SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR DEGREE OF MASTER OF SCIENCE AT UNIVERSITY OF CRETE, BY

Dimitrios Megremis
Candidate

Mathematics
Department

TITLE: **LLL ALGORITHM AND APPLICATIONS TO CRYPTOGRAPHY**

APPROVED: **Prof. Theodoulos Garefalakis** ____________________
Committee Chairperson Signature

Prof. Mihalis Kolountzakis ____________________
Faculty Member Signature

Prof. Nikolaos Tzanakis ____________________
Faculty Member Signature

Prof. Mihalis Kolountzakis ____________________
Department Chairperson Signature

DATE: **November 19, 2014**

LLL ALGORITHM AND APPLICATIONS TO CRYPTOGRAPHY

A Thesis
Presented to
The Faculty of the Department of Mathematics
University of Crete

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

By
Dimitrios Megremis
November 2014

ACKNOWLEDGMENTS

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Theodoulos Garefalakis, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Mihalis Kolountzakis and Prof. Nikolaos Tzanakis for their insightful comments and hard questions.

I thank all of my friends in University of Crete but especially Iro Maurogianni, Dimitris Gkiokas and Sotiris Kalpakoglou for the sleepless nights we were studying together and for all the fun we had in the last six years.

Last but not least, I want to thank my family: my parents Giorgos Megremis and Giorgia Kuriakopoulou and finally my brother Grigoris Megremis.

LLL Algorithm and Applications to Cryptography

By

Dimitrios Megremis

ABSTRACT

In this master thesis the security of public key cryptosystem and the security of the Digital Signature Algorithm has been studied. The results have been obtained using the theory of Continued Fractions and Lattice Theory.

All the necessary tools which were used in the study are presented in the introduction. We then review two approaches to the RSA cryptosystem with low private exponent. The first one is due to M. J. Wiener with the use of continued fractions and the second one is a with the use of lattice theory where we apply a powerful technique. Furthermore, a factoring attack has been applied in the modulus of the form $N = p^r q$. Finally, we describe a lattice attack on the discrete logarithm which is based the Digital Signature Algorithm.

TABLE OF CONTENTS

CHAPTER 1

Preliminaries

1.1 Continued Fractions

Let $\theta \neq 0$ real number. We define the following recursion procedure: Set $\theta_0 = \theta$ and $a_0 = [\theta_0]$. For $n \geq 1$, if $\theta_n - 1 = a_n - 1$ the recursion halts, else we set

$$\theta_n = \frac{1}{\theta_{n-1} - a_{n-1}}, \quad a_n = [\theta_n].$$

Thus,

$$\theta = a_0 + \frac{1}{\theta_1} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\theta_2}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{\ddots 1}{a_n + \cfrac{1}{\theta_{n+1}}}}}}. \tag{1.1}$$

This procedure is called expansion of θ into a continued fraction. We refer to the integers $a_0, a_1, ..., a_n$ as partial quotients and to the real numbers $\theta_0, \theta_1, ..., \theta_n$ as complete quotients of the fraction. This expansion can be halted for some $n = N$, but it can be infinite. We will be interested in rational numbers θ. In this case, the sequence of partial quotients is finite and can be computed efficiently, as the following theorem states. Continued fractions are important in many branches of Mathematics, and particularly in the theory of approximation to real numbers by rationals. There are more general types of continued fractions in which the 'numerators' are not all 1's, but we shall not require them.

Proposition 1.1. *Let $\theta = \frac{A}{B}$, where A, B are integers with $\gcd(A, B) = 1$ and $B > 0$. We run the euclidean division $A = Bq_0 + r_0, 0 \leq r_0 < B$. If $r_0 = 0$ then the expansion of θ is* *simply $\theta = q_0$, if not, we apply the euclidean division in the pair (B, r_0) and we set*

$$\begin{aligned}
B &= r_0 q_1 + r_1 \\
r_0 &= r_1 q_2 + r_2 \\
&\cdots \\
r_{n-2} &= r_{n-1} q_n + r_n \\
r_{n-1} &= r_n q_{n+1}
\end{aligned}$$

Then, in the expansion of θ into a continued fraction, the partial quotients $a_0, a_1, ..., a_{n+1}$ coincide with the quotients $q_0, q_1, ..., q_{n+1}$ of the euclidean algorithm, respectively. In addition, for the partial quotients θ_i holds $\theta_i = q_i + \frac{r_i}{r_{i-1}}$ for $1 \leq i \leq n$ and $\theta_{n+1} = q_{n+1}$. This means that the procedure halts in finite number of steps.

Proof. Theorem 161,page 136, in **?**. □

Until now we have seen, given a rational number θ, how one can compute its sequence of partial quotients.The fact that the procedure can be reversed is of great significance. If we are given any sequence of integers $a_0, a_1, ...$, where $a_i > 0$ for $n \geq 1$ then the sequence

$$a_0, a_0 + \frac{1}{a_1}, a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2}}, ..., a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}, ...$$

converges to some real number, let θ, which is expressed as $[a_0, a_1, a_2, ...]$. We will now use the notation

$$\theta = [a_0, a_1, a_2, ...].$$

We will use the auxiliary collection of functions $f_k(x)$. Let $a_0, a_1, a_2, ...$ sequence of integers, where $a_i > 0$ for $i \geq 1$. For every $k \geq 0$ we define

$$f_k(x) = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_k + x}}}, \quad x > 0.$$

Proposition 1.2. *For non-negative integers n we define recursively the integers p_n, q_n as follows:*

$$\begin{aligned}
p_0 &= a_0, \quad p_1 = a_0 a_1 + 1, \quad p_n = a_n p_{n-1} + p_{n-2}, \quad n \geq 2 \\
q_0 &= 1, \quad q_1 = a_1, \quad q_n = a_n q_{n-1} + q_{n-2}, \quad n \geq 2.
\end{aligned} \tag{1.2}$$

Then the following relations hold:

$$f_0(x) = x + p_0, \quad f_n(x) = \frac{p_{n-1}x + p_n}{q_{n-1}x + q_n}, n \geq 1.$$

In particular, $f_n(0) = p_n/q_n$ *for every* $n \geq 0$.

Proof. The proof is done by induction, and is based on the observation that $f_{k+1}(x) = f_k(1/x + a_{k+1})$. □

The fractions p_n/q_n are called convergent of $[a_0, a_1, ...]$.

Theorem 1.3. *Let* θ *be a non-negative real number and let* p/q *be a rational number which satisfies the inequality* $\left|\theta - \frac{p}{q}\right| < \frac{1}{2q^2}$. *Then* p/q *is one of the convergents of* θ.

Proof. Theorem 6.3, page 65, **?**. □

1.2 Lattices

Let the vector space $\mathbb{R}^n$ with dimension n. This space comes with the inner product $< x, y >= \sum_{i=1}^{n} x_i y_i$ and the Euclidean norm $\|x\| = \sqrt{< x, x >} = (\sum_{i=1}^{n} x_i^2)^{1/2}$. Let $A = u_1, ..., u_w \in \mathbb{Z}^n$, n linearly independent vectors with $w \leq n$. A lattice L spanned by $< u_1, ..., u_w >$ is the set of all integer linear combinations of $u_1, ..., u_w$. We say that the lattice is full rank if $w = n$. The set $< u_1, ..., u_w >$ is a basis of the lattice. The set L with the addition of vectors is a group. A very useful way to describe a lattice is to present a basis. It is common to do that with a matrix which has the vectors of the basis as rows. For example, the matrix

$$\begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,w} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,w} \\ \vdots & \vdots & \ddots & \vdots \\ u_{w,1} & u_{w,2} & \cdots & u_{w,w} \end{pmatrix}$$

It is easy to see that a lattice has more than one base. Now let $A = a_1, ..., a_w$ and $B = b_1, ..., b_w$ two distinct bases of the same lattice L, i.e., $L(A) = L(B) = L$. Then $a_i \in L$, $i = 1, ..., n$, so there exist integers m_{ik}, $i, k = 1, ..., n$ such that

$$a_i = \sum_{k=1}^{n} m_{ik} b_k, \quad i = 1, ..., n.$$

For any independent set of vectors $u_1, ..., u_n$ we denote by $u_1^*, ..., u_w^*$ the vectors obtained by applying the Gram-Schmidt process to these vectors. We define the determinant of the lattice L as

$$\det(L) := \prod_{i=1}^{w} \|u_i^*\|$$

If L is full rank lattice then the determinant of L is equal to the determinant of the $w \times w$ matrix of any basis of L.

Fact 1.4. *Let* L *be a lattice spanned by* $< u_1, ..., u_w >$. *Given* $< u_1, ..., u_w >$, *the LLL algorithm* **?** *runs in polynomial time and produces a new basis* $< b_1, ..., b_w >$ *of* L *satisfying:*

(1) $\|b_i^*\|^2 \leq 2\|b_{i+1}^*\|^2$ *for all* $1 \leq i < w$.

(2) *For all* i, *if* $b_i = b_i^* + \sum_{j=1}^{i-1} \mu_j b_j^*$ *then* $|\mu_j| \leq \frac{1}{2}$ *for all* j.

Proof. See **?**. □

Fact 1.5. *Let* L *be a lattice and* $b_1, ..., b_w$ *be an LLL-reduced basis of* L*. Then*

$$\|b_1\| \leq 2^{(w-1)/4} \det(L)^{1/w}$$

Proof. From the Gram-Schmidt procedure we have $b_1 = b_1^*$. We know that:

$$\|b_1\|^2 \leq 2^{i-1} \|b_i^*\|^2$$

for $i = 1, ..., w$. By multiplying all these inequalities we obtain

$$\|b_1\|^{2w} \leq \left(2^{0+1+2+...+(w-1)}\right) \prod_{i=1}^{w} \|b_i^*\|^2$$

Also we know that $\sum_{i=1}^{w-1} i = \frac{w(w-1)}{2}$. Hence,

$$\begin{aligned} \|b_1\|^{2w} &\leq 2^{\frac{w(w-1)}{2}} \prod_{i=1}^{w} \|b_i^*\|^2 \\ &\leq 2^{\frac{w(w-1)}{2}} \left(\prod_{i=1}^{w} \|b_i^*\|\right)^2 \\ &\leq 2^{\frac{w(w-1)}{2}} \det(L)^2 \end{aligned}$$

Passing this inequality to the power $\frac{1}{2w}$ we get

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

as we expected. □

Corollary 1.6. *It holds that* $\frac{w}{2} > \frac{w-1}{4}$*.So, the inequality in Fact* **??** *becomes:*

$$\|b_1\| \leq 2^{\frac{w}{2}} det(L)^{\frac{1}{w}}$$

Definition 1.7. *For a basis* $< u_1, ..., u_w >$ *of a lattice* L*, define*

$$u_{min}^* := min_i \|u_i^*\|$$

Fact 1.8. *Let* L *be a lattice spanned by* $< u_1, ..., u_w >$ *and let* $< b_1, ..., b_w >$ *be the result of applying LLL to the given basis. Suppose* $u_{min}^* \geq 1$*. Then*

$$\|b_2\| \leq 2^{\frac{w}{2}} \det(L)^{\frac{1}{w-1}}.$$

Proof. It is known that u^*_{min} (**?**, Lemma 17.2.11, page 371) is a lower bound on the length of the shortest vector of L. As a consequence, $\|b_1\| \geq u^*_{min}$. Then

$$\det(L) = \prod_i \|b^*_i\| \geq \|b^*_1\|\|b^*_2\|^{w-1}2^{-(w-1)^2/2}$$
$$\geq u^*_{min}\|b^*_2\|^{w-1}2^{-(w-1)^2/2}$$

Hence,

$$\|b^*_2\| \leq 2^{(w-1)^2/2}(\frac{\det(L)}{u^*_{min}})^{\frac{1}{w-1}}$$
$$\leq 2^{(w-1)^2/2}\det(L)^{\frac{1}{w-1}}$$

This leads to

$$\|b_2\|^2 \leq \|b^*_2\|^2 + \frac{1}{4}\|b_1\|^2$$
$$\leq 2^{w-1}\det(L)^{\frac{2}{w-1}} + \frac{1}{4}2^{\frac{2w}{2}}\det(L)^{\frac{2}{w}}$$
$$\leq 2^{w-1}\det(L)^{\frac{2}{w-1}} + 2^{w-2}\det(L)^{\frac{2}{w-1}}$$
$$\leq 2^{w}\det(L)^{\frac{2}{w-1}}(2^{-1} + 2^{-2})$$
$$\leq 2^{w}\det(L)^{\frac{2}{w-1}}$$

Remark 1.9. $\det(L) \geq 1$ *since* $u^*_{min} \geq 1$.

So,

$$\|b^*_2\| \leq 2^{\frac{w}{2}}det(L)^{\frac{1}{w-1}}$$

As we expected. □

1.3 RSA Scheme

The RSA cryptosystem invented by Rivest, Shamir and Adleman in 1978(**?**) is today's most important public-key cryptosystem. Let us denote $N = pq$ an RSA-modulus which is the product of two primes p, q of the same bit-size. Let e be an integer co-prime to Euler's totient function $\phi(N) = (p-1)(q-1)$. The RSA encryption function takes a message m to e^{th} power in the ring $\mathbb{Z}_N$. Let d be the inverse of $e \mod (\phi(N))$, i.e.

$$ed \equiv 1 \mod (\phi(N)). \tag{1.3}$$

Computing the d^{th} power in $\mathbb{Z}_N$ inverts the RSA encryption function. The public key consists of the modulus N and the public exponent e. Respectively, the secret key consists of the modulus N and the private exponent d.

We assume that $N = pq$ is a "good" RSA modulus with $p \approx q \approx \sqrt{N}$, then $N \approx \phi(N)$. In some cases we suppose that $e < \phi(N)$ is very close to N.

1.4 Finding Small Solutions to Polynomial Congruences

In this Chapter we will describe a technique which will apply to solve some cryptanalytic problems. The general approach was introduced by Coppersmith(**?**). We use a simplified version due to Howgrave-Graham(**?**).

Definition 1.10. *Given a polynomial* $h(x,y) = \sum_{i,j} a_{i,j} x^i y^j$, *define* $\|h(x,y)\|^2 := \sum_{i,j} |a_{i,j}^2|$.

1.4.1 Univariate Case

Theorem 1.11. *(HG98) Let* $h(x) \in \mathbb{Z}[x]$ *be a polynomial with degree* $d-1$. *Supposing that:*

- $h(x_0) \equiv 0 \mod p^{rm}$ *for some positive integers* r, m *where* $|x_0| < X$, *and*
- $\|h(xX)\| < \frac{p^{rm}}{\sqrt{d}}$

Then $h(x_0) = 0$.

Proof.

$$\begin{aligned} |h(x_0)| &= \Big|\sum_{i=0}^{d-1} a_i x_0^i\Big| \\ &\leq \sum_{i=0}^{d-1} |a_i x_0^i| \\ &\leq \sum_{i=0}^{d-1} |a_i X^i| \\ &\leq \sqrt{\sum_{i=0}^{d-1} 1^2 \sum_{i=0}^{d-1} |a_i X^i|^2} \quad by \quad the \quad Cauchy - Schwarz \quad inequality \\ &\leq \sqrt{d}\|h(xX)\| \\ &< p^{rm} \end{aligned}$$

□

1.4.2 Bivariate Case

Theorem 1.12. *(HG98) Let* $h(x,y) \in \mathbb{Z}[x,y]$ *be a polynomial which is a sum of at most* w *monomials. Supposing that:*

- $h(x_0, y_0) \equiv 0 \mod e^m$ *for some positive integer* m *where* $|x_0| < X$ *and* $|y_0| < Y$
- $\|h(xX, yY)\| < \frac{e^m}{\sqrt{w}}$

Then $h(x_0, y_0) = 0$.

Proof.

$$\begin{aligned}
|h(x_0, y_0)| &= \Big|\sum a_{i,j} x_0^i y_0^j\Big| \\
&\leq \sum |a_{i,j} x_0^i y_0^j| \\
&\leq \sum |a_{i,j} X^i Y^j| \\
&\leq \sqrt{\sum_{i,j} 1 \sum_{i,j} |a_{i,j} X^i Y^j|^2} \quad by \quad the \quad Cauchy - Schwarz \quad inequality \\
&\leq \sqrt{w} \|h(xX, yY)\| \\
&< e^m
\end{aligned}$$

□

CHAPTER 2

Low Private Exponent.

In order to reduce decryption or signature-generation time it is useful to use a small private exponent. Michael J. Wiener (?) proved that using a small d results in a total break of the cryptosystem.

Wiener had developed a method according to which, with the use of the algorithm of the continued fractions, the secret exponent can be found and finally N can be factored.

2.1 Continued Fraction Algorithm Applied to RSA due to Wiener

Algorithm 2.1. *We are given a public key* (e, N)*.*

(1) *Find the convergents of* e/N*, let* $k_j/d_j = [a_0, a_1, ..., a_j]$*.*

(2) *For each convergent one computes*

$$n' = (d_j e - 1)/k_j.$$

(3) *Form the equation*

$$x^2 - (N - n' + 1)x + N = 0$$

and find the integer solutions these are the two factors p *and* q *of* N*.*

In order for the attack to work, the secret exponent must be less than $N^{1/4}$*.*

Theorem 2.2. *Let* $N = pq$ *with* $q < p < 2q$*. Let* $d < \frac{1}{3}N^{\frac{1}{4}}$*. Given* $< N, e >$ *with* $ed \equiv 1 \mod \phi(N)$*, one can efficiently recover* d*.*

Proof. The proof is based on approximations using continued fractions. Since $ed \equiv 1 \mod \phi(N)$, there exists a k such that $ed - k\phi(N) = 1$. Therefore,

$$\left|\frac{e}{\phi(N)} - \frac{k}{d}\right| = \frac{1}{d\phi(N)}$$

Hence, $\frac{k}{d}$ is very close to $\frac{e}{\phi(N)}$. Even if $\phi(N)$ is not known, N can be used to approximate it. From the hypothesis $\phi(N) = N - p - q + 1$ and the fact that

$$p^2 < 2pq \quad \text{and } q^2 < pq$$

$$\begin{aligned} p^2+q^2 &< 3N \\ p^2+q^2+2pq &< 5N \\ p+q &< (5N)^{1/2} \\ p+q &< 3N^{1/2} \\ p+q-1 &< 3N^{1/2}-1 \end{aligned}$$

we have

$$|N-\phi(N)| < 3N^{1/2}-1$$

Using N in place of $\phi(N)$, we obtain

$$\begin{aligned} \Big|\frac{e}{N}-\frac{k}{d}\Big| &= \Big|\frac{ed-k\phi(N)-kN+k\phi(N)}{Nd}\Big| \\ &= \Big|\frac{1-k(N-\phi(N))}{Nd}\Big| \\ &\leq \Big|\frac{3kN^{1/2}}{Nd}\Big| \\ &\leq \frac{3k}{dN^{1/2}} \end{aligned}$$

Now, $k\phi(N)=ed-1<ed$. Since $ed>k\phi(N)$ and $e<\phi(N)$, we have that $k<d<\frac{1}{3}N^{1/4}$. So we obtain

$$\Big|\frac{e}{N}-\frac{k}{d}\Big| \leq \frac{1}{dN^{1/4}}$$

We observe that, $2d<3d<N^{1/4}$ and then $\frac{1}{2d}>\frac{1}{N^{1/4}}$. Finally, we substitute and have that

$$\Big|\frac{e}{N}-\frac{k}{d}\Big| \leq \frac{3k}{dN^{1/2}} < \frac{1}{d\cdot 2d} = \frac{1}{2d^2}.$$

Theorem **??** implies that k/d is a convergent of e/N. Finally, we can compute the convergents of e/N by using the Proposition **??**. For every convergent k_j/d_j we check if $(d_je-1)/k_j$ is a natural number. If this holds we set $n'=(d_je-1)/k_j$(see **??**) and we check if the quadratic equation $x^2-(N-n'+1)x+N=0$ has integer roots. If it does have integer roots then p and q must be the roots of this equation, otherwise we continue with the next convergent. □

Example 2.3. *We are given the public RSA pair* $(e,N)=(303703,1065023)$*. We will use the Algorithm* **??** *to break the specific RSA cryptosystem.*

The convergents are: $[0,1/3,1/4,2/7,73/256,221/775,957/3356,1178/4131]$

i	n_i	d_i	$n' = (d_i e - 1)/k_i$	$x^2 - (N - n' + 1) + N = 0$
0	*1*	*0*	-	-
1	*1*	*3*	*911108*	*no natural roots*
2	*1*	*4*	*121481*	*no natural roots*
3	*2*	*7*	*1062960*	$p = 1033$,$q = 1031$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

i=0 The convergent is 0*.*

i=1 The convergent is $1/3$*.We check that* n' *is natural. But the equation does not have natural solutions.*

i=2 The convergent is $1/4$*. Again we check that* n' *is natural, but the equation does not have natural solutions.*

i=3 The convergent is $2/7$*. We check that* n' *is natural. So, the equation* $x^2 - (N - n' + 1)x + N = 0$ *becomes*

$$x^2 - 2064x + 1065023 = 0.$$

This equation has roots $p = 1033$ *and* $q = 1031$*. So,* $\phi(N) = 1062960$ *and* $d = 7$*.*

2.2 Attack on RSA scheme due to D. Boneh and G. Durfee

Another approach for cryptanalyzing the Low Private Key RSA, due to D. Boneh and G. Durfee(**?**), is by solving the Small Inverse Problem.

Recall that an RSA public key pair is a pair of integers $< N, e >$, where $N = pq$ is the product of two $n - bit$ primes. The private key is an integer d which satisfies the following equation

$$ed \equiv 1 \mod (\phi(N)) \tag{2.1}$$

where $\phi(N) = N - p - q + 1$. So there exists an integer k such that

$$ed = 1 + k\big(N + 1 - (p + q)\big) \tag{2.2}$$

Writing $s = p + q$ and $A = N + 1$ we have

$$ed = k(A - s) + 1 \tag{2.3}$$

Reducing equation **??** $\mod e$ we know that

$$k(A - s) + 1 \equiv 0 \mod e \tag{2.4}$$

Typically, e is of the same magnitude as N. Supposing that the private exponent d satisfies $d < N^{\delta}$. Wiener's results show that when $\delta < 0.25$ the value of d can be found given e and

N. Boneh and Durfee showed that the same holds for larger values of δ. Specifically, they showed that if $\delta < 0.292$, one can reconstruct d.

Recall that $s = p + q$. From the RSA model that we study $p \approx \sqrt{N}$,$q \approx \sqrt{N}$ and $e \geq \frac{1}{3}N$. Thus,

$$|s| \leq 2\sqrt{3}e^{\frac{1}{2}} = Y$$

In addition let $X > 0$ such that $|k| < X$ (the value of X will be determined later). We set $f(x, y) = x(A - y) + 1$. So, we aim to find (x_0, y_0) satisfying

$$f(x_0, y_0) \equiv 0 \pmod{e} \quad \textit{where} \quad |x_0| < X \quad \textit{and} \quad |y_0| < Y.$$

We intend to apply Theorem **??**. The theorem suggests that we should be looking for a polynomial with small norm that has (x_0, y_0) as a root $\mod e$. To do so, we define the polynomials:

$$g_1 = f(x, y), \quad g_2(x, y) = ex, \quad g_3(x, y) = e$$

We can see that the g_i's have (x_0, y_0) as a root $\mod e$. Thus, every integer linear combination of these polynomials will have (x_0, y_0) as a root $\mod e$. Also, the theorem implies that we do the following transformation:

$$\begin{aligned} x &\mapsto xX \\ y &\mapsto yY \end{aligned} \tag{2.5}$$

Therefore, we are interested in finding a low-norm integer linear combination of $g_i(xX, yY)$, where X, Y as stated above. To do so, we form a lattice spanned by the corresponding coefficient vectors. Thus, we use the collection of these three polynomials ($g_i(xX, yY)$'s) in order to build the basis matrix. In the first column we insert the coefficient of x, in the second the coefficient of the term xy and in the last the constant.

$$\begin{pmatrix} AX & -XY & 1 \\ eX & 0 & 0 \\ 0 & 0 & e \end{pmatrix}$$

By Theorem **??** we must show that the lattice spanned by the polynomials has a sufficiently small determinant. Obviously, the lattice has dimension $w = 3$. The determinant of this lattice is e^2X^2Y. We intend to apply Fact **??** to the shortest vector in the LLL-reduced basis of L. To do so, we must ensure that the norm of b_1 is less than $e/\sqrt{w}$(by Theorem **??**).

Combining this with the Fact **??** we can find the largest value of X satisfying:

$$\begin{aligned}\|b_1\| \leq 2^{\frac{w-1}{4}} \det(L)^{1/w} &\leq \frac{e}{\sqrt{w}} \\ 2^{\frac{1}{2}}(e^2X^2Y)^{\frac{1}{3}} &\leq \frac{e}{\sqrt{3}} \\ X^{\frac{2}{3}}Y^{\frac{1}{3}} &\leq \frac{e^{\frac{1}{3}}}{\sqrt{6}} \\ X^2Y &\leq \frac{e}{6\sqrt{6}}\end{aligned}$$

If we plug in the estimate of $Y = e^{\frac{1}{2}}$, we have

$$\begin{aligned}X^2(2\sqrt{3}e^{\frac{1}{2}}) \leq& \frac{e}{6\sqrt{6}} \\ X^2 \leq& \frac{e^{\frac{1}{2}}}{12\sqrt{18}} \\ X \leq& \frac{e^{\frac{1}{4}}}{\sqrt{12\sqrt{18}}}\end{aligned}$$

So, as long as $X < e^{1/4}$ (ignoring the small constants) the system is vulnerable to this attack. Finally, the above bound proposes the way which we can follow to choose X and Y, in order for the LLL algorithm to provide us with a vector which satisfies the condition of Theorem **??**. By applying the LLL algorithm we take a shortest vector of the form:

$$\big(n_1AX + n_2eX, -n_1XY, n_1 + n_3e\big) = (a, b, c)$$

This vector corresponds to a polynomial. Every coordinate of this vector matches a coefficient of the corresponding monomial (the same order that we used in the formation of the basis matrix). So, the corresponding polynomial is

$$h(x, y) = (n_1A + n_2e)Xx - n_1XYxy + n_1 + n_3e$$

The final step to extract the polynomial which has the desired roots is to apply the reverse of the transformation **??**. In other words,

$$\begin{aligned}x &\mapsto x/X \\ y &\mapsto y/Y\end{aligned} \tag{2.6}$$

Then, we have the polynomial

$$g(x, y) = (n_1A + n_2e)x - n_1xy + n_1 + n_3e.$$

We know that $(x_0, y_0) = (k, s)$ is a root of $g(x, y) \mod e$ and also the fact that its norm is small. Thus, by Theorem **??** we have that $(x_0, y_0) = (k, s)$ is a root of $g(x, y)$ over the integers. So,

$$(n_1A + n_2e)k - n_1ks + n_1 + n_3e = 0$$

By rearranging the terms we have,

$$n_1k(A - s) + n_2ek + n_1 + n_3e = 0 \tag{2.7}$$

$$\begin{aligned} n_1k\phi(N) + n_1 + n_2ek + n_3e = 0 \\ n_1(1 + k\phi(N)) + (n_2k + n_3)e = 0 \\ n_1ed + (n_2k + n_3)e = 0 \\ n_1d + n_2k = -n_3 \end{aligned}$$

Thus, we solve the Diophantine equation (with the use of the Euclidean Algorithm)

$$n_1d + n_2k = -n_3 \tag{2.8}$$

in order to find the values of d and k. We can extract the values of n_1, n_2, n_3 by solving the linear system:

$$\begin{cases} -n_1 = \frac{b}{XY} \\ n_2 = \frac{a - n_1AX}{eX} \\ n_3 = \frac{c - n_1}{e} \end{cases}$$

Remark 2.4. $n_3 = 0$

Proof.

$$\begin{aligned} |n_1| &\leq \frac{2^{1/2}\det(L)^{1/3}}{XY} \\ &\leq 0.267 \cdot e^{1/4} \\ |n_3| &\leq \frac{1}{e}\big(|c| + |n_1|\big) \\ &\leq \frac{1}{e}\big(|2^{1/2}\det(L)^{1/3}| + |n_1|\big) \\ &\leq \frac{1}{e}\big(0.387 \cdot e + 0.267 \cdot e^{1/4}\big) \\ &< 1 \end{aligned}$$

$\Rightarrow n_3 = 0.$ □

So, we have to solve

$$n_1d + n_2k = 0.$$

$$\Rightarrow \frac{k}{d} = -\frac{n_1}{n_2}$$

If we eliminate the $\gcd(n_1, n_2)$ we have:

$$\frac{k}{d} = -\frac{n_1'}{n_2'}$$

We know that $\gcd(k, d) = 1$ so $|k| = |n_1'|$ and $|d| = |n_2'|$.

Finally, from equation **??** we can see that d and k are of the same magnitude, i.e. $d < \lambda N^{1/4}$.

Example 2.5. *Let $N = 1074200609$, $e = 519742771$ and $A = 1074200610$. We take $X = \lfloor \frac{e^{1/4}}{\lfloor\sqrt{12\sqrt{18}}\rfloor} \rfloor = 21$ and $Y = \lfloor 2\sqrt{3}e^{1/2} \rfloor = 78974$. We form the lattice:*

$$\begin{pmatrix} 22558212810 & -1658454 & 1 \\ 10914598191 & 0 & 0 \\ 0 & 0 & 519742771 \end{pmatrix}$$

We use the LLL algorithm and we are given the next reduced basis:

$$\begin{pmatrix} -20648229 & 24876810 & -15 \\ 0 & 0 & 519742771 \\ 439941222 & 346616886 & -209 \end{pmatrix}$$

Now we calculate n_1, n_2, n_3.

$$n_1 = -\frac{24876810}{21 \cdot 78974} = -15$$

$$n_2 = \big(\frac{-20648229}{21} + 15 \cdot (1074200610) = 31\big)$$

$$n_3 = (-15 - (-15))/519742771 = 0$$

We expect that n_3 is equal to zero because otherwise the last coordinate of the vector which is provided by the LLL algorithm would be too large (in terms of the norm) to satisfy the conditions of the Theorem **??**.

So, we have to solve

$$n_1 d + n_2 k = -n_3$$

$$-15d + 31k = 0$$

So one solution we take from the Extended Euclidean Algorithm is $d = 11$ *and* $k = 6$*, because* $\gcd(15, 31) = 1$*. Thus we can go back to the equation* **??** *in order to extract* s*.*

$$s = \frac{n_1 k A + n_2 e k + n_1 + n_3 e}{n_1 k} = 65550 = p + q.$$

Finally, as we established earlier $N = p \cdot q$ *so* $p = 32771$ *and* $q = 32779$*.*

CHAPTER 3

Factoring RSA Modulus of the form $N = p^r q$ for large r.

In recent years moduli of the form$N = p^r q$ have found many applications in cryptography, for example in financial cryptography. Always the security of the system relies on the difficulty of factoring N. We will describe an attack proposed by D. Boneh, G. Durfee, and N. Howgrave-Graham(**?**).

3.1 Lattice-based Factoring

We are given $N = p^r q$. Suppose that in addition, we are also given an integer P that matches p on a few of p's most significant bits. In other words $|P - p| < X$ for some large X. Now our aim is to find p with public information N, r, and P. Let the polynomial $f(x) = (P + x)^r$. Then the point $x_0 = p - P$ satisfies $f(x_0) \equiv 0 \mod p^r$. Hence, we are looking for a root of $f(x) \mod p^r$ satisfying $|x_0| < X$. Unfortunately, the modulus p^r is unknown. Instead, only a multiple of it, N, is known.

We intend to apply Theorem **??**. The theorem suggests that we should be looking for a polynomial with small norm that has x_0 as a root $\mod p^{rm}$. For $k = 0, ..., m$ and any $i \geq 0$ define ($m > 0$ be an integer to be determined later):

$$g_{i,k}(x) := N^{m-k} x^i f(x)^k. \tag{3.1}$$

Observe that x_0 is a root of $g_{i,k}(x) \mod p^{rm}$ for all i and all $k = 0, ..., m$. Thus, every integer linear combination of these polynomials will have x_0 as a root $\mod p^{rm}$. Also, the theorem implies that we do the following transformation:

$$x \mapsto xX \tag{3.2}$$

So we form a lattice spanned by the $g_{i,k}(xX)$ and use the LLL to find a short vector in this lattice. Once we find a short vector $h(xX)$ it will follow from Theorem **??** that x_0 is a root of $g(x) = h(x/X)$ over $\mathbb{Z}$. Then x_0 can be found using standard root finding methods over the reals.

Let L be the lattice spanned by the coefficients vectors of:

(1) $g_{i,k}(xX)$ for $k = 0, ..., m-1$ and $i = 0, ..., r-1$, and

(2) $g_{j,m}(xX)$ for $j = 0, ..., d - mr - 1$.

3.1.1 Case I (r=1)

We will examine the case where $r = 1$, $m = 3$. Thus, the polynomials are:

(1) $g_{0,0}(xX) = N^3$

(2) $g_{0,1}(xX) = N^2 f(xX) = N^2 xX + N^2 P$

(3) $g_{0,2}(xX) = N f(xX)^2 = N x^2 X^2 + 2NxXP + NP^2$

We form the lattice spanned by the corresponding coefficient vectors:

$$\begin{pmatrix} NX^2 & 2NXP & NP^2 \\ 0 & N^2X & N^2P \\ 0 & 0 & N^3 \end{pmatrix}$$

So, the determinant of the matrix is:

$$\det(L) = N^6 X^3$$

We see that the dimension of the lattice is $d = 3$. Thus, by Fact **??** guarantees that LLL algorithm will find a short vector u in L satisfying

$$\|u\|^d \leq 2^{d^2/2} \det(L) = 2^{9/2} N^6 X^3. \tag{3.3}$$

This vector u is the coefficients vector of some polynomial $h(xX)$ satisfying $\|h(xX)\| = \|u\|$. Furthermore, since $h(xX)$ is an integer linear combination of the polynomials $g_{i,k}$, we may write $h(x)$ as an integer linear combination of the $g_{i,k}(x)$. Therefore $h(x_0) \equiv 0 \mod p^{rm}$. To apply Theorem **??** we require that

$$\|h(xX)\| < p^{rm}/\sqrt{d+1}.$$

The factor of $\sqrt{d+1}$ in the denominator has little effect on the subsequent calculations, for reasons of simplicity it is omitted. We want to use the bound on $\|h(xX)\|$ from equation **??** so we need the above inequality raised to d'th power

$$\|h(xX)\|^d < p^{rmd}$$

$$2^{9/2} N^6 X^3 < p^9$$

$$X^3 < \frac{p^9 N^{-6}}{2^{\frac{3}{2}}}$$

Supposing $q < p^c$ for some c. Then $N < p^{r+c} = p^{1+c}$, so we need

$$X^3 < \frac{p^9 p^{-6-6c}}{2^{\frac{9}{2}}}$$

$$X < \frac{p^{1-2c}}{2^{\frac{3}{2}}}$$

Larger values of X allow us to use weaker approximations P, so we wish to find the largest X satisfying the bound.

$$X < \frac{p^{1-2c}}{\frac{3}{2}}. \tag{3.4}$$

When X satisfies the bound **??**, the LLL algorithm will find in L a vector $h(xX)$ satisfying $\|h(xX)\| < p^3/\sqrt{3}$. Then, we do the reverse of the transformation **??**, i.e.,

$$x \mapsto x/X \tag{3.5}$$

This short vector leads to a polynomial $g(x) = h(x/X)$ which is an integer linear combination of the $g_{i,k}(x)$ and thus has x_0 as root $\mod p^3$. But since $\|h(xX)\|$ is bounded, we have by Fact **??** that $g(x_0) = 0$ over the integers, and normal root-finding methods can be used to extract the desired x_0. Given x_0 one can reconstruct p by using the formula $p = P + x_0$.

Example 3.1. *Let* $N = pq = 9025890952536319$*.From the inequality* **??** *and the fact that* $q < p^{1/3}$*, i.e.* $c = 1/3$*, we have that* $X < p^{1/3}$*. We set* $X = 9000$ *and* $P = 1099511620000$*. We form the matrix*

$$\begin{pmatrix} 7310971671554418390000000 & 1786332956969979218254820400000000 & 1091163246320806745349828417118360000000000 \\ 0 & 7332003673836928193093311266278490000 & 8957359152518213926012999423944571012282000 \\ 0 & 0 & 7353096180405309131095908282401588578693521497 59 \end{pmatrix}$$

By LLL algorithm we get

$$\begin{pmatrix} 49266942680274592060585359000000 & -85297500093782077054226784882000 & 3691960129059200901830449338975 9 \\ -38275488368462995089770971042200000 & -70524078764370164321763047983893000 & 347878731107753261414991928577782329 \\ 3538343679319856994375310766036316000000 & 40874289394923588594880117815987387240000 & 472172197079129928019608726910681269024 7 \end{pmatrix}$$

The first vector yields the polynomial

$g(x) = 608233860250303605686239x^2 - 947750001042023078380297 6098x + 369196012905920090183044933897 59.$

This polynomial has $x_0 = 7791$ *as a root.*

So, $P + 7791 = 1099511620000 + 77911 = p$*.*

3.1.2 Case II (r=2)

We will examine the case where $r = 2, m = 3$.We will use these polynomials from the collection of polynomials that D. Boneh and G. Durfee defined:

(1) $g_{0,0}(x) = N^3$

(2) $g_{0,1}(x) = N^3x$

(3) $g_{1,0}(x) = N^2P^2 + 2PN^2x + N^2x$

(4) $g_{1,1}(x) = N^2P^2x + 2PN^2x^2 + N^2x^3$

(5) $g_{2,0}(x) = NP^4 + 4NP^3x + 6NP^2x^2 + 4NPx^3 + Nx^4$

(6) $g_{2,1}(x) = NP^4x + 4NP^3x^2 + 6NP^2x^3 + 4NPx^4 + Nx^5$

Now we work in a similar way as in the previous case. First, we form the lattice spanned by the corresponding coefficient vectors:

$$\begin{pmatrix} N^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & N^3X & 0 & 0 & 0 & 0 \\ N^2P^2 & 2PXN^2 & X^2N^2 & 0 & 0 & 0 \\ 0 & N^2P^2X & 2PX^2N^2 & N^2X^3 & 0 & 0 \\ NP^4 & 4NP^3X & 6NP^2X^2 & 4NPX^3 & NX^4 & 0 \\ 0 & NP^4X & 4NP^3X^2 & 6NP^2X^3 & 4NPX^4 & NX^5 \end{pmatrix}$$

The determinant of the lattice is

$$\det(L) = N^{12}X^{15}.$$

Fact **??** guarantees that LLL algorithm will find a short vector u in L satisfying

$$\|u\|^d \leq 2^{d^2/2}\det(L) = 2^{36/2}N^{12}X^{15}. \tag{3.6}$$

As in the previous case, suppose that we have a short vector $h(xX)$. In order the hypothesis of the Theorem **??** to hold we need:

$$\|h(xX)\|^d < p^{rmd}$$

$$2^{\frac{36}{2}}N^{12}X^{15} < p^{36}$$

$$X^{15} < \frac{p^{36}N^{-12}}{2^{\frac{18}{15}}}$$

Supposing $q < p^c$ for some c. Then $N < p^{r+c} = p^{2+c}$, so we need

$$X^{15} < \frac{p^{36}p^{-24-24c}}{2 \cdot 2^{\frac{1}{5}}}$$

$$X < \frac{p^{\frac{4-4c}{5}}}{2 \cdot 2^{\frac{1}{5}}}$$

Larger values of X allow us to use weaker approximations P, so we wish to find the largest X satisfying the bound.

$$< \frac{p^{\frac{4-4c}{5}}}{2 \cdot 2^{\frac{1}{5}}}. \tag{3.7}$$

When X satisfies the bound **??**, the LLL algorithm will find in L a vector $h(xX)$ satisfying $\|h(xX)\| < p^6/\sqrt{6+1}$. Then, we do the reverse of the transformation **??**, i.e.,

$$x \mapsto x/X \tag{3.8}$$

This short vector leads to a polynomial $g(x) = h(x/X)$ which is an integer linear combination of the $g_{i,k}(x)$ and thus has x_0 as root $\mod p^3$. But since $\|h(xX)\|$ is bounded, we have by Fact **??** that $g(x_0) = 0$ over the integers, and normal root-finding methods can be used to extract the desired x_0. Given x_0 one can reconstruct p by using the formula $p = P + x_0$.

Example 3.2. *Let $N = p^2q = 39329557$.From the inequality* **??** *and the fact that $q < p^{4/5}$, i.e. $c = 4/5$, we have that $X < p^{4/5}$. We set $X = 64$ and $P = 1000$. (The numbers are too large, so some of the calculations are omitted.)*

- *We form the matrix as described above.*
- *We use the LLL algorithm.*
- *We get the first vector.*

The first vector corresponds to the polynomial $h(x) = 1152873274378970726 4x^5 - 296439930887990149 12x^4 + 7043982629123129344x^3 - 4210387930503905280x^2 + 2045725019284554835 2x - 8397249715018075758$.

So the $g(x) = h(x/X) = 10736969061x^5 - 1766919677782x^4 + 26870661274426x^3 - 1027926740845680x^2 + 319644534263211693x - 8397249715018075758$.

This polynomial has $x_0 = 557$ as a root.

So, $P + 31 = 1000 + 31 = p$.

CHAPTER 4

Security of the Digital Signature Algorithm

4.1 The Digital Signature Algorithm

DSA bases its security on the presumed intractability of the discrete logarithm problem in the multiplicative group of finite fields , and in prime order subgroups. So, we choose the following quantities:

- a prime p of size between 512 and 1024 bits in increments of 64
- a prime q of size 160 bits, s.t. $q|p-1$
- a hash function h mapping messages to the subgroup of order q
- a secret integer α in the subgroup of order q.

The parameters determine the finite field $\mathbb{F}_p$, and its unique subgroup G of order q. Let g be the generator of this subgroup, i.e. $G =< g >$.

Algorithm 4.1. *To sign a message m, Alice performs the following steps:*

(1) *Choose $k \in \{1, ..., q\}$ uniformly at random.*

(2) *Compute $r = (g^k \mod p) \mod q$.*

(3) *Compute $s = k^{-1}(h(m) + \alpha \cdot r) \pmod q$.*

(4) *Send (r, s) as the digital signature of the message m.*

In this procedure the key α is referred to as the secret key, intended to be chosen only once, and k is the ephemeral key chosen differently for each message. The assumption here is that the only way to break this signing algorithm is to recover either the secret key α, or the ephemeral key k.

4.2 Attack on DSA

In this section we describe an attack on DSA due to Ian F.Blake and Theodoulos Garefalakis(**?**). Their approach is not to recover the secret key by solving the related discrete logarithm directly. Instead their approach uses the form of the equation in step **??** of the algorithm.

Note here that step **??** does not reveal any information about α or k. However, we will use this equation together with the assumption that α and k are of *relatively* small size, to break the system. By rearranging terms in equation **??**, we have

$$\begin{aligned} s \cdot k - h(m) - \alpha \cdot r &\equiv 0 \pmod q \\ k + \big(- rs^{-1}\big) \cdot \alpha + \Big(- h(m)s^{-1}\Big) &\equiv 0 \pmod q \end{aligned}$$

Hence, the pair (α, k) satisfies a modular equation of the form

$$f(x,y) \equiv 0 \pmod q, \tag{4.1}$$

where, in our case,

$$f(x,y) = y + Ax + B$$

with

$$A = -rs^{-1} \mod q \quad \textit{and} \quad B = -h(m)s^{-1} \mod q.$$

Now we assume that the solution we are looking for is small, i.e., $|\alpha| < X$, and $|k| < Y$, for some bounds X, Y that we specify later. Given a modular equation such as equation **??**, which is assumed to have small solution, Theorem **??** (where $e := q$ and $m := t$) show us the cases in which this small modular solution is a solution to the integer equation. Since our equation **??** has three monomials of first degree at the most, define the polynomials (similar to $x - shifts$ in Small Inverse Problem)

$$g_{0,0}(x,y) = q \ , \ g_{0,1}(x,y) = f(x,y) = y + Ax + B \text{ and } g_{1,0}(x,y) = qx.$$

These polynomials evaluated xX and yY lead to the basis matrix

$$\begin{pmatrix} AX & Y & B \\ qX & 0 & 0 \\ 0 & 0 & q \end{pmatrix}$$

The matrix has determinant q^2XY. By combining the Remark **??** and Theorem **??** (where $e := q$ and $m := t$) a solution of the integer equation is guaranteed if

$$2^{w/2} \det(L)^{1/w} \leq \frac{q^t}{\sqrt{w}}$$

where w is the dimension of the matrix and t the power of the modulus q in the modular equation **??**, i.e. 3 and 1 respectively. So the inequality becomes

$$2^{3/2}(q^2XY)^{1/3} \leq \frac{q}{\sqrt{3}}$$

If we let $X = q^\kappa$ and $Y = q^\lambda$

$$\begin{aligned}
2^{\frac{3}{2}q^{\frac{2+\kappa+\lambda}{3}}} &\le \frac{q}{\sqrt{3}} \\
2^{9/2}q^{2+\kappa+\lambda} &\le \frac{q^3}{(\sqrt{3})^3} \\
q^{\kappa+\lambda-1} &\le \frac{1}{2^{\frac{9}{2}}3\sqrt{3}} \\
(\kappa+\lambda-1)\log_2 q &\le -\log_2{(2^{\frac{9}{2}}3\sqrt{3})} \\
(\kappa+\lambda-1)\log_2 q &\le -(\frac{9}{2}\log_2 2 + \log_2 3 + \frac{1}{2}\log_2 3) \\
(\kappa+\lambda-1)\log_2 q &\le -(\frac{9}{2} + \frac{3\cdot 3}{2\cdot 3}\log_2 3) \\
(\kappa+\lambda-1)\log_2 q &\le -4.5(1 + \frac{\log_2 3}{3}) \\
\kappa+\lambda &\le 1 - \frac{4.5}{\log_2 q}(1 + \frac{\log_2 3}{3})
\end{aligned}$$

$$\kappa+\lambda \le 1 - \frac{6.877}{\log_2 q} \tag{4.2}$$

If condition **??** is satisfied, then the shortest vector of the reduced basis is guaranteed to yield a polynomial $H_1(x,y)$ with the desired root over the integers. However, in order to obtain this solution, we need one more 'small' equation. For this purpose, we use the second shortest vector. If the bound

$$\|b_2\| < \frac{q}{\sqrt{3}}$$

holds for the size of the second shortest vector, we obtain a second polynomial $H_2(x,y)$. It is important to note that $H_1(x,y)$ and $H_2(x,y)$ are linear in x and y, and are *linearly independent*. Thus solving the linear system we would obtain the values that we desired. We now proceed to show that the second shortest vector is indeed short enough. From Fact **??** we know that

$$\|b_2\|^2 \le \|b_2^*\| + \frac{1}{4}\|b_1\|^2. \tag{4.3}$$

We need to give an upper bound for $\|b_2^*\|$. Again form Fact **??** we have

$$\det(L) \ge \|b_1\|\|b_2^*\|^2 2^{-\frac{4}{2}}$$

By rearranging the terms we have the bound

$$\|b_2^*\|^2 \le 4\frac{\det(L)}{\|b_1\|},$$

which is equivalent to

$$\|b_2^*\|^2 \leq 4\frac{q^{2+\kappa+\lambda}}{\|b_1\|}$$

In order for the second vector of the LLL-reduced basis to also meet the bound of Theorem **??** we need $\|b_2^*\| < q/\sqrt{3}$ and in order to satisfy this bound, from the above estimate, we have to choose

$$\|b_1\| \geq 16q^{\kappa+\lambda}. \tag{4.4}$$

With two linearly independent equations, the solution can be obtained.

Example 4.2. *Let* $p = 93450983094850938450983409623$ *and* $q = 46725491547425469225491704811$ *and* $g = 18$ *and* $r = 11019960576$*. We also know that* $h(m) = 20$*. We see by inequality* **??** *that* $\kappa + \lambda < 60/67$*. So we choose* $\kappa = 45/67$ *and* $\lambda = 14/67$*. Thus,* $X = \lfloor q^{\kappa} \rfloor = 18016503716681878979$ *and* $Y = \lfloor q^{\lambda} \rfloor = 978707$*. We can check that* $\|b_1\| > 16XY = 16q^{\kappa+\lambda}$*. We form the matrix:*

$$\begin{pmatrix} 25614990270667240277684768180830119674771787987 & 978707 & 26820971257104291911305105461 \\ 841829992127978687003746550688630782125494067969 & 0 & 0 \\ 0 & 0 & 46725491547425469225491704811 \end{pmatrix}$$

Using the LLL algorithm we get:

$$\begin{pmatrix} -49208092849844050731320447 3 & -72413904314090222160111959 2 & 5919148780102310507466 \\ -64005086988663054636906901 1 & 73243863102801212842161662 7 & -5986990026865660771162 \\ 0 & 0 & 46725491547425469225491704811 \end{pmatrix}$$

The first and the second vector yield two linear polynomials. We transform them according to the following:

$$x \mapsto x/X$$

$$y \mapsto y/Y$$

In this way, we extract two new polynomials that have (κ, α) *as root.*

$f(x) = -27312787x - 7398935975127410 16056y + 5919148780102310507466$ *and*

$h(x) = -35525809x + 748373753358269766561y - 5986990026865660771162$

So by solving the linear system $f(x) = 0$ *and* $h(x) = 0$ *we can retrieve* k *and* α*. We can check that* $k = 8$ *and* $\alpha = 14$*.*

We can test that the bound **??** *is satisfied.*