vdots & & p_{1\delta}XY^{\delta} \\
0 & p_{\delta\delta}X^{\delta}Y^{\delta} & & 0 \\
\vdots & 0 & & \vdots \\
 & \vdots & & p_{\delta\delta}X^{\delta}Y^{\delta}
\end{pmatrix}$$

Figure 4.5: The F block of size $((\delta+1)(\delta+k)+1) \times k$ in the A_4 matrix.

and

$$\det(\Delta_2) = \det(\Delta_3) = \prod_{i=0}^{k-1}\prod_{j=0}^{k-1} X^{-i}Y^{-j} = (XY)^{\frac{k^2(k-1)}{2}},$$

where Δ_2 and Δ_3 have the same determinant since they both are diagonal matrices and the only difference is some 1's on the diagonal. We multiply the determinants of Δ_1 and Δ_2 and get

$$\begin{aligned}\det(\Delta_1)\det(\Delta_2) &= (XY)^{[(k+\delta)^2(k+\delta-1)-k^2(k-1)]/2} \\ &= (XY)^{[3k^2\delta+k(3\delta^2-2\delta)+(\delta^3-\delta^2)]/2}. \qquad (4.11)\end{aligned}$$

We can rewrite (4.10) to

$$|\det(\Delta_1)|\,|\det(M_1T)|\,|\det(\Delta_3)| \geq |W^{k^2}2^{-6k^2\delta^2-2k^2}|$$

and

$$|\det(M_1T)| \geq \frac{W^{k^2}2^{-6k^2\delta^2-2k^2}}{\det(\Delta_1)\det(\Delta_2)}, \qquad (4.12)$$

where we use that the determinant of Δ_2 and Δ_3 are the same. We combine (4.11) and (4.12),

$$|\det(M_1T)| \geq W^{k^2}2^{-6k^2\delta^2-2k^2}(XY)^{-[3k^2\delta+k(3\delta^2-2\delta)+(\delta^3-\delta^2)]/2}.$$

Let this lower bound be called Z.

We take M_3 as described before, and extend it with the last rows of M_2 so that we get a matrix of size $(k+\delta)^2 \times (k+\delta)^2 + k^2$. We call this matrix M_3'. We now get $\hat{M}_3$ as our sublattice.

$$M_1T \sim M_3'T = \begin{pmatrix} \hat{M}_3 & 0 \\ C_3 & I \end{pmatrix}.$$

$M_3'T$ is obtained from M_1T by multiplying with integer matrices with determinant 1, so the determinants of the two matrices are the same in absolute value. Since the M_3T matrix have a block lower triangular structure, we get that the determinant of the M_3T matrix equals that of $\hat{M}_3$

$$|\det(M_1T)| = \left|\det(M_3'T)\right| = \left|\det(\hat{M}_3)\right| |\det(I)| = \left|\det(\hat{M}_3)\right|. \tag{4.13}$$

$$\left|\det(M_3'T)\right| = |\det(M_1T)| \geq Z. \tag{4.14}$$

The row $\mathbf{s}T$ in M_3T is obtained from $\mathbf{s}$ by deleting columns and its Euclidean norm is bounded by the length of $\mathbf{s}$,

$$\|\mathbf{s}T\| \leq \|\mathbf{s}\| < k+\delta. \tag{4.15}$$

From (4.13) and (4.14) we get

$$|\det(\hat{M}_3)| = |\det(M_3T)| \geq Z. \tag{4.16}$$

To apply Lemma 2.1 to $\hat{M}_3$ and $\mathbf{s}T$, with $n = 2k\delta + \delta^2$ the norm of $\mathbf{s}T$ must satisfy

$$\|\mathbf{s}T\| < \det(\hat{M}_3)^{\frac{1}{n}}2^{-\frac{n-1}{4}}.$$

This requires that (4.15) and (4.16) holds in addition to

$$k+\delta \leq Z^{\frac{1}{n}}2^{-\frac{n-1}{4}},$$

which translates to

$$(k+\delta)^n \leq Z \times 2^{-\frac{n(n-1)}{4}}. \tag{4.17}$$

We recall that $n = 2k\delta + \delta^2$ and

$$Z = W^{k^2}2^{-6k^2\delta^2-2k^2}(XY)^{-\frac{[3k^2\delta+k(3\delta^2-2\delta)+(\delta^3-\delta^2)]}{2}}.$$

After some tedious computations (4.17) can be written as

$$XY \leq W^{2/3\delta-\epsilon'}2^{14\delta/3-o(\delta)}$$

where

$$\epsilon' \approx \frac{2}{3k}\left(1-\frac{2}{3\delta}\right).$$

We now assume that XY satisfies this bound. We use the LLL-algorithm on $\hat{M}_3$, and we can use Lemma 2.1 to confine the short vectors to a hyperplane, included the $\mathbf{s}T$ vector.

4.2.3 Finding the solution (x_0, y_0)

We do this exactly in the same way as in the univariate case. But instead of the **s** vector we now use the **s**T vector which is shortened. We will need to find more matrices, but we will in the end get an equation on the same form, but with two variables instead of one. We denote the equation as

$$c(x_0, y_0) = \sum_{g=0}^{k+\delta-1} \sum_{h=0}^{k+\delta-1} a_{gh} x_0^g y_0^h = 0.$$

All the multiples of $p(x, y)$ of sufficiently low degree are already used to define the sublattice $\hat{M}$ and $c(x, y)$ is hence not a multiple of $p(x, y)$.

Definition 4.1. *Let $f_a(x)$ and $g_c(x)$ be in $\mathbb{Z}[\mathbf{x}]$ and let $a_0, ..., a_n, c_0, ..., c_m$ be elements in $\mathbb{Z}$. We have two polynomials:*

$$f_a(x) = a_0 + a_1 x + ... + a_n x^n,$$

$$g_c(x) = c_0 + c_1 x + ... + c_m x^m.$$

We define the resultant *of f_a, g_c, $\mathrm{res}(f_a, g_c)$, to be the determinant of this $(m+n) \times (m+n)$ matrix*

$$\begin{pmatrix} a_0 & a_1 & \cdots & a_n & & \\ & a_0 & a_1 & \cdots & a_n & \\ & & & \cdots & & \\ & & a_0 & a_1 & \cdots & a_n \\ c_0 & c_1 & \cdots & c_m & & \\ & c_0 & c_1 & \cdots & c_m & \\ & & & \cdots & & \\ & & c_0 & c_1 & \cdots & c_m \end{pmatrix}$$

where there are m rows with a and c rows of w. The blank spaces are filled with zeros.

We take the resultant of the polynomials $p(x, y)$ and $c(x, y)$

$$q(x) = \mathrm{res}_y(c(x, y), p(x, y)).$$

Since $p(x, y)$ is an irreducible polynomial the resultant will give a nonconstant integer polynomial. Since $q(x)$ only depends on x it is easy to compute its roots, which will include x_0. Given x_0 we can find y's by solving $p(x_0, y) = 0$, where y_0 will then be one of the roots. We now have the desired solution (x_0, y_0).

4.3 Howgrave-Graham - the univariate case

An alternative technique for finding small roots of univariate modular equations was put forward by Nicholas Howgrave-Graham in [5]. It is related to Coppersmith's method, but is more direct. Howgrave-Graham's method uses polynomials that are multiples of

$p(x)$ and N, whereas Coppersmith only uses $p(x)$. The matrix in Howgrave-Graham's method are smaller than in Coppersmith's method, and it is lower triangular so it is easy to find the determinant.

Howgrave-Graham proved that Coppersmith's algorithm and his algorithm are equivalent, but Howgrave-Graham's may be preferred for computational efficiency.

As for Coppersmith's method the purpose is to find the small roots of a monic, univariate modular equation

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{\delta-1}x^{\delta-1} + x^\delta \equiv 0 \pmod{N}.$$

The solution is x_0, and δ is the maximal degree for the polynomial. For any polynomial $h(x)$ and natural number X we have the following upper bound on the absolute size of $h(x)$

$$\begin{aligned}|h(x)| &\leq |a_0| + |a_1x| + \cdots + |x^\delta| \\ &\leq |a_0| + |a_1X| + \cdots + |X^\delta|\end{aligned}$$

for all $|x| \leq X$.

4.3.1 Building the matrix M_5

Howgrave-Graham uses a different indexation, where $1 \leq i, j \leq n$, $v = \left\lfloor \frac{i-1}{\delta} \right\rfloor$ and $u = i - 1 - \delta v$. We define a lower triangular $(h\delta) \times (h\delta)$ matrix M_5, where $h \geq 2$ and X a natural number. The matrix $M_5 = (m_{ij})$ as seen in Figure 4.6 have entries

$$m_{ij} = e_{ij}X^{j-1},$$

where e_{ij} is the coefficient of x^{j-1} in

$$q_{u,v}(x) = N^{(h-1-v)}x^u(p(x))^v.$$

We observe that this matrix is smaller than the matrix in Coppersmith's univariate case.

For all $u, v \geq 0$ the polynomial equation

$$q_{u,v}(x_0) = 0 \pmod{N^{h-1}}$$

have a solution x_0. The determinant of the matrix M_5 will be the entries on the diagonal,

$$\det(M_5) = X^{h\delta(h\delta-1)}N^{(h\delta(h-1))/2}. \tag{4.18}$$

Notice that we can find the determinant directely, and we do not need to find a sublattice of M_5. We can work directely with M_5.

$$M_5 = \begin{bmatrix}
N^{h-1} & 0 & \cdots & & & & & & & \cdots & 0 \\
0 & N^{h-1}X & 0 & \cdots & & & & & & & \vdots \\
\vdots & & \ddots & \cdots & & & & & & & \\
0 & \cdots & 0 & N^{h-1}X^{\delta-1} & 0 & \cdots & & & & & \\
a_0N^{h-2} & a_1N^{h-2}X & \cdots & & N^{h-2}X^{\delta} & 0 & \cdots & & & & \\
0 & a_0N^{h-2}X & \cdots & & & N^{h-2}X^{\delta+1} & & & & & \\
0 & 0 & a_0N^{h-2}X^2 & \cdots & & & N^{h-2}X^{\delta+2} & & & & \\
\vdots & & & & & & & \ddots & & & \\
0 & \cdots & & & & a_0N^{h-2}X^{\delta-1} & a_1N^{h-2}X^{\delta} & \cdots & N^{h-2}X^{2\delta-1} & & \\
a_0^2N^{h-3} & \cdots & & & & & & & & N^{h-3}X^{2\delta} & \\
\vdots & & & & & & & & & & \ddots \\
0 & & & & & & & & & & 0
\end{bmatrix}$$

Figure 4.6: M_5 matrix

4.3.2 Finding the solution x_0

Let B_5 be the LLL-reduced basis of the rows of M_5, and the first row vector of B_5 is $\mathbf{b}_1$. We have that

$$B_5 = H_5M_5,$$

for a matrix $H_5 = B_5M_5^{-1}$. The equation in Proposition 2.2 and (4.18) give

$$\|\mathbf{b}_1\| \leq 2^{(h\delta-1)/4}X^{(h\delta-1)/2}N^{(h-1)/2},$$

which is exactly the same as in Coppersmith's case. Let $\mathbf{b}_1 = \mathbf{c}M$ for some $\mathbf{c} \in \mathbb{Z}^n$. We look at the norm of $\mathbf{b}_1$,

$$\begin{aligned}
\|\mathbf{b}_1\| &\geq \frac{1}{\sqrt{h\delta}} \sum_{j=1}^{h\delta} |b_{1i}| \\
&= \frac{1}{\sqrt{h\delta}} \left(\left| \sum_{i=1}^{h\delta} c_i m_{i,1} \right| + \left| \sum_{i=1}^{h\delta} c_i m_{i,2} \right| + \cdots + \left| \sum_{i=1}^{h\delta} c_i m_{i,h\delta} \right| \right) \\
&= \frac{1}{\sqrt{h\delta}} \left(\left| \sum_{i=1}^{h\delta} c_i e_{i,1} \right| + \left| \left(\sum_{i=1}^{h\delta} c_i e_{i,2} \right) X \right| + \cdots + \left| \left(\sum_{i=1}^{h\delta} c_i e_{i,h\delta} \right) X^{h\delta-1} \right| \right) \\
&\geq \frac{1}{\sqrt{h\delta}} |h(x)| \qquad (4.19)
\end{aligned}$$

for all $|x| \leq X$ where

$$\begin{aligned} h(x) &= \sum_{i=1}^{h\delta} c_i e_{i,1} + \left(\sum_{i=1}^{h\delta} c_i e_{i,2} \right) x + \cdots + \left(\sum_{i=1}^{h\delta} c_i e_{i,h\delta} \right) x^{h\delta-1} \\ &= c_1 \sum_{j=1}^{h\delta} e_{1,j} x^{j-1} + c_2 \sum_{j=1}^{h\delta} e_{2,j} x^{j-1} + \cdots + c_{h\delta} \sum_{j=1}^{h\delta} e_{h\delta,j} x^{j-1}. \end{aligned} \tag{4.20}$$

From (4.19) we observe that $\mathbf{b}_1$ is almost an upper bound for the polynomial $h(x)$, for $|x| \leq X$. Since each sum in (4.20) is zero modulo N^{h-1}, we have that

$$h(x_0) \equiv 0 \pmod{N^{h-1}}.$$

We combine (4.19) and (4.20) and we can form a polynomial $h(x)$ that satisfies $h(x_0) \equiv 0 \pmod{N^{h-1}}$ and

$$|h(x)| \leq \left(2^{\frac{(h\delta-1)}{4}} \sqrt{h\delta} \right) X^{\frac{h\delta-1}{2}} N^{\frac{h-1}{2}}$$

for all $|x| \leq X$. Thus choosing

$$X = \left\lceil \left(2^{-\frac{1}{2}} (h\delta)^{-\frac{1}{h\delta-1}} \right) N^{\frac{h-1}{h\delta-1}} \right\rceil - 1$$

shows that we can make a polynomial $h(x)$ such that $h(x_0) = 0 \pmod{N^{h-1}}$ and $|h(x)| < N^{h-1}$ for all $|x| \leq X$. This implies that $h(x_0) = 0$ over the integers, for any x_0 that satisfies $|x_0| \leq X$ and $p(x_0) = 0 \pmod N$.

Let us now look at how we can find $h(x)$ by using our matrices. Let the columns of M_5 be $\mathbf{m}_j$,

$$M_5 = \begin{pmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \cdots & \mathbf{m}_{h\delta} \end{pmatrix}.$$

We create a vector $\mathbf{r}(x)$

$$\mathbf{r}(x) = \left(1, \frac{x}{X}, \left(\frac{x}{X}\right)^2, \ldots, \left(\frac{x}{X}\right)^{h\delta-1} \right).$$

The polynomial $h(x)$ is created in such a way that

$$\begin{aligned} h(x) &= h_1 M_5 \cdot \mathbf{r}(x) \\ &= \mathbf{b}_1 \cdot \mathbf{r}(x), \end{aligned}$$

where h_1 is the first row in H_5, noticing that $\mathbf{b}_1 = h_1 M_5$. As shown above this vector will have x_0 as a root. We can now solve this univariate polynomial equation over the integers. After finding the solutions one can test each solution to see if it satisfies $p(x_0) \equiv 0 \pmod N$.

Chapter 5

An example

Given the polynomial

$$p(x) = x^3 - 4x^2 - 3x - 10 \pmod{1131}$$

where

$$(p(x))^2 = x^6 - 8x^5 + 10x^4 + 89x^2 + 60x + 100.$$

The degree δ is 3, and we choose h to be 3 so the requirement $h\delta \geq 7$ is fulfilled. Our aim is to find the small roots of the polynomial equation,

$$p(x) = x^3 - 4x^2 - 3x - 10 \pmod{1131} = 0$$

and we set the requirement $X = 6$, which means that we will find roots $|x_0| < 6$, if there exist some. In this small example it is easy to see that $x = 5$ is a root.

5.1 Coppersmith's method

We will now try to find the small roots of our polynomial using Coppersmith's method. We make the matrix as in Coppersmith's univariate case, and get a 15×15 M matrix as shown in Figure 5.1. We take the determinant of the matrix and get

$$\det(M) = \frac{153841020405122283630137}{20301882330868921115530247326924 8}.$$

We now want to do row operations on the M matrix, where we can interchange two rows, and multiply one row and add it to another row. After doing about 50 operations we get the following matrix $\tilde{M}$ on the desirable form, as shown in Figure 5.2. We take the determinant of this matrix as well and get

$$\det(\tilde{M}) = \frac{153841020405122283630137}{203018823308689211155302473269248},$$

and we see that the two determinants are the same as expected. The matrix $\tilde{M}$ is block upper triangular and we can now look at the smaller matrix $\hat{M}$ which is a sublattice.

$$\begin{bmatrix}
\frac{1}{9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 100 & 0 & 0 \\
0 & \frac{1}{54} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & -10 & 0 & 60 & 100 & 0 \\
0 & 0 & \frac{1}{6^2 9} & 0 & 0 & 0 & 0 & 0 & 0 & -4 & -3 & -10 & 89 & 60 & 100 \\
0 & 0 & 0 & \frac{1}{6^3 9} & 0 & 0 & 0 & 0 & 0 & 1 & -4 & -3 & 4 & 89 & 100 \\
0 & 0 & 0 & 0 & \frac{1}{6^4 9} & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 10 & 4 & 89 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{6^5 9} & 0 & 0 & 0 & 0 & 0 & 1 & -8 & 10 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6^6 9} & 0 & 0 & 0 & 0 & 0 & 1 & -8 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6^7 9} & 0 & 0 & 0 & 0 & 0 & 1 & -8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6^8 9} & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1131^2
\end{bmatrix}$$

Figure 5.1: The M matrix

We see $\hat{M}$ in Figure 5.3. We use the LLL-algorithm on this matrix and end up with the matrix B_2, as seen in Figure 5.4. The first row is $\mathbf{b}_1$, and so on.

We have that $\tilde{M} = H_1 M$, and we have computed H_1^{-1} as seen in Figure 5.5. After using the LLL-algorithm on $\hat{M}$ we get the B_2 matrix, and since the LLL-algorithm is just row operations we can write $B_2 = H_2 \hat{M}$. The inverse of the H_2 matrix is in Figure 5.6.

We have the vector $\mathbf{r}(x)$,

$$\mathbf{r}(x) = \left(1, x, \ldots, x^{h\delta-1}, -\frac{p(x)}{N}, -x\frac{p(x)}{N}, \ldots, -x^{\delta-1}\frac{p(x)}{N}, -\left(\frac{p(x)}{N}\right)^2, \ldots, -x^{\delta-1}\left(\frac{p(x)}{N}\right)^{h-1}\right)$$

which in this example is

$$\mathbf{r}(x) = \left(1, x, x^2, \ldots, x^8, -\frac{p(x)}{1131}, -x\frac{p(x)}{1131}, -x^2\frac{p(x)}{1131}, -\left(\frac{p(x)}{1131}\right)^2, x\left(\frac{p(x)}{1131}\right)^2, x^2\left(\frac{p(x)}{1131}\right)^2\right)$$

where $p(x)$ is our polynomial.

$$\begin{bmatrix}
\frac{1}{9} & 0 & 0 & \frac{5}{972} & \frac{5}{1458} & \frac{95}{34992} & \frac{245}{104976} & \frac{815}{419904} & \frac{125}{78732} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{54} & 0 & \frac{1}{648} & \frac{11}{5832} & \frac{97}{69984} & \frac{121}{104976} & \frac{2447}{2519424} & \frac{2015}{2519424} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{324} & \frac{1}{486} & \frac{19}{11664} & \frac{49}{34992} & \frac{163}{139968} & \frac{305}{314928} & \frac{12047}{15116544} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{377}{648} & -\frac{377}{972} & -\frac{7163}{23328} & -\frac{377}{1296} & -\frac{214513}{839808} & -\frac{92365}{419904} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{377}{3888} & -\frac{377}{5832} & -\frac{4147}{69984} & -\frac{4147}{69984} & -\frac{275587}{5038848} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{377}{23328} & -\frac{377}{17496} & -\frac{377}{15552} & -\frac{10933}{419904} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{46656} & -\frac{142129}{34992} & -\frac{142129}{31104} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{279936} & -\frac{142129}{209952} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{1679616} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{1944} & \frac{1}{2916} & \frac{19}{69984} & \frac{1}{3888} & \frac{569}{2519424} & \frac{245}{1259712} & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{11664} & \frac{1}{17496} & \frac{11}{209952} & \frac{11}{209952} & \frac{731}{15116544} & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{69984} & \frac{1}{52488} & \frac{1}{46656} & \frac{29}{1259712} & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{419904} & \frac{1}{314928} & \frac{1}{279936} & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2519424} & \frac{1}{1889568} & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{15116544} & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Figure 5.2: The $\tilde{M}$ matrix.

$$\begin{bmatrix}
\frac{1}{9} & 0 & 0 & \frac{5}{972} & \frac{5}{1458} & \frac{95}{34992} & \frac{245}{104976} & \frac{815}{419904} & \frac{125}{78732} \\
0 & \frac{1}{54} & 0 & \frac{1}{648} & \frac{11}{5832} & \frac{97}{69984} & \frac{121}{104976} & \frac{2447}{2519424} & \frac{2015}{2519424} \\
0 & 0 & \frac{1}{324} & \frac{1}{486} & \frac{19}{11664} & \frac{49}{34992} & \frac{163}{139968} & \frac{305}{314928} & \frac{12047}{15116544} \\
0 & 0 & 0 & -\frac{377}{648} & -\frac{377}{972} & -\frac{7163}{23328} & -\frac{377}{1296} & -\frac{214513}{839808} & -\frac{92365}{419904} \\
0 & 0 & 0 & 0 & -\frac{377}{3888} & -\frac{377}{5832} & -\frac{4147}{69984} & -\frac{4147}{69984} & -\frac{275587}{5038848} \\
0 & 0 & 0 & 0 & 0 & -\frac{377}{23328} & -\frac{377}{17496} & -\frac{377}{15552} & -\frac{10933}{419904} \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{46656} & -\frac{142129}{34992} & -\frac{142129}{31104} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{279936} & -\frac{142129}{209952} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{142129}{1679616}
\end{bmatrix}$$

Figure 5.3: $\hat{M}$ matrix

We compute $\mathbf{r}(x)H_1^{-1}$ to get a new polynomial

$$\mathbf{r}(x)H_1^{-1} = \left(1, x, x^2, -\frac{p(x)}{1131}, -x\frac{p(x)}{1131}, -x^2\frac{p(x)}{1131}, -\left(\frac{p(x)}{1131}\right)^2, x\left(\frac{p(x)}{1131}\right)^2, x^2\left(\frac{p(x)}{1131}\right)^2,\right.$$
$$\left. 0, 0, 0, 0, 0, 0\right).$$

We now shorten this vector by removing the last 6 entries with zeros. The vector is now 9 elements long, so it can be multiplied with the last column of H_2^{-1} matrix in Figure 5.6. We get

$$[\mathbf{r}(x)H_1^{-1}]_{sh} \cdot ((H_2^{-1})_9)^T = -\frac{1}{1131^2}(p(x))^2 = 0.$$

Finding the roots of this polynomial is the same as finding the roots of $p(x)$, except that we do not have a modular equation. We solve and get that $x_0 = 5$ is a root. It is smaller than the X bound which was 6 in this example, and we got the desired root.

5.2 Howgrave-Graham's method

We now try to solve the same polynomial equation using Howgrave-Graham's method. We make the matrix, as seen in Figure 5.7. This matrix is lower triangular, so we do not have to do all the computations as we did in Coppersmith's case to get it on this form.

The determinant of the M_5 matrix is

$$\det(M_5) = 86757285942377701182907139222879294247536272365689 2416.$$

$$\begin{bmatrix}
0 & 0 & \frac{1}{324} & \frac{1}{486} & \frac{19}{11664} & \frac{49}{34992} & \frac{163}{139968} & \frac{305}{314928} & \frac{12047}{15116544} \\
0 & \frac{1}{54} & -\frac{1}{324} & -\frac{1}{1944} & \frac{1}{3888} & -\frac{1}{69984} & -\frac{5}{419904} & \frac{7}{2519424} & \frac{43}{15116544} \\
0 & 0 & \frac{1}{81} & \frac{2}{243} & \frac{19}{2916} & -\frac{739}{69984} & -\frac{197}{11664} & -\frac{25657}{1259712} & -\frac{43175}{1889568} \\
0 & 0 & -\frac{1}{324} & -\frac{1}{486} & -\frac{19}{11664} & \frac{1033}{69984} & \frac{317}{15552} & \frac{29317}{1259712} & -\frac{224405}{3779136} \\
0 & 0 & \frac{11}{324} & \frac{11}{486} & -\frac{461}{5832} & -\frac{37}{2187} & -\frac{469}{139968} & -\frac{19}{157464} & \frac{23233}{3779136} \\
\frac{1}{9} & 0 & -\frac{1}{324} & \frac{1}{324} & \frac{7}{3888} & \frac{23}{17496} & \frac{491}{419904} & \frac{1225}{1259712} & \frac{11953}{15116544} \\
0 & 0 & -\frac{11}{324} & -\frac{11}{486} & -\frac{209}{11664} & \frac{1279}{8748} & \frac{28367}{139968} & -\frac{695261}{2519424} & -\frac{4265}{1889568} \\
0 & \frac{1}{27} & \frac{41}{162} & -\frac{797}{1944} & \frac{157}{3888} & \frac{313}{69984} & -\frac{1609}{104976} & \frac{9311}{2519424} & \frac{167177}{15116544} \\
0 & 0 & -\frac{1}{27} & -\frac{2}{81} & -\frac{1207}{3888} & \frac{3517}{2592} & -\frac{160673}{139968} & \frac{55063}{419904} & -\frac{87773}{5038848}
\end{bmatrix}$$

Figure 5.4: The B_2 matrix

The M_5 matrix is LLL-reduced by the LLL-algorithm and we get the B_5 matrix, $B_5 = H_5 M_5$. The first row is $\mathbf{b}_1$, and so on. The B_5 matrix can be seen in Figure 5.8, and the H_5 matrix in Figure 5.9.

We now have all the information we need to make the new polynomial which have x_0 as a root. We take the dot product of $\mathbf{b}_1$ and $\mathbf{r}(x)$ which is our constructed vector

$$\mathbf{r}(x) = \left(1, \frac{x}{X}, \left(\frac{x}{X}\right)^2, \ldots, \left(\frac{x}{X}\right)^{h\delta-1}\right).$$

In this example it will be

$$\mathbf{r}(x) = \left(1, \frac{x}{X}, \left(\frac{x}{X}\right)^2, \ldots, \left(\frac{x}{X}\right)^5\right).$$

We have that $h(x) = \mathbf{b}_1 \cdot \mathbf{r}(x)$, and we get the following polynomial

$$\begin{aligned}
h(x) &= 100 + \frac{360}{6}x + \frac{3204}{6^2}x^2 + \frac{864}{6^3}x^3 + \frac{12960}{6^4}x^4 - \frac{62208}{6^5}x^5 + \frac{46656}{6^6}x^6 \\
&= 100 + 60x + 89x^2 + 4x^3 + 10x^4 - 8x^5 + x^6.
\end{aligned}$$

This polynomial have x_0 as a root, and we are able to find the roots, one of which is $x_0 = 5$.

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 100 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & -10 & 0 & 60 & 100 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & -3 & -10 & 89 & 60 & 100 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & -3 & 4 & 89 & 100 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 10 & 4 & 89 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -8 & 10 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -8 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1131 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1279161 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1279161 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1279161
\end{bmatrix}$$

Figure 5.5: H_1^{-1} matrix

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-141 & -2 & 6 & 0 & 3 & 0 & 0 & 1 & 0 \\
-19 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\
-4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-477 & 0 & 145 & 14 & -3 & 0 & 4 & 0 & 1 \\
-44 & 0 & 15 & 5 & 0 & 0 & 1 & 0 & 0 \\
-3 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Figure 5.6: H_2^{-1} matrix

$$\begin{bmatrix}
1279161 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 7674966 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 7674966 & 0 & 0 & 0 & 0 & 0 & 0 \\
-11310 & -20358 & -162864 & 244296 & 0 & 0 & 0 & 0 & 0 \\
0 & -67860 & -122148 & -162864 & 244296 & 0 & 0 & 0 & 0 \\
0 & 0 & 407160 & -732888 & -5863104 & 8794656 & 0 & 0 & 0 \\
100 & 360 & 3204 & 864 & 12960 & -62208 & 46656 & 0 & 0 \\
0 & 600 & 2160 & 19224 & 5184 & 77760 & -373248 & 279936 & 0 \\
0 & 0 & 3600 & 12960 & 115344 & 31104 & 466560 & -2239488 & 1679616
\end{bmatrix}$$

Figure 5.7: The M_5 matrix

$$\begin{bmatrix}
100 & 360 & 3204 & 864 & 12960 & -62208 & 46656 & 0 & 0 \\
-11310 & -20358 & -162864 & 244296 & 0 & 0 & 0 & 0 & 0 \\
0 & -67860 & -122148 & -162864 & 244296 & 0 & 0 & 0 & 0 \\
400 & 2040 & 14976 & 22680 & 57024 & -171072 & -186624 & 279936 & 0 \\
1279161 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1500 & 76260 & 184608 & 284904 & 91368 & -513216 & -699840 & -839808 & 1679616 \\
-41950 & -718830 & -1124856 & -712584 & -1334880 & 909792 & 1679616 & 1959552 & 1679616 \\
-235420 & -1884156 & 2275038 & 1456488 & 1435968 & 1578528 & 1492992 & 1679616 & 1679616 \\
-269950 & 5319816 & 1032174 & 1187784 & 2818800 & 1874016 & 1586304 & 1399680 & 1679616
\end{bmatrix}$$

Figure 5.8: The B_5 matrix

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 15 & 5 & 1 \\
0 & 0 & 0 & 5 & 10 & 1 & 146 & 15 & 1 \\
0 & 0 & 1 & 22 & 22 & 1 & 134 & 14 & 1 \\
0 & 1 & 1 & 25 & 28 & 1 & 128 & 13 & 1
\end{bmatrix}$$

Figure 5.9: The H_5 matrix

Bibliography

[1] Don Coppersmith. Finding a small root of a univariate modular equation. In *EUROCRYPT*, pages 155–165, 1996.

[2] Don Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[3] Jean-Sébastien Coron. Finding small roots of bivariate integer polynomial equations revisited. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 492–505. Springer, 2004.

[4] Jean-Sébastien Coron. Finding small roots of bivariate integer polynomial equations: A direct approach. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 379–394. Springer, 2007.

[5] Nick Howgrave-Graham. Finding small roots of univariate modular equations revisited. In Michael Darnell, editor, *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 1997.

[6] A.K Lenstra, H.W Lenstra, and L Lovász. Factoring polynomials with rational coefficients. *Matematische Annalen*, 261(4), 1982.