NTNU – Trondheim
Norwegian University of
Science and Technology

The Number Field Sieve

Elin Margrete Trondsen

Master of Science in Mathematics
Submission date: November 2012
Supervisor: Kristian Gjøsteen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Acknowledgements

I want to thank associate professor Kristian Gjøsteen for supervising me. You answered all my cloudy and fuzzy questions illuminating, and that has made all the difference for this paper.

I thank Haidar Nuri for teaching me that Matteland is not just a place for studies, but also for fun and social activities. Always remember H that the night is dark and full of terror. A big thanks to Torkil Utvik Stai and Anders Smedstuen Lund for always taking the time to answer my silly questions. And I am really happy for all the coffee hours we have spend together!

Thanks to Laila Solli for taking care of me at the end. Writing a paper is hard, but being served dinner every day makes it a tiny bit easier.

And to the rest of you who remain unnamed, I am forever grateful for the support and love.

Abstract

We present two algorithms for splitting a general composite number, the quadratic sieve algorithm (QS) and the general number field sieve algorithm (NFS). The former is the method of choice for integers between 50 and 110 digits, and the latter beyond. They share a common strategy, but the NFS is far more sophisticated. We therefore present the QS as a preparation for the NFS.

We also give two algorithms for the discrete logarithm problem in fields $\mathbb{F}_q$, q a prime, the index-calculus method (ICM) and the number field sieve for the discrete logarithm problem (NFS-dlog). They have crossover point at 66-digit primes. The only limitation made was restricting to the prime field case. The NFS-dlog uses ideas from both the NFS and the ICM.

In addition we show that the running times of both the NFS and the NFS-dlog are given as $L_k(1/3; (64/9)^{1/3})$, where k is the number to split and the characteristic of the prime field respectively. The subexponential function L is given as

$$L_k(v; c) = e^{(c+o(1))(\ln k)^v (\ln \ln k)^{1-v}}, \text{ for } k \to \infty$$

We also study the algebraic background for the two main algorithms.

Sammendrag

Vi presenterer to algoritmer for å splitte et generelt sammensatt tall, den kvadratiske solden (QS) og den generelle tallkroppssolden (NFS). Den første solden er førstevalget for tall bestående av mellom 50 og 110 siffer, og den andre for større tall. De deler strategi, men NFS er mer avansert. Vi presenterer derfor QS som en forberedelse til NFS.

Vi presenterer også to algoritmer for å finne diskrete logaritmer i kropper $\mathbb{F}_q$, q et primtall. De kalles indeks-kalkulus metoden (ICM) og tallkroppssolden for diskrete logaritmer (NFS-dlog) og har 66 siffers primtall som veiskille. Vi begrenset oss til å studere primkroppen. NFS-dlog bruker ideer fra både NFS og ICM.

I tillegg viser vi at både NFS og NFS-dlog har $L_k(1/3; (64/9)^{1/3})$ som et godt tidsestimat, hvor k er henholdsvis tallet å splitte og karakteristikken til primkroppen. Den subeksponensielle funksjonen L er gitt ved

$$L_k(v; c) = e^{(c+o(1))(\ln k)^v (\ln \ln k)^{1-v}}, \text{ for } k \to \infty$$

Vi studerer også den algebraiske bakgrunnen til algoritmene.

CONTENTS

CHAPTER

ONE

INTRODUCTION

In this thesis we are concerned with algorithms that solve two basic problems in computational number theory: splitting integers into its prime factors and finding discrete logarithms in fields $\mathbb{F}_q$, q a prime.

They are both simple mathematical concepts, but the problem is the sizes. Being doable in principle is not enough when a number gets too big. Since just trying all options it way too time-consuming, a more sophisticated approach is needed. We will study some of these methods.

Although the integer factorization and the discrete logarithm problem are different, they have a few common features. They are both difficult problems, but some efficient algorithms are known for both. And if we know an algorithm for one problem, we can adapt it to the other. Lastly, various cryptographic systems have been constructed based on the difficulty of both problems.

The problems we look at are for instance used as security in the RSA and the Diffie-Hellman key exchange protocol. The RSA is based upon the difficulty of splitting a product of two primes and the other the discrete logarithm problem.

The RSA cryptosystem was introduced in 1978 by Ronald Rivest, Adi Shamir and Leonard Adleman. It is a public-key cryptosystem, meaning an encryption key is made public allowing anyone to encrypt a message. The decryption key is kept secret. B wants to send a message m to A. A makes the system by choosing two primes p and q and calculates $n = pq$. Then she picks an encryption exponent e, such that $\gcd(e, (p-1)(q-1)) = 1$. The public key is (n, e). B computes $m^e \pmod{n}$ and sends it to A. A can easily calculate the decryption key d such that $ed \equiv 1 \pmod{(p-1)(q-1)}$, and hence find $(m^e)^d \equiv m \pmod{n}$.

The Diffie-Hellman key exchange protocol can be described as follows. Two users, A and B, wants to decide on a secret cryptographic key and they have only an insecure communication channel to work in. They agree on a group G and an element $g \in G$. Now A chooses a random integer x, computes g^x and sends this element to B, while keeping x secret. B does the same, chooses y, computes g^y and sends it to A while keeping y secret. They can now both compute the secret key by $(g^y)^x$ and $(g^x)^y$, while an intruder will only have g^y and g^x. It is conjectured that it is impossible to compute the secret key from this information, so the intruder's only hope is to calculate one of the discrete logarithms.

In the 70s, before one began to take factorization seriously, the numbers considered hard were 20-digit numbers. This limit was expanded to 50-digit numbers by the continued fraction factorization method and further, in 1981, Carl Pomerance's quadratic sieve factoring algorithm increased the limit, hitting a record in 1994 with a 129-digit RSA challenge number [9]. And then John Pollard's number field sieve arrived and split a 130-digit number in about 15 % of the time the quadratic sieve would have used. It is still the most efficient algorithm for general numbers, with a latest record in 2009 splitting the 232-digit RSA challenge number [5].

The computation of discrete logarithms in prime fields is not as developed as the splitting of integers of approximately the same size as the order of the field, even though the index-calculus method was already described in 1922 by Maurice Kraitchik. After the arrival of the number field sieve algorithm, Daniel M. Gordon and others modified it to fit the discrete logarithm problem and it had a recent record in 2007 when Thorsten Kleinjung computed a discrete logarithm modulo a 160-digit prime [12].

The goal of this thesis is to describe the two number field sieves, explain relevant background information and analyse the algorithms. Beyond their practical values, the number field sieves use a lot of mathematical concepts and they are therefore also academically interesting.

The algorithms for splitting numbers builds upon the following idea. If we have $X^2 \equiv Y^2 \pmod{n}$ and $X \not\equiv \pm Y \pmod{n}$, then $\gcd(X \pm Y, n)$ is a nontrivial factor of n, with probability at least $1/2$. So if we find the numbers X and Y, we can easily factor n.

As we will see, the number field sieve for the discrete logarithm problem has the same strategy as the splitting algorithms, but has a different purpose with its action. Instead of searching for squared elements, it seeks lth powers, where l divides the order of the field.

We have organized the thesis as follows. We begin the next chapter with a description of the smooth numbers and the sieving process. Then we present the quadratic sieve and the index-calculus method together with their analysis.

The main algorithms and their analysis depends on many different parts of number theory and we cannot hope to present a complete exposition, but we will outline some of the relevant algebraic number theory in Chapter 3 . Then Chapter 4 contains the presentation of the number field sieve for integer factorization and Chapter 5 the number field sieve for the discrete logarithm problem. In Chapter 6 we analyse the two last algorithms and in Chapter 7 we make some concluding remarks.

CHAPTER

TWO

COMPUTATIONAL BACKGROUND

We introduce the sieving method. After a thorough description, we demonstrate it in the presentation of the quadratic sieve in Section 2.2. The quadratic sieve will also be analysed and during the analysation we present two important tools, Theorem 1 and Theorem 2. After a small demonstration of the quadratic sieve with a trivial splitting, we present the index-calculus method in Section 2.3.

2.1 The Method of Sieves

The sieving method is a process where members on a list get cut out due to some premade rules. There are different ways to use the method and we will use it to detect smooth polynomial values, to be defined.

The earliest use of a sieve is Eratosthenes sieve. It finds all the primes in a list up to a bound. We describe the sieve shortly to grasp the sieving idea.

Let b be a composite number and make the list $N = \{2, 3, 4, \ldots, b\}$. The first prime element in N is $p_1 = 2$. Start at p_1 and go through the list by renaming every other number 1, that is, remove all multiples of 2. The next nontrivial element in N is $p_2 = 3$. Start at p_2 and go through the list by renaming every third number 1, that is, delete all multiples of 3. The numbers already renamed 1 are unchanged. This process is repeated until the next nontrivial element, say p_i, have a size such that $p_i^2 > b$. Then every nontrivial element in the list is a prime and the list will look like $\widehat{N} = \{2, 3, 1, 5, 1, 7, 1, 1, 1, 11, \ldots, 1\}$. The bound could of course be a prime, then the last element in $\widehat{N}$ would be b.

Sieving to find B-smooth Numbers

We want to find the smooth integers in N. An integer is defined to be B-smooth if all of its prime divisors are less than B. Collect all primes $\leq B$ in $\mathcal{B}$. We call it the factor base and define $\pi(B) = |\mathcal{B}|$.

We begin with the first prime $p_1 \in \mathcal{B}$ and find the first integer in the list having p_1 as a factor. Now divide through N as described in the Eratosthenes sieve, but instead of replacing the divided terms with 1, we store the divisor. The next element to sieve by is p_1^2, then p_1^3, until $p_1^i \geq b$ for some i. After this is repeated for all primes in $\mathcal{B}$, we multiply all the biggest divisors together for each element in N. If we get the original number back, it is B-smooth.

We demonstrate the sieve with a small example. Let $N = \{2, 3, \ldots, 12\}$ and $B = 3$, so that $\mathcal{B} = \{2, 3\}$. We sieve N by $\{2, 2^2, 2^3, 3, 3^2\}$

	2	3	4	5	6	7	8	9	10	11	12
2	**2**		2		**2**		2		**2**		2
2^2			$\mathbf{2^2}$				2^2				$\mathbf{2^2}$
2^3							$\mathbf{2^3}$				
3		**3**			**3**			3			**3**
3^2								$\mathbf{3^2}$			
	$\underline{2}$	$\underline{3}$	$\underline{4}$	-	$\underline{6}$	-	$\underline{8}$	$\underline{9}$	2	-	$\underline{12}$

The set $\hat{N} = \{2, 3, 4, 6, 8, 9, 12\}$ contains the 3-smooth numbers in N.

Sieving a Polynomial to find B-smooth Numbers

The sieve described next will detect the B-smooth values of $f(x) = c_d x^d + \ldots + c_0$ over $N = [1, b]$. We begin by making the list

$$F = \{f(1), f(2), \ldots, f(b)\}$$

The following procedure should be performed for all primes in $\mathcal{B}$.

Let p_1 be the first prime in $\mathcal{B}$ and locate the first element i in N such that

$$f(i) \equiv 0 \pmod{p_1}$$

Naturally $f(i + kp_1) \equiv 0 \pmod{p_1}$, $\forall k \in \mathbb{N}$. Test if there is an $i' \in [i, i + p_1]$ such that $f(i') \equiv 0 \pmod{p_1}$.

We find all elements in $[1, b]$ that divides f mod p_1 by adding p_1 to the root i of f mod p_1 and they will be located in the places $f(i) + kp_1$ in F. Divide $f(i + kp_1)$ by the highest power of p_1 dividing it. The same for i' if it exists.

After repeating for all primes in $\mathcal{B}$, the B-smooth polynomial values can be located in F as the ± 1's.

Different ways of storing the divisors will save time. Also, if we do the sieving with the logarithms instead we can change from divison to subtraction, which is a cheaper operation. In further use of the technique we do not always bother with higher terms of the primes or with the small and time-consuming primes. Note that we are not only interested in the polynomial values being B-smooth, but also the x's that leads to the smooth $f(x)$ values. Why will be clear in the description of the quadratic sieve given next.

2.2 The Quadratic Sieve

This section contains a presentation of the quadratic sieve factoring algorithm (QS) followed by an heuristic time analysis and a small example. It was the best general factoring algorithm one had in the 80s and early 90s, and it is still the method of choice for integers between 50 and 110 digits. We will mainly follow the presentation in [11].

Algorithm

We are given a composite number n and want to split it. To achieve the splitting we locate two integers X and Y satisfying $X^2 \equiv Y^2 \pmod{n}$. Then hopefully $\gcd(X \pm Y, n)$ is a nontrivial factor of n.

The quadratic sieve has three main steps. First we make a factor base, then we use sieving to locate integers of the correct form and lastly linear algebra is used to reveal X and Y.

We begin by calculating the limit B and finding all the primes $p \leq B$. A good limit is estimated in the next section. The factor base M will consist of $p_1 = 2$ and the primes $p \leq B$ for which $\left(\frac{n}{p}\right) = 1$. Now n is a quadratic residue modulo p, $\forall p \in M$, and there exists some $a \in [1, p-1]$ such that $a^2 \equiv n \pmod{p}$. Notice that $|M| \approx \frac{1}{2}\pi(B)$. We have saved time for the sieving by removing the primes that never give integers of the right form. What are the integers of the correct form?

We are looking for two integers X and Y satisfying $X^2 \equiv Y^2 \pmod{n}$. If we let one side be a square and search for its B-smooth residues, a combination might provide the other square. Formally, we want $f(x_i) = x_i^2 - n$ to be B-smooth for some set $\{x_i\}_i$, meaning each x_i should satisfy

$$f(x_i) = \prod_{j=1}^{|M|} p_{i,j}^{e_{i,j}}, \text{ where } p_{i,j} \in M,\ e_{i,j} \in \mathbb{N},\ \forall i, j$$

To achieve this we sieve a list of $f(x)$ values and keep the x's that return $f(x)$ as B-smooth. The x's that have highest possibility of success are the ones keeping $f(x)$ small. Let therefore $\lceil\sqrt{n}\rceil \leq x \leq \lceil\sqrt{n}\rceil + U$ and perform the sieving process described in Section 2.1. An estimate for a good bound U is given in the next section.

The sieving is completed when we have collected more than $|M|$ B-smooth $f(x_i)$ values.

Now associate an exponent vector $\mathbf{v}_i(x_i^2 - n) = (e_{i,1}, e_{i,2}, \ldots, e_{i,|M|})$ to each $f(x_i)$ and collect the vectors in a matrix. Since we have more vectors than there are primes in M, a linearly dependent subset is possible to find. Actually, when a combination of different vectors modulo 2 equals the zero vector, we have found the squared property we are searching for. To see this, assume linear algebra over the matrix has returned a subset $S = \{\mathbf{v}(x_i^2 - n)\}_{i=1}^{k}$ such that

$$\mathbf{v}_1 + \mathbf{v}_2 + \ldots + \mathbf{v}_k \equiv \mathbf{0} \pmod 2 \tag{2.1}$$

This is equivalent to

$$\sum_{i=1}^{k} e_{i,j} = 2z_j, \text{ where } z_j \in \mathbb{Z} \text{ for } 1 \le j \le |M|$$

For every vector in (2.1), locate the associated $f(x_i)$ value. In the sieving step we found the factorization of $f(x_i), \forall x_i$, and can now compute

$$\begin{align} Y^2 &= f(x_1)f(x_2)\ldots f(x_k) \tag{2.2} \\ &= \prod_{j=1}^{|M|} p_j^{2z_j} \tag{2.3} \end{align}$$

In other words, we have found the squared property. What we really seek is the square root of (2.2)

$$Y = \sqrt{f(x_1)f(x_2)\ldots f(x_k)}$$

This is a huge number and is almost impossible to compute, so we use the easier (2.3) instead and get

$$Y \equiv \prod_{j=i}^{|M|} p_j^{z_j} \pmod n$$

We are working modulo n all the way, but await the reduction until this very last step to maintain the desired properties outlined above.

The other square X^2 was squared from the beginning and it looks like

$$X^2 = \prod_{i=1}^{k} x_i^2 - n$$

We easily find X to be

$$X \equiv \prod_{i=1}^{k} x_i \pmod n$$

Lastly we calculate $\gcd(X \pm Y, n) = D$. If D is nontrivial, we have managed to split n. If not, go back to the linear algebra step and locate a new subset satisfying (2.1) and repeat the remaining part of the algorithm. If no such subset is found we need to extend the bounds B and U and locate more relations.

Before we move on to the analysis of the algorithm, we will briefly mention a possible improvement. There are different advancements to make, for instance sieving x's centered at $\sqrt{n}$ or carry out the sieve with the logarithms. We will look at Montgomery's multiple polynomial variation of the quadratic sieve method [11].

The polynomial values $f(x) = x^2 - n$ will increase rapidly as x runs away from $\sqrt{n}$. A consequence is that the smooth numbers will occur at a decreasing rate, because small numbers are more likely to be smooth than greater numbers. If we now replace the polynomial with a well-chosen family of polynomials, then perhaps the numbers will stay smaller overall.

We require the new polynomial to sieve by the same factor base M since more primes would be time-consuming. Also, the shape must be preserved to keep the polynomial a square modulo n. Essentially we replace x with a linear polynomial satisfying some demands. Montgomery chose

$$y(x) = (ax + b)^2 - n = a^2x^2 + 2abx + b^2 - n \; a, b \in \mathbb{Z}$$

The a should be chosen such that it is a square times a B-smooth integer and b such that $b^2 - ac = n$ for some $c \in \mathbb{Z}$. Then $y(x) = a(ax^2 + 2bx + c)$ with $f(x) = ax^2 + 2bx + c$ minimized, and we get the relation $(ax + b)^2 \equiv af(x) \pmod{n}$. If a, b and c are chosen optimally, $f(x)$ will stay small during the sieving.

Let $-U_M < x < U_M$ be the sieving interval, and choose $a \approx \sqrt{2n}/U_M$ and $b \approx \frac{1}{2}a$. Then $f(x)$ is bounded by $\frac{U_M\sqrt{n}}{\sqrt{2}}$. The $x^2 - n$ would have been bounded by $2U_M\sqrt{n}$ if we had sieved by $\sqrt{n} - U_M < x < \sqrt{n} + U_M$ in the original quadratic sieve, and then Montgomery's method would have saved a factor of $2\sqrt{2}$. Since our interval is $[\sqrt{n}, \sqrt{n} + U]$ with $U = 2U_M$ in the original sieve, even more time is saved. It is said in [11] that using multiple polyomials speeds up the quadratic sieve by a factor of $\frac{1}{2}\sqrt{\ln n \ln \ln n}$. This is why the multiple polynomial variation is preferred over the standard polynomial when the method is implemented.

Nevertheless we will analyse the original quadratic sieve algorithm.

Analysis

In this section we give estimates on the bounds B and U, find the probability that a random integer $x^2 - n$ is B-smooth and give an heuristic estimate of the complexity.

We want to know where to place the limit B so that the frequency of B-smooth numbers is optimal. If the bound is small the matrix will be easy to work with, but the number of smooth integers will be rare and the sieving step will dominate the algorithm. If the limit is big we will find the smooth numbers quickly, but we need a lot of them to locate the linearly dependent subset and hence the matrix will be very large and control the algorithm. The two forces must be balanced. Before we can calculate where the optimal limit should be placed, we need some other estimates.

We begin with a definition.

Definition 1. $\Psi(n, B) = |\{x | x \textit{ is } B\textit{-smooth and } \leq n\}|$

All the B-smooth integers can be written uniquely as $\prod_{i=1}^{\pi(B)} p_i^{e_i}$ for the primes $p_i \leq B$ and some integers e_i, and we found their factorization during the sieving.

Now, the likelihood of $x^2 - n$ being B-smooth in an interesting range is an unsolved and hard problem in analytic number theory, and many people have approached the problem with a lot of different estimates. The following from [6] gives an heuristic estimat that is good enough for our use. Let $u = \frac{\ln n}{\ln B}$.

Theorem 1. *The probability that a number less than n is B-smooth is u^{-u}.*

Proof. A brief overview of a heuristic proof follows.

Assume polynomial values are just as likely to be smooth as random numbers of same size. Every smooth integer less than n has a corresponding $\pi(B)$-tuple of e_i's. Hence we think of $\Psi(n, B)$ as the number of integer solutions e_i to the inequality $\sum_{i=1}^{\pi(B)} e_i \ln p_i \leq \ln n$.

If we now ignore the smallest primes, the primes less than B have about equal logarithms as B and we can exchange $\ln p_i$ with $\ln B$ for all p_i's. If we also replace $\pi(B)$ with B, then $\Psi(n, B)$ will be roughly equal to the number of B-tuple integer solutions to the inequality

$$\sum_{i=1}^{B} e_i \leq u \tag{2.4}$$

The total number of B-tuples e_i in (2.4) is precisely the binomial coefficient $\binom{[u]+B}{B}$, with [u] being the greatest integer function.

The probability that a nonnegative number smaller than n is B-smooth is

$$\Pr[\text{random integer} \leq n \text{ is smooth}] = \frac{\Psi(n, B)}{n} \tag{2.5}$$

We use the binomial coefficient as an approximation for $\Psi(n, B)$ and try to calculate a good estimate for (2.5) by

$$\begin{aligned}
\ln\left(\frac{\Psi(n,B)}{n}\right) &= \ln\left(\frac{\binom{[u]+B}{B}}{n}\right) \\
&= \ln\left(\frac{([u]+B)!}{[u]!B!}\right) - \ln n \\
&= \ln\left(([u]+B)!\right) - \ln[u]! - \ln B! - u\ln B \\
&= u\ln B + B\ln B - u - B - u\ln u + u - B\ln B + B - u\ln B \\
&= -u\ln u
\end{aligned}$$

We simplified the calculation by replacing $[u]$ with u and $\ln(u + B)$ with $\ln B$, since $u \ll B$. We also used Stirling's formula $\ln n! = n \ln n - n$. Together, this yields

$$\frac{\Psi(n, B)}{n} \approx u^{-u} \tag{2.6}$$

□

We will justify that u is much smaller than both B and $\pi(B)$ with a small example. By the Prime Number Theorem we get that $\pi(B) \approx \frac{B}{\ln B}$. If $n = 10^{70}$ and $B = 10^6$, then $\ln n = 161, \ln B = 14$, $u = \frac{161}{14} = 12$ and $\pi(B) = \frac{10^6}{14} = 10^5$. Hence the claim is clear. Now using the theorem we get that the probability that a random number between 2 and 10^{70} is 10^6-smooth is approximately $\frac{1}{12^{12}}$.

In Theorem 1 we used n as an upper bound for the integers x, but as we will now show, it is too big. Since the sieving begins at $x = \lceil\sqrt{n}\rceil$, the first residues $x^2 \bmod n$ are simply $x^2 - n$. When $\sqrt{n} < x < \sqrt{n} + n^{\varepsilon}$, the upper limit for $x^2 - n$ is

$$\begin{aligned}(\sqrt{n} + n^{\varepsilon})^2 - n &= n + 2n^{\frac{1}{2}+\varepsilon} + n^{2\varepsilon} - n \\ &\approx n^{\frac{1}{2}+\varepsilon} \qquad (2.7)\end{aligned}$$

The probability that x is a B-smooth number is still u^{-u}, but we replace u with the more modest $\frac{\ln n}{2 \ln B}$. Hence the probability that a random number between 2 and 10^{70} is 10^6-smooth is $\frac{1}{6^6}$.

In the remaining part we will be dealing with the analysis of the algorithm and at the end a good estimate for the bound B will follow.

The first important consideration is how long we need to spend with a particular x to decide whether $x^2 - n$ is B-smooth. We use the sieving method described in Section 2.1, where the average number of arithmetic operations spent per value of x is about $\ln \ln B$. This estimate follows from the Eratosthenes sieve where the number of steps are $\sum_{p \leq p_i} b/p$, with p running over the first primes, p_i being the smallest prime such that ${p_i}^2 > b$ and b the biggest integer in the list being sieved. We use Mertens' Second Theorem and get

$$\sum_{p \leq p_i} \frac{b}{p} = b \ln \ln p_i + O(1)$$

Since our primes are less or equal to B, it follows that average time spent per integer $\leq n$ is $\ln \ln B$.

We defined the factor base M to consist of $p_1 = 2$ and the primes $p \leq B$ for which $\left(\frac{n}{p}\right) = 1$. Heuristically, $|M| = \frac{1}{2}\pi(B)$ and we need at least $|M| + 1$ B-smooth values to get a linear dependency among their exponent vectors. Since the probability that x gives a B-smooth number is $\frac{1}{u^u}$, we expect u^u x's to achieve a smooth number. Hence the total amount of x's required are $u^u(|M| + 1)$ and we have found a good estimate on U, the length of the list to sieve in.

The total number of x's together with the amount of work on average on every x gives the time expression

$$T_n = u^u(|M| + 1) \ln \ln B \qquad (2.8)$$

We want to minimize T_n by finding B as a function of n. We begin by computing the logarithm

$$\begin{aligned}\ln T_n &= \ln\left(u^u(|M|+1)\ln\ln B\right)\\ &= u\ln u + \ln\left(\frac{B}{2\ln B}+1\right) + \ln\ln\ln B\\ &\approx u\ln u + \ln B \qquad (2.9)\end{aligned}$$

Now u is replaced with the estimated value $\frac{\ln n}{2\ln B}$ and the derivative calculated

$$\begin{aligned}\ln T_n &= \frac{\ln n}{2\ln B}\ln\left(\frac{\ln n}{2\ln B}\right) + \ln B\\ &= \frac{\ln n}{2\ln B}(\ln\ln n - \ln 2 - \ln\ln B) + \ln B\end{aligned}$$

$$\begin{aligned}\frac{d\ln T_n}{dB} &= \frac{-\ln n}{2(\ln B)^2 B}(\ln\ln n - \ln 2 - \ln\ln B) + \frac{\ln n}{2\ln B}\left(-\frac{1}{\ln B}\frac{1}{B}\right) + \frac{1}{B}\\ &= \frac{-\ln n}{2(\ln B)^2 B}(\ln\ln n - \ln 2 - \ln\ln B + 1) + \frac{1}{B}\\ &= \frac{-\ln n(\ln\ln n - \ln 2 - \ln\ln B + 1) + 2(\ln B)^2}{2(\ln B)^2 B}\end{aligned}$$

We let the derivative be equal to zero and find an estimate for B

$$\begin{aligned}2(\ln B)^2 &= -\ln n(\ln\ln n - \ln 2 - \ln\ln B + 1)\\ \ln B &= \sqrt{\frac{1}{2}\ln n(\ln\ln n - \ln\ln B)} \qquad (2.10)\end{aligned}$$

At this point we need to think a bit since our estimate for B now depends on B. If B is close to n, then $\ln\ln B \approx \ln\ln n$ and $\ln B = a\sqrt{\ln n}$ for some constant a. On the other hand, if B is relatively small compared to n, then $\ln\ln B \ll \ln\ln n$ and $\ln B = b\sqrt{\ln n\ln\ln n}$ for some constant b. We take the logarithms of the two extremal points and find that $\ln\ln B \approx \frac{1}{2}\ln\ln n$. This estimate replaces $\ln\ln B$ in (2.10) and we get

$$\begin{aligned}\ln B &= \sqrt{\frac{1}{2}\ln n(\ln\ln n - \frac{1}{2}\ln\ln n)}\\ &= \frac{1}{2}\sqrt{\ln n\ln\ln n} \qquad (2.11)\end{aligned}$$

Also

$$\begin{aligned}u &= \frac{\ln n}{2\ln B}\\ &= \sqrt{\frac{\ln n}{\ln\ln n}} \qquad (2.12)\end{aligned}$$

We combine (2.9), (2.11) and (2.12)

$$\begin{aligned}
\ln T_n &= u \ln u + \ln B \\
&= \sqrt{\frac{\ln n}{\ln\ln n}}\frac{1}{2}(\ln\ln n - \ln\ln\ln n) + \frac{1}{2}\sqrt{\ln n \ln\ln n} \\
&= \frac{1}{2}\frac{\sqrt{\ln n}(\ln\ln n - \ln\ln\ln n + \ln\ln n)}{\sqrt{\ln\ln n}} \\
&\approx \frac{1}{2}\frac{\sqrt{\ln n}(2\ln\ln n)}{\sqrt{\ln\ln n}} \\
&= \sqrt{\ln n \ln\ln n} \qquad (2.13)
\end{aligned}$$

Finally we can conclude that the optimal choice for B is $e^{\frac{1}{2}\sqrt{\ln n \ln\ln n}}$ and the running time to split n is $T_n = e^{\ln T_n} = e^{\sqrt{\ln n \ln\ln n}} = B^2$.

The subexponential function L is often used to describe the running time of an algorithm, and it is given as

$$L_n(v;c) = e^{(c+o(1))(\ln n)^v(\ln\ln n)^{1-v}}, \text{ for } n \to \infty \qquad (2.14)$$

When $v = 1$ we have exponential time and when $v = 0$ we get polynomial time. So our algorithm will be faster the closer we get towards $v = 0$. In our analysis of the QS algorithm we see that $v = 1/2$ and $c = 1$.

The following theorem from [11] will be helpful in further analysis.

Theorem 2. *Have a sequence $x_1, x_2, \ldots$ of integers in $[1, K]$, each chosen independently and with uniform distribution. Let k be the least integer such that a nonempty subsequence from $x_1, x_2, \ldots, x_k$ has product being a square. Then the expected value for k is $L(K)^{\sqrt{2}+o(1)}$. We can expect the same value for k if all x_i are B-smooth, with $B = L(K)^{1/\sqrt{2}}$.*

In (2.7) we calculated the bound K to be $n^{1/2+\varepsilon}$ and the overall complexity of the quadratic sieve is therefore $L(n)^{1+o(1)}$, with $o(1)$ taking care of the smaller terms. Our complexity estimate is in consensus with Theorem 2, as expected.

If we had taken Montgomery's method in consideration in the analysis it would only have affected the $o(1)$ term.

For the purpose of this thesis, the linear algebra is not studied, but it can be shown that if for instance the Lanczos sparse-matrix method is applied, the matrix has a time bound of $B^{2+o(1)}$. Details can be found in [8].

Example

To illustrate the method we will split $n = 10379$. It is a trivial number and trial divison would use no time, but it will illustrate the method. We let $B = 25$ and find $M = \{2, 5, 17, 19, 23\}$. Since n is so small, the estimated boundaries are not valuable and that is why we chose the bound B bigger than the estimated limit from Section 2.2. We choose $U = 26$ and sieve $f(x) = x^2 - n$ over $[102, 128]$

$$F = \{f(102), f(103), \ldots, f(128)\}$$

We find the smooth numbers

$$\begin{aligned}
f(102) &= 5 \cdot 5 \\
f(103) &= 2 \cdot 5 \cdot 23 \\
f(104) &= 19 \cdot 23 \\
f(123) &= 2 \cdot 5^3 \cdot 19 \\
f(127) &= 2 \cdot 5^3 \cdot 23
\end{aligned}$$

Their associated exponent vectors are collected in a matrix modulo 2

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

After some searching we see that

$$\begin{aligned}
f(103)f(104)f(123) &= (2 \cdot 5 \cdot 23)(19 \cdot 23)(2 \cdot 5^3 \cdot 19) \\
&= (2 \cdot 5^2 \cdot 19 \cdot 23)^2 \\
&\equiv 4370^2 \pmod{10379} \\
&= Y^2
\end{aligned}$$

The other side

$$\begin{aligned}
X &= 103 \cdot 104 \cdot 123 \\
&\equiv 3829 \pmod{10379}
\end{aligned}$$

The $\gcd(4370 \pm 3829, 10379) = 1$, so we must search some more. Let us try $f(103)f(127) = (2 \cdot 5 \cdot 23)(2 \cdot 5^3 \cdot 23)$ with

$$\begin{aligned}
(103 \cdot 127)^2 &\equiv (2 \cdot 5^2 \cdot 23)^2 \pmod{10379} \\
2702^2 &\equiv 1150^2 \pmod{10379}
\end{aligned}$$

Finally, $\gcd(2702 - 1150, 10379) = 97$ and we can conclude that

$$10379 = 97 \cdot 107$$

2.3 The Index-Calculus Method

Given a finite cyclic group G and $g, t \in G$, then a solution z of $g^z = h$ is called the discrete logarithm of t with respect to the base g in G. Locating the z is called the discrete logarithm problem of t with respect to the base g in G.

Although the problem is different from the integer factorization issue, the procedure we now describe from [11] has similiar steps to the already explored QS. To simplify further notation we will just call the index-calculus method the ICM.

Algorithm

We consider a finite field $\mathbb{F}_q$, q a prime, with a generator g for a multiplicatively subgroup and an element t generated by g. We want to find the least integer z such that $g^z \equiv t \pmod{q}$.

Elements in $\mathbb{F}_q$ can be represented as integers and integers can be factorized, hence we can use the idea from the QS.

First we choose a bound B and collect all primes $\leq B$ and -1 in the factor base M. An estimate for the bound is given in the next section. We will gather relations between the elements in M and powers of g, so pick a random $h \in [1, q-2]$ and test for smoothness. If g^h is smooth, then

$$g^h \equiv (-1)^{h_0} p_1^{h_1} \cdots p_y^{h_y} \pmod{q}, \text{ where } y = \pi(B) \text{ and } h_i \in \mathbb{Z} \tag{2.15}$$

Note that this random method is not efficient since the upper bound for g^h is q, but there does not exists a sieve to detect smooth values in the ICM.

We use the relations to find the logarithms of the primes in M. To see how this is done, let $\log_g p_i \equiv x_i \pmod{q-1}$ such that $g^{x_i} \equiv p_i \pmod{q}, \forall i$. Then

$$\begin{aligned} g^h &\equiv (-1)^{h_0} p_1^{h_1} \cdots p_k^{h_y} \\ &\equiv (g^{x_0})^{h_0} (g^{x_1})^{h_1} \cdots (g^{x_y})^{h_y} \\ &\equiv g^{h_0 x_0} g^{h_1 x_1} \cdots g^{h_y x_y} \\ &\equiv g^{h_0 \log_g(-1)} g^{h_1 \log_g p_1} \cdots g^{h_y \log_g p_y} \pmod{q} \end{aligned}$$

The relation (2.15) therefore gives us the congruence

$$h \equiv h_0 \log_g(-1) + h_1 \log_g p_1 + \ldots + h_y \log_g p_y \pmod{q-1}$$

We make the linear system $AX = H$, where the rows in A consists of the h_i values, X the unknown $\log_g p_i$ and H the h values. When we have collected more than $|M|$ relations we use linear algebra to solve for the various $\log_g p_i$'s. Since we might need to invert a nonzero residue over $\mathbb{Z}_{q-1}$, where $q-1$ is composite, this is harder than previous linear algebra over $\mathbb{Z}_2$. The problem can be reduced to linear algebra over the different prime divisors $q-1$. Then it is just linear algebra over a finite field, so usual reduction methods will do.

If one of the prime divisors p are such that $p^i|q-1$, we need to lift the located solution $AX \equiv H \pmod{p}$ to a solution $AX \equiv H \pmod{p^i}$. Assume we have X_1 such that $AX_1 \equiv H \pmod{p}$, meaning $AX_1 = H - pY_1$ for some Y_1. If we solve $AX_2 \equiv Y_1 \pmod{p^2}$, then $A(X_1 + pX_2) \equiv H \pmod{p^2}$ [13]. Repeat the process until the power i is reached. Now using the Chinese Remainder Theorem on all the prime divisors and prime power divisors will give the final solution X mod $q-1$.

The last stage is to find an element $s \in [1, q-2]$ such that $g^s t \pmod{q}$ is close to zero and B-smooth. Let $g^s t \equiv (-1)^{s_0} p_1^{s_1} \cdots p_y^{s_y} \pmod{q}$. The solution to the discrete logarithm problem is then given as

$$\log_g t \equiv -s + s_0 \log_g(-1) + s_1 \log_g p_1 + \ldots + s_y \log_g p_y \pmod{q-1}$$

Analysis

We will just briefly sketch an analysis, since we are merely interested in ICM as a preparation for Chapter 5.

As with the QS, Theorem 2 provides a good time estimate and choice for B. The relations in (2.15) are bounded by q, since g^h can take any value in $\mathbb{Z}_q^*$. So the best estimate for B is $L(q)^c$ for some constant c which depends on how we test the g^h values for smoothness and how we perform the linear algebra. If we test by just trial division, then $c = 1/2$ and $B = L(q)^{1/2}$ with the overall running time $L(q)^{2+o(1)}$. If the elliptic curve method is used, then $c = 1/\sqrt{2}$ and the running time is $L(q)^{\sqrt{2}+o(1)}$. For more details, see [10].

The same procedure as in the QS can also be followed to analyse the ICM and give a limit B.

The probability that an integer g^h is B-smooth is still given in Theorem 1.

How long do we need to spend with a particular element to decide if it is B-smooth? We consider trial division and expect $O(\pi(B)) = O(B/\ln B)$ divisions, each requiring $O((\ln K)^2)$ time, so a total time of $O((\ln K)^2 B/\ln B)$. We can express how many relations we need with a constant a times the number of elements in the factor base $\pi(B)$, and we still need u^u values for one success. The total number of integers to test with the total amount of time used is therefore

$$T_q = au^u \left(\frac{B \ln K}{\ln B}\right)^2$$

Take the logarithm of T_q and calulate the derivative with respect to B. After removing small terms and sat equal to zero, we find the optimal choice

$$B = L_q\left(1/2; c_B + o(1)\right)$$

According to Theorem 2, total complexity is $B^{2+o(1)}$. Study [10, 16] for details.

CHAPTER

THREE

ALGEBRAIC SETTING

The number field sieve for integer factorization share strategy with the QS from Section 2.2 and the number field sieve for the discrete logarithm problem uses ideas from both the QS and the ICM from Section 2.3. So we have an idea on how the two main algorithms in this thesis works. However, when we present the two algorithms in their respective chapters, we will see that they are far more sophisticated than previous methods. Some of these features the algorithms share, as their names might reveal, and we present them in this chapter.

The ring will be given in Section 3.1. Section 3.2 provides us the polynomial and the map, while the concept norm is explored in Section 3.3. The last section presents the common sieving part.

It will be clear in the presentations of the algorithms how this algebraic number theory is linked and used.

To ease notation, we refer to the number field sieve for the integer factorization when we write NFS, and NFS-dlog when we talk about the number field sieve for the discrete logarithm problem.

3.1 The Ring

Assume we have a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and a root $\theta \in \mathbb{C}$. We have the number field $\mathbb{Q}(\theta)$, the finite field extension of $\mathbb{Q}$ of degree $[\mathbb{Q}(\theta) : \mathbb{Q}] = d$.

Definition 2. *A complex number α is called an* algebraic integer *if it is a root of some monic polynomial with coefficients in $\mathbb{Z}$.*

Proposition 1. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$, the set of all algebraic integers in $\mathbb{Q}(\theta)$, $\mathcal{O}_{\mathbb{Q}(\theta)}$, forms a subring of the field $\mathbb{Q}(\theta)$.*

In order to prove the proposition, we use the following proposition from [7].

Proposition 2. *An element $\alpha \in \mathbb{Q}(\theta)$ is an algebraic integer if and only if there exists a nonzero $\mathbb{Z}$-submodule $\mathbf{M}$ of $\mathbb{Q}(\theta)$ such that $\alpha\mathbf{M} \subset \mathbf{M}$.*

Proof. We have $\alpha^k + a_{k-1}\alpha^{k-1} + \ldots + a_0 = 0, a_i \in \mathbb{Z}$, some degree k. Then the $\mathbb{Z}$-submodule $\mathbf{M}$ of $\mathbb{Q}(\theta)$ with basis $\{1, \alpha, \ldots, \alpha^{k-1}\}$ satisfy $\alpha\mathbf{M} \subset \mathbf{M}$.

The other way Cramer's rule is used, stating: If $\sum_{j=1}^{k} a_{i,j}x_j = b_i$, for $1 \le i \le k$, then $x_j = \det(A_j)/\det(A)$, where A_j is obtained from $A = (a_{i,j})$ by replacing the elements in the jth column by the b_i's.

We have a $\mathbb{Z}$-module $\mathbf{M}$ of $\mathbb{Q}(\theta)$ such that $\alpha\mathbf{M} \subset \mathbf{M}$ with basis $\{y_1, y_2, \ldots, y_m\}$. We can express each basis element as

$$\alpha y_i = \sum a_{i,j}y_j, \ a_{i,j} \in \mathbb{Z}$$

Combine and rewrite the linear equations into

$$\begin{aligned}
(\alpha - a_{1,1})y_1 - a_{1,2}y_2 - \ldots - a_{1,m}y_m &= 0 \\
-a_{2,1}y_1 + (\alpha - a_{2,2})y_2 - \ldots - a_{2,m}y_m &= 0 \\
&\vdots \\
-a_{m,1}y_1 - a_{m,2}y_2 - \ldots + (\alpha - a_{m,m})y_m &= 0
\end{aligned}$$

Let A be the matrix of coefficients on the left side. Then Cramer's rule gives that $\det(A)y_i = 0, \forall i$. Since not all generators are zero and $\mathbb{Q}(\theta)$ is a field, we get that $\det(A) = 0$, which is the same as $\alpha^k + c_{k-1}\alpha^{k-1} + \ldots + c_0 = 0$, some $c_i \in \mathbb{Z}$, when written out. □

Proof of Proposition 1. First, if α and β are in $\mathbb{Q}(\theta)$, so are $\alpha\beta$ and $\alpha \pm \beta$, since $\mathbb{Q}(\theta)$ is a field.

Let α be a root of a monic polynomial $h(x)$ with integer coefficients. Then one of the $\pm h(-x)$ will also be a monic polynomial with integer coefficients and have $-\alpha$ as an algebraic integer.

Let α and β be algebraic integers. By Proposition 2, there exists finitely generated $\mathbb{Z}$-modules $\mathbf{M}_1$ and $\mathbf{M}_2$ such that $\alpha\mathbf{M}_1 \subset \mathbf{M}_1$ and $\beta\mathbf{M}_2 \subset \mathbf{M}_2$. We define $\mathbf{M}_1\mathbf{M}_2 = \{\sum m_{1,i}m_{2,i} | m_{1,i} \in \mathbf{M}_1, m_{2,i} \in \mathbf{M}_2\}$. $\mathbf{M}_1\mathbf{M}_2$ will be a finitely generated $\mathbb{Z}$-submodule since its generators are just the product of the generators of $\mathbf{M}_1$ and $\mathbf{M}_2$. It follows that $(\alpha \pm \beta)\mathbf{M}_1\mathbf{M}_2 \subset \mathbf{M}_1\mathbf{M}_2$ and $\alpha\beta\mathbf{M}_1\mathbf{M}_2 \subset \mathbf{M}_1\mathbf{M}_2$. □

We denote $\mathcal{O}_{\mathbb{Q}(\theta)}$ the ring of integers and it have some remarkable features, some which we will outline later. To ease notation, we will just write $\mathcal{O}$ when we refer to the ring of integers of $\mathbb{Q}(\theta)$. Notice that the ring of integers of $\mathbb{Q}$ is simply $\mathbb{Z}$ and we can make the following diagram

$$\begin{tikzcd}
\mathbb{Q}(\theta) \arrow[r, dash] \arrow[d, dash] & \mathcal{O}_{\mathbb{Q}(\theta)} \arrow[d, dash] \\
\mathbb{Q} \arrow[r, dash] & \mathbb{Z}
\end{tikzcd}$$

We will work in a subring of $\mathcal{O}$ (and a ring extension of $\mathbb{Z}$) given below.

Proposition 3. *Given a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and a root $\theta \in \mathbb{C}$, the set of all $\mathbb{Z}$-linear combinations of the elements $\{1, \theta, \ldots, \theta^{d-1}\}$ forms a subring $\mathbb{Z}[\theta] \subseteq \mathcal{O}$.*

Remark that $\mathbb{Z}[\theta]$ can be a proper subring of $\mathcal{O}$, since the number field $\mathbb{Q}(\theta)$ can contain algebraic integers which is not a $\mathbb{Z}$-linear combination of $\{1, \theta, \ldots, \theta^{d-1}\}$ and hence is not in $\mathbb{Z}[\theta]$ [1]. This we can easily see with the monic irreducible polynomial $x^2 - 5$. It produces the field extension $\mathbb{Q}(\sqrt{5})$. In the subring $\mathcal{O}_{\mathbb{Q}(\sqrt{5})}$ we have the algebraic integer $\alpha = (1 + \sqrt{5})/2$, which is a root in $x^2 - x - 1$. Since it is not a $\mathbb{Z}$-linear combination of $\{1, \sqrt{5}\}$, $\alpha \notin \mathbb{Z}[\sqrt{5}]$. Hence, $\mathbb{Z}[\sqrt{5}]$ is a proper subring of $\mathcal{O}_{\mathbb{Q}(\sqrt{5})}$.

3.2 Choosing a Polynomial

We will work in $\mathbb{Z}[\theta]$, the ring generated by one of the roots θ of $f(x)$. Where did $f(x)$ come from?

We want $f(x)$ to have small integer coefficients and in the NFS we want an integer $m \in \mathbb{Z}$ such that $f(m)$ is a multiple of n. In the NFS-dlog, we want it to be a multiple of q, but it is the same procedure. It is accomplished by choosing the degree d such that $d^{2d^2} < n$, $m = \lfloor n^{\frac{1}{d}} \rfloor$ and writing n in base m

$$n = m^d + c_{d-1}m^{d-1} + \ldots + c_0, \text{ with } c_i \in [0, m-1]$$

The method gives us a monic polynomial $f(x) = x^d + c_{d-1}x^{d-1} + \ldots + c_0$ with integer coefficients and $m \in \mathbb{Z}$ such that $f(m) \equiv 0 \pmod{n}$. We discuss the degree d in Chapter 6.

In addition we require the polynomial to be irreducible. If it is not irreducible in the NFS, then there could exist $g(x)$ and $h(x)$ such that $f(x) = g(x)h(x)$, which implies that

$$n = f(m) = g(m)h(m)$$

This means that we have found the two nontrivial factors of n. The probability that this situation occur is so small that $f(x)$ will most likely be irreducible.

If $f(x)$ is not irreducible in the NFS-dlog, we use instead an irreducible factor of f which has m as a root mod q.

Now there exist a natural ring homomorphism

$$\begin{aligned} \varphi : \mathbb{Z}[\theta] &\to \mathbb{Z}_n \\ \theta &\mapsto m \pmod{n} \end{aligned} \tag{3.1}$$

The map is well-defined and by construction $f(m) \equiv 0 \pmod{n}$.

In the NFS-dlog we also want the discriminant of $f(x)$ to be relatively prime to q and all prime divisors of $q - 1$. It is not crucial for the algorithm to work, but it simplifies the calculation. The property is explored in Chapter 5.

3.3 The Norm and Factoring Ideals

In order to use the ideas of QS and ICM in $\mathbb{Z}[\theta]$, we need to generalize the notion of primes and smooth elements. The natural idea would be to look for the "primes" in $\mathbb{Z}[\theta]$, namely the irreducible elements. However, we would then need to assume that $\mathbb{Z}[\theta]$ is a unique factorization domain and expand the factor base with the units. We are studying the general case, and $\mathbb{Z}[\theta]$ is not a UFD in general, just look at $4 \in \mathbb{Z}[\sqrt{5}]$. It is $2 \cdot 2 = 4 = (3 - \sqrt{5}) \cdot (3 + \sqrt{5})$, neither being associates. Also locating all the units and finding a proper place for them in the algorithm could be hard. To skip this cumbersome notation and to overcome the fact that unique factorization of irreducible elements in a ring does not apply in general, we use this section to show that every nonzero ideal factorizes uniquely, up to order, into a product of prime ideals, and illustrate how this is helpful [1, 15]. First some preliminaries.

An integral domain $\mathbf{A}$ has a field of fractions K with the property that every $k \in K$ can be written as $k = a_1^{-1}a_2$ for $a_1, a_2 \in \mathbf{A}$ and $a_1 \neq 0$. The field of fractions of $\mathcal{O}$ is $\mathbb{Q}(\theta)$ and $\mathbb{Q}$ is the field of fractions of $\mathbb{Z}$.

Definition 3. *An integral domain* $\mathbf{A}$ *is* integrally closed *if whenever α is in the field of fractions of* $\mathbf{A}$ *and it satisfies a monic polynomial in* $\mathbf{A}[x]$*, then $\alpha \in \mathbf{A}$.*

Definition 4. *A* Dedekind domain *is an integral domain which is a Noetherian ring, is integrally closed and every nonzero prime ideal is maximal.*

Since $\mathbb{Z}$ is a PID, it satisfies all demands and is a Dedekind domain.

Theorem 3. *$\mathcal{O}$ is a Dedekind domain.*

Proof. $\mathcal{O}$ is integrally closed in $\mathbb{Q}(\theta)$, hence $\mathcal{O}$ is integrally closed.

$\mathcal{O}$ is finitely generated as a $\mathbb{Z}$-module, so it is finitely generated as a ring over $\mathbb{Z}$. Since $\mathbb{Z}$ is Noetherian, we can conclude from the Hilbert Basis Theorem that $\mathcal{O}$ is Noetherian.

Pick an element $x \in \mathfrak{p}$, a nonzero prime ideal in $\mathcal{O}$. Since $x \in \mathcal{O}$, it satisfies

$$x^f + a_{f-1}x^{f-1} + \ldots + a_0 \text{ , with } a_i \in \mathbb{Z}$$

Let the degree f be as small as possible so that $a_0 \neq 0$. If we solve for a_0, we get that

$$a_0 \in \mathcal{O}x \cap \mathbb{Z} \subset \mathfrak{p} \cap \mathbb{Z}$$

Since $\mathfrak{p} \cap \mathbb{Z}$ is a nonzero prime ideal in $\mathbb{Z}$ and maximal by assumption, $\mathbb{Z}/(\mathfrak{p} \cap \mathbb{Z})$ is a field. Also, $\mathcal{O}/\mathfrak{p}$ is an integral domain containing $\mathbb{Z}/(\mathfrak{p} \cap \mathbb{Z})$ and since all elements in $\mathcal{O}$ are algebraic integers, $\mathcal{O}/\mathfrak{p}$ will be algebraic over $\mathbb{Z}/(\mathfrak{p} \cap \mathbb{Z})$ and hence a field. It follows that $\mathfrak{p}$ is a maximal ideal in $\mathcal{O}$. □

Theorem 4. *Every proper nonzero ideal of $\mathcal{O}$ can be written as a product of prime ideals of $\mathcal{O}$ and this representation is unique up to order.*

To prove the theorem, we need the three following lemmas from [7].

Lemma 5. *Every ideal in a Noetherian ring contains a product of nonzero prime ideals.*

Proof. Assume there is a maximal ideal I not containing a product of prime ideals. Then the ideal is not prime and there exist $x_1x_2 \in I$ such that neither $x_1 \in I$ or $x_2 \in I$. But now $I + \langle x_1 \rangle$ and $I + \langle x_2 \rangle$ strictly contains I at the same time as their product is in I. Since I is a maximal counterexample, both $I + \langle x_1 \rangle$ and $I + \langle x_2 \rangle$ contains a product of nonzero prime ideals, and hence so does I. □

For the next lemma, recall that two ideals I_1 and I_2 in a ring $\mathbf{R}$ are relatively prime if $I_1 + I_2 = \mathbf{R}$.

Lemma 6. *If I_1 and I_2 are relatively prime ideals in a ring, then so are I_1^i and I_2^j, for some $i, j \in \mathbb{N}$.*

Proof. Assume I_1^i and I_2^j are not relatively prime. Then they are both contained in a prime ideal. But then so are I_1 and I_2, a contradiction. □

Definition 5. *An integral domain is called a* discrete valuation ring *if it is a PID and has a unique maximal ideal.*

For every nonzero prime ideal $\mathfrak{p}$ in $\mathcal{O}$, we get that the local ring $\mathcal{O}_\mathfrak{p}$ is a discrete valuation ring.

Lemma 7. *Let $\mathfrak{p}$ be a maximal ideal in $\mathcal{O}$, $\mathfrak{q} = \mathfrak{p}\mathcal{O}_\mathfrak{p}$ and $o \in \mathcal{O}$. Then the following map is an isomorphism:*

$$\begin{aligned} \phi : \mathcal{O}/\mathfrak{p}^i &\to \mathcal{O}_\mathfrak{p}/\mathfrak{q}^i \\ o + \mathfrak{p}^i &\mapsto o + \mathfrak{q}^i \end{aligned}$$

A proof of Lemma 7 can be found in [7].

Proof of Theorem 4. Let I be an ideal in $\mathcal{O}$.

By Lemma 5 we know I contains a product of distinct prime ideals, say $I_1 = \mathfrak{p}_1^{f_i} \cdots \mathfrak{p}_k^{f_k}$. If we let $\mathfrak{q}_i$ be the maximal ideal of $\mathcal{O}_{\mathfrak{p}_i}$, we can write

$$\begin{aligned} \mathcal{O}/I_1 &\simeq \mathcal{O}/\mathfrak{p}_1^{f_1} \times \cdots \times \mathcal{O}/\mathfrak{p}_k^{f_k} \\ &\simeq \mathcal{O}_{\mathfrak{p}_1}/\mathfrak{q}_1^{f_1} \times \cdots \mathcal{O}_{\mathfrak{p}_k}/\mathfrak{q}_k^{f_k} \end{aligned}$$

The first isomorphism is given by the Chinese Remainder Theorem and Lemma 6 and the last one of Lemma 7. Since the $\mathcal{O}_{\mathfrak{p}_i}$'s are discrete valuation rings, I/I_1 will correspond to $\mathfrak{q}_1^{e_1}/\mathfrak{q}_1^{f_1} \times \cdots \times \mathfrak{q}_k^{e_k}/\mathfrak{q}_k^{f_k}$, for $e_i \leq f_i$ under the last isomorphism above. But now $I = \mathfrak{p}_1^{e_1} \times \cdots \mathfrak{p}_k^{e_k}$ in $\mathcal{O}/I_1$, because I/I_1 is the image of $\mathfrak{p}_1^{e_1} \times \cdots \times \mathfrak{p}_k^{e_k}$. Since there is a one-to-one correspondence between ideals in $\mathcal{O}/I_1$ and the ideals in $\mathcal{O}$ containing I_1, we get that $I = \mathfrak{p}_1^{e_1} \times \cdots \times \mathfrak{p}_k^{e_k}$ in $\mathcal{O}$.

It remains to prove the uniqueness part. Assume $\mathfrak{p}_1\mathfrak{p}_2 \cdots \mathfrak{p}_k = \mathfrak{p}'_1\mathfrak{p}'_2 \cdots \mathfrak{p}'_{k'}$, and that $\mathfrak{p}'_i$ is not contained in $\mathfrak{p}_1$, for any i. Then we have elements $a_i \in \mathfrak{p}'_i$ such that $a_i \notin \mathfrak{p}_1, \forall i$. But $a_1a_2 \cdots a_{k'} \in \mathfrak{p}_1 \cdots \mathfrak{p}_k \subset \mathfrak{p}_1$, contradicting $\mathfrak{p}_1$ being prime. Hence $\mathfrak{p}_1 = \mathfrak{p}'_i$, for some i. Now remove both ideals from the equation and repeat the argument to obtain uniqueness. □

Theorem 4 will be our main tool in the search for smooth elements in $\mathbb{Z}[\theta]$, as we define them to be smooth if the principal ideal they generates factorizes completely over a chosen factor base of prime ideals.

To be able to use this theory on a computer we must associate the factorization of elements in $\mathbb{Z}[\theta]$ to a factorization in $\mathbb{Z}$. This is easily accomplished by the function norm.

Definition 6. *Let $\alpha \in \mathbb{Q}(\theta)$ define the $\mathbb{Q}$-linear transformation*

$$\begin{aligned} \mathbf{l}_\alpha : \mathbb{Q}(\theta) &\to \mathbb{Q}(\theta) \\ x &\mapsto \alpha x \end{aligned}$$

Since $\mathbb{Q}(\theta)$ is a free $\mathbb{Q}$-module of rank d, we can make an invariant basis $\{e_1, \ldots, e_d\}$ for $\mathbb{Q}(\theta)$ over $\mathbb{Q}$ and express $\alpha e_i = \sum a_{i,j} e_j$. The trace *of α is defined as*

$$Tr(\alpha) = \sum a_{i,i}$$

The norm *of α is defined as*

$$N(\alpha) = \det(\mathbf{l}_\alpha) = \det(a_{i,j})$$

Let Q be a nonsingular matrix and consider the norm of QA/Q, A a matrix. It is the norm of Q times A/Q, or simply A. Similar matrices represent the same linear transformation under two different bases, with Q being the change of basis matrix, and hence the norm is invariant to the basis. Same for the trace.

Theorem 8. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$, there are exactly d embeddings from $\mathbb{Q}(\theta)$ to $\mathbb{C}$, given by $\sigma_i(\mathbb{Q}) = \mathbb{Q}$ and $\sigma_i(\theta) = \theta_i$ for $1 \leq i \leq d$ and the norm of $\alpha \in \mathbb{Q}(\theta)$ is*

$$N(\alpha) = \sigma_1(\alpha)\sigma_2(\alpha)\ldots\sigma_d(\alpha)$$

Proof. Each σ_i gives a distinct isomorphic copy of $\mathbb{Q}(\theta)$ by $\sigma_i : \mathbb{Q}(\theta) \to \mathbb{Q}(\theta_i)$, for $1 \leq i \leq d$. Hence there is at least d embeddings. Assume we have one more such that

$$\begin{aligned} \sigma_{d+1} : \mathbb{Q}(\theta) &\to \mathbb{C} \\ \theta &\mapsto \beta \text{ , for some } \beta \in \mathbb{C} \end{aligned}$$

Then

$$\begin{aligned} f(\beta) &= \beta^d + c_{d-1}\beta^{d-1} + \ldots + c_0 \\ &= \sigma_{d+1}(\theta)^d + c_{d-1}\sigma_{d+1}(\theta)^{d-1} + \ldots + c_0 \\ &= \sigma_{d+1}(\theta^d + c_{d-1}\theta^{d-1} + \ldots + c_0) \\ &= \sigma_{d+1}(0) = 0 \end{aligned}$$

Hence $\beta = \theta_i$ and $\sigma_{d+1} = \sigma_i$, for some $i \in [1, d]$.

To show the last part we compute the characteristic polynomial of α since its constant term is $|\det(\mathbf{l}_\alpha)|$. Let $v(x)$ be the minimal polynomial of α over $\mathbb{Q}$. We make the field extension $\mathbb{Q}(\alpha)$ of degree $\deg(v)$ with basis $\{1, \alpha, \ldots, \alpha^{\deg(v)-1}\}$ over $\mathbb{Q}$. The characteristic polynomial of $\mathbf{l}_\alpha$ acting on $\mathbb{Q}(\alpha)$ is just v, since α satisfies a polynomial if and only if $\mathbf{l}_\alpha$ does. Let $\{e_1, \ldots, e_k\}$ be a basis for $\mathbb{Q}(\theta)$ over $\mathbb{Q}(\alpha)$. Now $\{\alpha^i e_j\}_{i,j}$ is a basis for $\mathbb{Q}(\theta)$ over $\mathbb{Q}$ and $\mathbf{l}_\alpha$ acts invariant on $\{\alpha^i e_j\}_i$ for every fixed j. The characteristic polynomial of $\mathbf{l}_\alpha$ will therefore be $v^{[\mathbb{Q}(\theta):\mathbb{Q}(\alpha)]}$, since the matrix of $\mathbf{l}_\alpha$ is a block direct sum of copies of the matrix of $\mathbf{l}_\alpha$ acting on $\mathbb{Q}(\alpha)$. The roots of $v^{[\mathbb{Q}(\theta):\mathbb{Q}(\alpha)]}$ are the images of $\sigma_i(\alpha)$ with multiplicity $[\mathbb{Q}(\theta) : \mathbb{Q}(\alpha)]$ and the claim follows [15]. □

Proposition 4. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$, the norm maps elements from $\mathbb{Q}(\theta)$ to $\mathbb{Q}$. In particular, algebraic integers in $\mathbb{Q}(\theta)$ are mapped to elements in $\mathbb{Z}$.*

Proof. We show the last claim, since it is the relevant part for us. By Theorem 8 we only need to prove that the minimal polynomial has integer coefficients. Let $v(x)$ be the minimal polynomial to the algebraic integer α and assume $v(x) \notin \mathbb{Z}[x]$. Since α is an algebraic integer, there exists a $u(x) \in \mathbb{Z}[x]$ such that $u(\alpha) = 0$. Also $u(x) = v(x)w(x)$ for some $w(x) \in \mathbb{Q}[x]$. Now a prime p_v will divide the denominator of some coefficient of v. Let p_v^i be the highest power dividing the denominator and likewise let p_v^j be the highest power dividing some denominator of a coefficient of w. Then $p_v^{i+j}u = (p_v^i v)(p_v^j w)$. Now the left side is 0 mod p_v, while the right side is a nonzero product of two nonzero polynomials, a contradiction. Hence the minimal polynomial of α has integer coefficients and the norm maps to $\mathbb{Z}$. □

Corollary 9. *Given a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and a root $\theta \in \mathbb{C}$, the norm maps elements from $\mathbb{Z}[\theta]$ to $\mathbb{Z}$.*

Since $\mathbb{Z}(\theta) \subseteq \mathcal{O}$, the statement is clear.

We will see the full strength of Corollary 9 when we tie the norm of an element in $\mathcal{O}$ together with the norm of the ideal generated by the same element.

Definition 7. *Given a ring $\mathbf{R}$ and an ideal I of $\mathbf{R}$, the* norm *of I is defined to be $[\mathbf{R} : I] = |\mathbf{R}/I|$.*

Proposition 5. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$, then the norm maps ideals in $\mathcal{O}$ to positive integers. Moreover, if $\alpha \in \mathbb{Z}[\theta]$, then $N(\langle\alpha\rangle) = |N(\alpha)|$.*

Proposition 6. *If $\mathfrak{p}$ is an ideal of $\mathcal{O}$ with $N(\mathfrak{p}) = p$ for some prime integer p, then $\mathfrak{p}$ is a prime ideal of $\mathcal{O}$. Conversely, if $\mathfrak{p}$ is a prime ideal of $\mathcal{O}$, then $N(\mathfrak{p}) = p^{e_\mathfrak{p}}$, for some prime integer p and some positive integer $e_\mathfrak{p}$.*

Proof. If $N(\mathfrak{p}) = p$, then $[\mathcal{O} : \mathfrak{p}] = p$, which implies that $\mathcal{O}/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$, a field. Then $\mathfrak{p}$ is a maximal ideal and hence a prime ideal.

Every prime ideal contains a unique prime element, so let $p \in \mathfrak{p}$. Now $\mathcal{O}/\mathfrak{p}$ will be a finite extension of $\mathbb{Z}/p\mathbb{Z}$ of degree $e_\mathfrak{p}$ and hence $N(\mathfrak{p}) = |\mathcal{O}/\mathfrak{p}| = p^{e_\mathfrak{p}}$. □

We know from Theorem 3 that $\mathcal{O}$ is a Dedekind domain, so if we have an element $\alpha \in \mathcal{O}$, the principal ideal $\langle \alpha \rangle$ will by Theorem 4 factorize uniquely as

$$\langle \alpha \rangle = \mathfrak{p}_1^{d_1} \mathfrak{p}_2^{d_2} \dots \mathfrak{p}_k^{d_k}$$

The prime ideals $\mathfrak{p}_i$ of $\mathcal{O}$ are distinct and all exponents are natural numbers. Furthermore, by Corollary 9, Proposition 5 and Proposition 6 we get

$$\begin{aligned} |N(\alpha)| = N(\langle \alpha \rangle) &= N(\mathfrak{p}_1^{d_1} \mathfrak{p}_2^{d_2} \dots \mathfrak{p}_k^{d_k}) \\ &= N(\mathfrak{p}_1)^{d_1} N(\mathfrak{p}_2)^{d_2} \dots N(\mathfrak{p}_k)^{d_k} \\ &= (p_1^{f_1})^{d_1} (p_2^{f_2})^{d_2} \dots (p_k^{f_k})^{d_k} \text{ , } f_i \in \mathbb{N} \\ &= p_1^{f_1+d_1} p_2^{f_2+d_2} \dots p_k^{f_k+d_k} \end{aligned} \tag{3.2}$$

Notice that the primes in (3.2) not necessarily are distinct.

We defined an element to be smooth if the ideal it generates factorizes over a chosen factor base of prime ideals and we are finally able to give the factor base. It will consist of the prime ideals with norm divisible only by primes less than or equal to a bound. We name it the algebraic factor base. An element in $\mathbb{Z}[\theta]$ is therefore smooth if when we consider the norm of the ideal the element generates it factorizes completely over the prime integers from the norm of the prime ideals in the algebraic factor base.

3.4 The Sieve

We have a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients, a root $\theta \in \mathbb{C}$, an integer m such that $f(m) \equiv 0 \pmod{n}$ and we have $\mathbb{Z}[\theta]$, the ring generated by the root θ of f. We want to find a set T of pairs (a, b) such that both $a + bm \in \mathbb{Z}$ and $a + b\theta \in \mathbb{Z}[\theta]$ are smooth.

Let U be a bound. The overall set of pairs where we will find the T from is

$$\mathcal{U} = \{(a, b) | a, b \in \mathbb{Z}, \gcd(a, b) = 1, |a| \leq U, 0 < b \leq U\}$$

The parameter U need to be sufficiently large so we can find enough elements in T to satisfy further use of the smooth values, but also as small as possible to avoid being a time-bottleneck. We estimate U in Chapter 6.

We will locate the smooth elements using two sieves. The first one is described in the next section and we call it the rational sieve since it finds the smooth elements in $\mathbb{Z}$. The other is given in Section 3.4 as the algebraic sieve, locating the smooth algebraic integers. To obtain the set T, we combine the outcome of the two sieves.

We lastly briefly mention another more complicated method to locate the smooth elements, the lattice sieve.

The NFS and the NFS-dlog uses the smooth elements in different ways and therefore meet individual barriers. We will describe how to overcome these in the chapters describing the algorithms respectively.

The Rational Sieve

To locate the smooth $a + bm$ values we proceed in the same fashion as in the QS, the only difference is that in the QS we had just one variable. Now there are two, both a and b. It is common to fix b and sieve over the range of a, then do the next b until enough smooth values are found.

First we decide on a smoothness bound B and make a rational factor base $\mathcal{B} = \{p | p \leq B, p \in \mathbb{Z}, p \text{ prime}\}$. An estimate of B is outlined in Chapter 6. Now fix a $b \in [0, U]$ and make a list of the integers $a + bm$ for $|a| \leq U$. We begin the sieving with the first prime in $\mathcal{B}$, say p_1, and locate the first element in the list such that $a + bm \equiv 0 \pmod{p_1}$. Then it is easy to find the rest of the elements in the list having p_1 as a factor, as they are the elements $a + bm \equiv kp_i$, $k \in \mathbb{Z}$. We also get that the a's will be of the form $a = -bm + kp_i, k \in \mathbb{Z}$.

The sieving works as follows, we have a sieve array of computer memory with a single position assigned to each a and for the fixed b each position is initialized with the suitable $a + bm$ value. When we have located all the a's in $-u \leq a \leq u$ such that $a = -bm + kp_1$ with $k \in \mathbb{Z}$, the prime p_1 is divided out from the number in the position corresponding to the a in the sieve array and the rest is stored. When performed for all the primes in $\mathcal{B}$, the array is scanned for 1's, as they will correspond to smooth numbers. Then the process is repeated for the next b value.

There are a few changes that will speed up the sieve. Divison can be changed into subtraction by storing $\ln(a + bm)$ instead of $a + bm$. Then we subtract $\ln p_i$ from $\ln(a + bm)$ whenever $a + bm \equiv 0 \pmod{p_i}$ and search for numbers close to $0 = \ln(1)$ instead of 1. This rules out most nonsmooth numbers and trial division on remaining numbers will complete the task. We could also initialize the array by zeros and add $\ln p_i$ since adding is a cheaper operation than subtraction.

If $a+bm$ is divisible by $p_i^{e_i}$, $e_i > 1$, $a+bm$ should be divided by $p_i^{e_i}$ not just p_i. Or equivalently, $e_i \ln p_i$ should be added instead of just $\ln p_i$. To make sure that this particular smooth number is not denied as a smooth number, the search can be switched to numbers in a range close to zero.

The different divisors should also be stored in some way, because we are not only interested in the smooth elements, but also their factorization, as we were in the QS.

We collect the smooth integers in $T_1 = \{(a, b) | (a, b) \in \mathcal{U}, a + bm \text{ is } B\text{-smooth}\}$.

The Algebraic Sieve

We are searching for a set T_2 containing the smooth algebraic integers, that is

$$T_2 = \{(a, b) | (a, b) \in \mathcal{U}, a + b\theta \text{ is } B\text{-smooth}\}$$

To find T_2 we will use the norm defined in Section 3.3 together with the idea of the rational sieve. In addition, we present the first degree prime ideals since they can be represented in a way that can be stored on a computer which makes it possible to sieve by prime ideals. And with those ideals defined we are able to show why we want the smooth elements to have their particular shape.

A subset of $\mathbb{Z}[\theta]$ is an ideal if and only if it is the kernel of some ring homomorphism defined on $\mathbb{Z}[\theta]$ and it is a prime ideal if it is equal to the kernel of a ring homomorphism from $\mathbb{Z}[\theta]$ to a finite field. If the finite field is a prime field, then the ideal is called a first degree prime ideal. The statement is equivalent to saying that the norm of the prime ideal is a prime integer.

Proposition 7. *Let $f(x)$ be a monic, irreducible polynomial of degree d with integer coefficients and a root $\theta \in \mathbb{C}$. Let the integers a and b be coprime. Then all prime ideals $\mathfrak{p}$ containing $a + b\theta$ are first degree prime ideals.*

Proof. Let $a + b\theta \in \mathfrak{p}$ and let $\varphi : \mathbb{Z}[\theta] \to \mathbb{F}$, $\mathbb{F}$ a field, such that $\ker \varphi = \mathfrak{p}$. Also, let $\text{char}(\mathbb{F}) = p$ such that $\mathbb{F}_p$ is the prime field of $\mathbb{F}$.

Now $\varphi(a + b\theta) = 0$ so $\varphi(a) = -\varphi(b)\varphi(\theta)$. Since $a, b \in \mathbb{Z}$, $\varphi(a), \varphi(b) \in \mathbb{F}_p$. Also, $\varphi(b) \neq 0$ since otherwise $\varphi(a) = 0$ as well, contradicting $\gcd(a, b) = 1$. Hence $\varphi(\theta) \in \mathbb{F}_p$ and we get that φ is a ring homomorphism from $\mathbb{Z}[\theta]$ to $\mathbb{F}_p$ with kernel $\mathfrak{p}$. Hence, $\mathfrak{p}$ is a first degree prime ideal. □

For the representation we define $R(p) = \{\forall r \in \mathbb{Z}/p\mathbb{Z} | f(r) \equiv 0 \pmod{p}\}$ for each prime p in the rational factor base $\mathcal{B}$. There could be as many as d elements in $R(p)$ for each prime, but heuristically the mean is only one per prime [4].

Proposition 8. *Given a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and a root $\theta \in \mathbb{C}$. The set of all pairs (p, r), with p a prime and $r \in R(p)$, are in a bijective correspondence with all the first degree prime ideals of $\mathbb{Z}[\theta]$.*

Proof. Pick a first degree prime ideal $\mathfrak{p}$ of $\mathbb{Z}[\theta]$. Then $[\mathbb{Z}[\theta] : \mathfrak{p}] = p$ for some prime $p \in \mathbb{Z}$ and the map $\varphi : \mathbb{Z}[\theta] \to \mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ sends θ to $r \bmod p$ and $\varphi(1) = 1$ with kernel $\mathfrak{p}$. Hence r is a root of $f(x) \pmod{p}$ and $\mathfrak{p}$ represents the unique pair (p, r). Each first degree prime ideal gives a unique φ-map so every ideal will correspond to a unique pair (p, r).

In the opposite direction, let p be a prime and $r \in R(p)$. There exists a natural epimorphism that maps polynomials in θ to polynomials in r, by $\varphi(1) \equiv 1 \pmod{p}$ and $\varphi(\theta) \equiv r \pmod{p}$. Let $\ker \varphi = \mathfrak{p}$, a prime ideal in $\mathbb{Z}[\theta]$. Since φ is onto and $\ker \varphi = \mathfrak{p}$, it follows that $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ and that $[\mathbb{Z}[\theta] : \mathfrak{p}] = p$. This implies that $\mathfrak{p}$ is a first degree prime ideal [1]. □

So the first degree prime ideals can be stored on a computer as a pair (p, r) and they are the only prime ideals occuring in the factorization of $\langle a + b\theta \rangle$. Therefore, let the algebraic factor base consist of the first degree prime ideals instead of the general prime ideals

$$\mathcal{A} = \{(p, r) | p \in \mathcal{B}, r \in R(p)\}$$

It remains to connect the algebraic factor base to the norm. We begin by calculating the norm of an element $a + b\theta \in \mathbb{Z}[\theta]$. Theorem 8 gives

$$\begin{aligned}N(a+b\theta) &= \sigma_1(a+b\theta)\sigma_2(a+b\theta)\ldots\sigma_d(a+b\theta)\\ &= (a+b\theta_1)(a+b\theta_2)\ldots(a+b\theta_d)\\ &= -b^d(-a/b-\theta_1)\ldots(-a/b-\theta_d)\\ &= -b^d f(-a/b)\end{aligned}$$

Let $F(x,y)$ be the homogenous form of $f(x) = x^d + c_{d-1}x^{d-1} + \ldots + c_0$ with $c_i \in \mathbb{Z}$ for $0 \le i \le d-1$,

$$\begin{aligned}F(x,y) &= x^d + c_{d-1}x^{d-1}y + \ldots + c_0y^d\\ &= -y^d f(-x/y)\end{aligned}$$

We view $F(a,b) = N(a+b\theta)$ as a polynomial in the two variables a and b, and have just found a practical way to sieve the norm values. However, as the small example below implies, the norm does not distinguish the different primes, so various integers can give rise to the same norm.

Let $f(x) = x^2+1$. We make the field extension $\mathbb{Z}[i]$, with the norm $N(a+bi) = a^2+b^2$. For $3+4i \in \mathbb{Z}[i]$, $N(3+4i) = 3^2+4^2 = 25 = 5^2 = N(5i)$, so two different algebraic integers have the same squared norm. Also, for $5i \in \mathbb{Z}[i]$ we get that $5i = (2+i)(1+2i)$ is not a square in $\mathbb{Z}[i]$, even though $N(5i) = 25$ is a square. So a product $\prod(a+b\theta)$ with squared norm is to weak to imply $\prod(a+b\theta)$ square in $\mathbb{Z}[\theta]$. This squared property is important in the NFS.

We need a function that keeps track of the primes and that is exactly what Proposition 8 gave us [1].

For each prime p in our factor base, the set of zeros of $f \bmod p$, denoted by $R(p) = \{\forall r \in \mathbb{Z}/p\mathbb{Z} | f(r) \equiv 0 \pmod p\}$, are collected in $\mathcal{A}$. For any fixed b, $0 < b < U$, and $b \neq 0 \pmod p$, the integers a that give $N(a+b\theta) \equiv 0 \pmod p$ are the a's with $a \equiv -br \pmod p$ for some $r \in R(p)$. This is because $p|N(a+b\theta)$ and $p \nmid b$ implies that $f(-a/b) \equiv 0 \pmod p$. We notice that if $b \equiv 0 \pmod p$, then $\gcd(a,b) \neq 1$ and $(a,b) \notin \mathcal{U}$.

We fix a b and initialize an array with the numbers $N(a+b\theta)$ for $|a| \le U$. For every $p \le B$ that does not divide b and all $r \in R(p)$, the positions corresponding to the $a \equiv -br \pmod p$ are found, and highest power of p that divide $N(a+b\theta)$ are divided out. The quotient replaces the previous value at this position. When we have sieved all the primes in the factor base, we search through the list for the 1's. These locations will contain the $a+b\theta$ values with B-smooth norm. Collect the smooth algebraic integers in T_2 and repeat with the next b value.

To speed up the sieve we can make the same changes as we did with the rational sieve.

Finally taking the intersection of T_1 and T_2 we get the set T of both smooth $a+bm$ and $a+b\theta$ values.

The Lattice Sieve

Another more complicated approach to locate the smooth values is the lattice sieve introduced by Pollard [4]. Compared to the strategy just given, the lattice sieve sieves fewer integers, but still returns almost all the smooth values. The downside is the time-consuming operations and that collisions of smooth values appear. We sketch the idea.

If we think of all pairs (a, b) with $|a| \leq U, b \leq U$ as a lattice $\mathbf{L}$, we can make the sublattice $\mathbf{L}_q$ with a special prime q containing all the pairs such that $a + bm \equiv 0 \pmod{q}$. We divide the factor base $\mathcal{B}$ into two bases. Let $\mathcal{B}_0$ consist of the small primes below a limit B_0 and $\mathcal{B}_1$ the rest of the primes $\leq B$. The limit B_0 separating $\mathcal{B}_0$ and $\mathcal{B}_1$ should be such that $0.1 \leq |\mathcal{B}_0|/|\mathcal{B}_1| \leq 0.5$. The idea is that if q is chosen from $\mathcal{B}_1$, it is likely that the elements in $\mathbf{L}_q$ are B_0-smooth.

So, the main principle of the sieve is to only sieve in this smaller region $\mathbf{L}_q$ with the smaller factor base $\mathcal{B}_0$ for the $a + bm$'s and sieve $N(a + b\theta)$ with the usual factor base, take the intersection to locate the B-smooth values and repeat with as many special primes as necessary.

Find any basis for the sublattice and compute a reduced basis $(V_1, V_2) = ((a_1, b_1), (a_2, b_2))$ using the LLL basis reduction method. A reduced basis consists of short vectors, meaning vectors of a small orthogonality defect. Now any element $(a, b) \in \mathbf{L}_q$ can be represented as (c_1V_1, c_2V_2), or equivalently $(a, b) = (c_1a_1 + c_2a_2, c_1b_1 + c_2b_2)$, with the c_i's from a smaller interval than the a and b's.

The basis has q as the only common factor. If one of the V_i's has p as a factor, we get that the whole pth row or pth column of $\mathbf{L}_q$ should be sieved. Now say that $\gcd(V_1, p) = 1$. We have two choices for the sieving, depending on the size of the special prime q. Sieving by rows is preferred when the region is large, and sieving by vectors when the region is small. The row sieving requires a small amount of memory compared to the vector sieving.

The row sieving is familiar, every pth element in the row is to be sieved and by calculating the inverse of $V_1 \bmod p$ one finds where to start in the row.

The vector sieving is a more complicated part where one sieves the (c_1, c_2)-plane. Again, the points to be sieved in $\mathbf{L}_q$ form a sublattice, so one can make a reduced basis for the sublattice and sieve in this even smaller area. A good description is given in [5] where they use the lattice sieve to split the 232-digit RSA challenge number.

The smooth $a + bm$ values we miss are the ones that do not have a prime factor in $\mathcal{B}_1$. And since the sieve is done for several special primes, collisions are likely to happen, meaning that a special prime is simply a regular factor base prime for another special prime and the relations are duplicated. The time-consuming operations are the ones involving multiplication and division, as finding the basis and the inverse.

An advantage the lattice sieve has over the line sieve is that the line sieve has declining rate of smooth elements as we move away from the origin, while the lattice sieve brings the sieving region back to the origin again when the special prime is changed. Regardless, we use and study the other approach.

CHAPTER

FOUR

THE NUMBER FIELD SIEVE

In this chapter we will describe the fastest known factorization algorithm to factor large integers as used in the RSA system, known as the general number field sieve.

In the late 80s the QS began to reach its limit. The numbers to split were becoming too big and the smooth numbers was not found fast enough anymore. John Pollard circulated a letter in 1988 with an idea on factoring large composite numbers via algebraic number fields. He was wondering if it was possible to change the polynomial into a cubic, quartic or even higher powers to produce more smooth values than what quadratic polynomials did. And perhaps other rings could have more smooth values than $\mathbb{Z}$ and $\mathbb{Z}/n\mathbb{Z}$? And could these other rings be used to produce a difference of squares?

Many contributed in improving and polishing the proposed algorithm, Carl Pomerance, the Lenstra brothers, Brian LaMacchia and Andrew Odlyzko among others, and in 1996 the method was used to factor a 130-digit challenge number in about 15% of the time the quadratic sieve would have used [9]. A recent record was in 2009 when the 232-digit RSA challenge number was split [5]. This is as yet the largest number with cryptographic meaning ever factored and in addition it tells us that the number field sieve is worth a study.

We will primarily follow the presentations given in [1, 4], but we also found inspiration in [11, 14].

As the method is build on the QS, the strategy is still to find smooth elements and combine them into a congruence of squares. In addition to the two known steps, the sieving and the linear algebra, there is now extra work in picking a suitable polyomial and calculating the square roots.

The strategy is outlined in the next section. Then Section 4.2 will show the smooth numbers in action and introduce the quadratic characters. The square roots are given in Section 4.3 and the algorithm is presented in Section 4.4.

4.1 Strategy

We are given a composite number n, an odd number which is not a power of a prime integer, and we want to split it. As mentioned in the introduction, the general number field sieve share strategy with the quadratic sieve. The idea is to write n as a difference in squares, with a few modifications.

We construct the number ring $\mathbb{Z}[\theta]$ from $f(x)$ as implied in Chapter 3. To see how the homomorphism defined in (3.1) is used in the NFS to produce the difference in squares, assume we have a set S of coprime (a, b) pairs such that

$$\prod_{(a,b)\in S} (a + bm) \text{ is a square in } \mathbb{Z} \tag{4.1}$$

$$\prod_{(a,b)\in S} (a + b\theta) \text{ is a square in } \mathbb{Z}[\theta] \tag{4.2}$$

Let $X \in \mathbb{Z}$ be a square root in (4.1) and $\beta \in \mathbb{Z}[\theta]$ a square root in (4.2). If $Y \equiv \varphi(\beta) \pmod{n}$, then

$$\begin{aligned} Y^2 \equiv \varphi(\beta)^2 &\equiv \varphi(\beta^2) \\ &\equiv \varphi\left(\prod_S (a + b\theta)\right) \\ &\equiv \prod_S \varphi\left(a + b\theta\right) \\ &\equiv \prod_S (a + bm) = X^2 \pmod{n} \end{aligned}$$

We have the desired congruence of squares $Y^2 \equiv X^2 \pmod{n}$ and $\gcd(Y \pm X, n)$ is hopefully a factor of n. If $Y \equiv \pm X \pmod{n}$ or $\gcd(Y \pm X, n) = 1$, we try again with different numbers.

4.2 Constructing the Congruence of Squares

In Section 3.4 we described how to locate the set $T = T_1 \cap T_2$ of coprime (a, b) pairs such that both $a + bm$ and $a + b\theta$ were smooth at the same time. The set S satisfying (4.1) and (4.2) will be a subset of T and we will now show how to locate it. As with the sieving we present the rational and the algebraic side separately.

We have the rational factor base $\mathcal{B} = \{p | p \leq B, p \in \mathbb{Z}, p \text{ prime}\}$ and the set $T_1 = \{(a, b) | (a, b) \in \mathcal{U}, a + bm \text{ is } B\text{-smooth}\}$. Let $y = \pi(B)$, p_i denote the ith prime in $\mathcal{B}$ and $e_i \in \mathbb{Z}$, for $0 \leq i \leq y$. For the B-smooth number $t \in T_1$, we get

$$t = \prod_{i=1}^{y} p_i^{e_i}$$

The associated exponent vector is $\mathbf{v}(t) = (e_1, e_2, \ldots, e_y)$.

We require $|T_1| > \pi(B) + 1$, so that when we form an exponent vector for all $(a, b) \in T_1$, the number of vectors exceed the dimension of the vectors. Then there exists a linearly dependent relation among the vectors, a subset $S_1 \subset T_1$, such that

$$\sum_{(a,b)\in S_1} \mathbf{v}(a + bm) \equiv 0 \pmod 2$$

Hence we have located a set S_1 satisfying

$$\prod_{(a,b)\in S_1} (a + bm) = X_1^2 \in \mathbb{Z}, \text{ for some } X_1 \in \mathbb{Z}$$

This is just a copy of the QS and it is rather straightforward to locate the subset of T_1 satisfying (4.1). The main work will be on the algebraic side, because we cannot imitate what we just did with the exponent vectors. If we tried to combine the exponent vectors of the $N(a + b\theta)$ for $(a, b) \in T_2$ into a square, it would only lead to a subset of T_2 where the norm of the $\prod(a + b\theta)$ is squared.

By Proposition 9 we know that the norm is multiplicative. If $\zeta = \eta^2$ for some $\zeta, \eta \in \mathbb{Z}[\theta]$, then $N(\zeta)$ is an integer square

$$\begin{aligned} N(\zeta) &= N(\eta^2) \\ &= N(\eta)N(\eta) \\ &= N(\eta)^2 \end{aligned}$$

It is therefore necessary that the norm is squared to produce a square in $\mathbb{Z}[\theta]$, but it is not sufficient. As we saw earlier in the small example in Section 3.4, $5i = (2 + i)(1 + 2i)$ is not a square in $\mathbb{Z}[i]$, even though $N(5i) = 25$ is.

We will instead try to find the powers of all the first degree prime ideals $\mathfrak{p}$ in $\langle a + b\theta\rangle$ and mimic the rational side with prime ideals.

If $\mathbb{Z}[\theta] = \mathcal{O}$ we know that all ideals factors uniquely as a product of prime ideals, and we could simply take the next defined function as exponent. Since we look at general rings, we need in addition the next proposition to state it.

Let the $\mathrm{ord}_p(x)$ be the number of times p divides x. We define the function $e_{p,r}(a + b\theta)$ by

$$e_{p,r}(a + b\theta) = \begin{cases} \mathrm{ord}_p(N(a + b\theta)) & \text{if } a + br \equiv 0 \pmod p \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

If we use this, then

$$N(a + b\theta) = \pm \prod_{p,r} p^{e_{p,r}(a+b\theta)}$$

The next nontrivial result that depends upon the Jordan-Hölder theorem establishes a homomorphism of prime ideals.

Proposition 9. *For every prime ideal $\mathfrak{p} \in \mathbb{Z}[\theta]$, there exists a group homomorphism $\mathfrak{l}_{\mathfrak{p}} : \mathbb{Q}(\theta)^* \to \mathbb{Z}$ such that:*

1. $\mathfrak{l}_{\mathfrak{p}}(\alpha) \geq$ *for all* $\alpha \in \mathbb{Z}, \alpha \neq 0$

2. $\mathfrak{l}_{\mathfrak{p}}(\alpha) > 0$ *if and only if* $\alpha \in \mathfrak{p}$ *for* $\alpha \neq 0$

3. *For each* $\alpha \in \mathbb{Q}(\theta)^*$, $\mathfrak{l}_{\mathfrak{p}}(\alpha) = 0$ *for all but finitely many* $\mathfrak{p}$ *and* $\prod_{\mathfrak{p}} N(\mathfrak{p})^{\mathfrak{l}_{\mathfrak{p}}(\alpha)} = |N(\alpha)|$, *where* $\mathfrak{p}$ *are all prime ideals in* $\mathbb{Z}[\theta]$

A proof can be found in [4]. Since we only use principal ideals, the following lemma restrict $e_{p,r}$ to the first degree prime ideals.

Lemma 10. *Let a and b be coprime integers and let $\mathfrak{p} \in \mathbb{Z}[\theta]$. If $\mathfrak{p}$ is not a first degree prime ideal, then $\mathfrak{l}_{\mathfrak{p}}(a + b\theta) = 0$. If $\mathfrak{p}$ is a first degree prime ideal in a bijective correspondence with (p, r), then $\mathfrak{l}_{\mathfrak{p}}(a + b\theta) = e_{p,r}(a + b\theta)$.*

Proposition 9 and Lemma 10 can be used to prove the next proposition.

Proposition 10. *If S_2 is a finite set of coprime integer pairs (a, b) such that $a+b\theta$ is B-smooth for each pair and $\prod_{(a,b)\in S_2}(a + b\theta)$ is the square of an element of $\mathcal{O}$, then, for each prime number p and $r \in R(p)$, we have*

$$\sum_{(a,b)\in S_2} e_{p,r}(a + b\theta) \equiv 0 \pmod 2 \tag{4.4}$$

What we really seek is the opposite of Proposition 10. We will see that the opposite can be assumed to be true if we consider quadratic characters, described later in Section 4.2 [1, 4, 11]. There is now just four small obstructions left in the description of the algebraic side.

Four Barriers

We have a subset $S_2 \subset T_2$ such that

$$\sum_{(a,b)\in S_2} e_{p,r}(a + b\theta) \equiv 0 \pmod 2 \tag{4.5}$$

There are four barriers keeping this from being enough to conclude (4.2). Let

$$\prod_{(a,b)\in S_2} (a + b\theta) = \gamma$$

1. The ideal $\langle\gamma\rangle$ does not need to be a square of an ideal, since we are working in $\mathbb{Z}[\theta]$, not $\mathcal{O}$.

2. Even if $\langle\gamma\rangle = I^2$ for some ideal $I \in \mathcal{O}$, I need not be principal.

3. Even if $\langle\gamma\rangle = \langle\delta\rangle^2$ for some $\delta \in \mathcal{O}$, it may not be that $\gamma = \delta^2$.

4. Even if $\gamma = \delta^2$ for some $\delta \in \mathcal{O}$, it may not be that $\delta \in \mathbb{Z}[\theta]$ [11].

Note that if $\mathbb{Z}[\theta] = \mathcal{O}$, then (1) and (4) would not be a problem. If $\mathcal{O}$ was a PID, then (2) would be fixed. And if $\mathcal{O}$ is a PID and we have all the units of $\mathcal{O}$, then (3) could be solved by including the units in a linear algebra step. Since we study the general case, none of these assumptions can be made. Luckily two modifications take care of these obstructions. The fourth problem is easily overcome by the following lemma.

Lemma 11. *Let $f(x)$ be a monic, irreducible polynomial with integer coefficients and a root $\theta \in \mathbb{C}$. Let $\gamma \in \mathcal{O}$. Then $f'(\theta)\gamma \in \mathbb{Z}[\theta]$.*

Proof. A proposition by Euler states that: If we have $f(x) = x^d + \ldots + c_0 \in \mathbb{Z}[x]$ and the basis $\{1, \theta, \ldots, \theta^{d-1}\}$ for $\mathbb{Z}[\theta]$ over $\mathbb{Z}$ and also $\frac{f(x)}{x-\theta} = b_{d-1}x^{d-1} + \ldots + b_0 \in \mathbb{Z}[\theta][x]$ and the basis $\{b_0/f'(\theta), b_1/f'(\theta), \ldots, b_{d-1}/f'(\theta)\}$ for $\mathbb{Q}(\theta)$ over $\mathbb{Q}$, then the two bases are complementary, meaning $\mathrm{Tr}\left(\theta^i \frac{b_j}{f'(\theta)}\right) = \delta_{ij}$, the Kronecker delta [17].

The trace is the sum of the embeddings outlined in Theorem 8, and as the norm it takes elements from $\mathbb{Q}(\theta)$ to $\mathbb{Q}$. In particular, elements in $\mathcal{O}$ are sent to $\mathbb{Z}$.

For $\gamma \in \mathcal{O}$, let $\gamma = a_{d-1}\frac{b_{d-1}}{f'(\theta)} + \ldots + a_0\frac{b_0}{f'(\theta)}$, $a_i \in \mathbb{Q}$. We want $a_i \in \mathbb{Z}$. But

$$\mathrm{Tr}(\gamma\theta^k) = \mathrm{Tr}\left(\sum_i a_i \frac{b_i}{f'(\theta)}\theta^k\right) = \sum_i a_i \mathrm{Tr}\left(\frac{b_i}{f'(\theta)}\theta^k\right) = \sum_i a_i \delta_{ik} = a_k$$

Hence $a_i \in \mathbb{Z}, \forall i$, and we can conclude that $f'(\theta)\gamma \in \mathbb{Z}[\theta]$ [11] □

Because of Lemma 11 we replace (4.1) and (4.2) with

$$f'(m)^2 \prod_{(a,b)\in S} (a + bm) \text{ being a square in } \mathbb{Z} \tag{4.6}$$

$$f'(\theta)^2 \prod_{(a,b)\in S} (a + b\theta) \text{ being a square in } \mathbb{Z}[\theta] \tag{4.7}$$

We can assume that for $1 < f'(m) < n$, the $\gcd(f'(m), n) = 1$, otherwise n is split. The other obstructions are overcome by the quadratic characters.

The Quadratic Characters

It was Adleman who introduced the quadratic characters and the idea was based upon the Legendre symbols. A general fact about the Legendre symbol $\left(\frac{w}{q}\right)$ is that if q is an odd prime, w an integer and $\left(\frac{w}{q}\right) = -1$, then w is *not* a square. Adleman stated the converse, even though it does not hold in general [11].

That is, we have the integer w and k randomly chosen odd primes q_i, such that $\left(\frac{w}{q_i}\right) = 1, \forall i$. The probability that w is not a square is heuristically 2^{-k}.

This is easily seen. Let $l_n = q_1 q_2 \ldots q_k$. Then $\mathbb{Z}^*_{l_n} \cong \mathbb{Z}^*_{q_1} \times \ldots \times \mathbb{Z}^*_{q_k}$, where

$$\left(\frac{w}{l_n}\right) = \prod_{i=1}^{k} \left(\frac{w}{q_i}\right)$$

To be a square in $\mathbb{Z}^*_{l_n}$ is identical to being a square in all the $\mathbb{Z}^*_{q_i}$'s. Exactly half the elements in $\mathbb{Z}^*_{q_i}$ are squares and we have k primes to check. Therefore, if k is reasonably large and $\left(\frac{w}{q}\right) = 1$ for all k primes, then it is highly likely that w is square, so we assume it is.

We use this idea to make a squareness test for $\prod_{(a,b)\in S_2}(a + b\theta)$, since we have that it is a square of an ideal in $\mathbb{Z}[\theta]$, but we want it to be a square of an element in $\mathbb{Z}[\theta]$. The Legendre symbol will hold for the primes in the factor base, because $\sum_{(a,b)\in S_2} e_{p,r}(a + b\theta) \equiv 0 \pmod 2$. If we test enough primes outside the factor base and the Legendre symbols are all 1, then we can assume with high probability that $\prod_{(a,b)\in S_2}(a+b\theta)$ is a square of an element of $\mathbb{Z}[\theta]$. Because of the next proposition, we can do the squareness test with algebraic integers $a + b\theta$.

Proposition 11. *Let $f(x)$ be a monic, irreducible polynomial of degree d with integer coefficients and a root $\theta \in \mathbb{C}$. Let q be an odd prime integer and $s \in R(q)$ such that $f'(s) \not\equiv 0 \pmod q$. Let S_2 be a set of coprime integers (a, b) such that $f'(\theta)^2 \prod_{(a,b)\in S_2}(a + b\theta)$ is a square in $\mathbb{Z}[\theta]$. Let $q \nmid a + bs$ for any pair $(a, b) \in S_2$. Then*

$$\prod_{(a,b)\in S_2} \left(\frac{a + bs}{q}\right) = 1 \tag{4.8}$$

Proof. Let $\varphi : \mathbb{Z}[\theta] \to \mathbb{Z}/q\mathbb{Z}$ with $\ker \varphi = \mathfrak{q}$ be the epimorphism defined in Proposition 8, which maps θ to $s \pmod q$. Then $\mathfrak{q}$ is a first degree prime ideal in bijective correspondence with (q, s). Restrict φ to elements in $\mathbb{Z}[\theta]$ not in $\mathfrak{q}$, such that the composition $\chi : \mathbb{Z}[\theta] - \mathfrak{q} \to \{\pm 1\}$ defined as $\chi(\alpha) = \left(\frac{\varphi(\alpha)}{q}\right)$ is well-defined. We have $f'(\theta)^2 \prod_{(a,b)\in S_2}(a + b\theta) = \gamma^2$ for some $\gamma \in \mathbb{Z}[\theta]$. Then $\varphi(\gamma^2) \equiv f'(s)^2 \prod_{(a,b)\in S_2}(a + bs) \not\equiv 0 \pmod q$, by assumption. Hence, $\chi(\gamma^2) = \left(\frac{\varphi(\gamma^2)}{q}\right) = \left(\frac{\varphi(\gamma)\varphi(\gamma)}{q}\right) = \left(\frac{\varphi(\gamma)}{q}\right)^2 = 1$. Also $\chi(f'(\theta)^2) = 1$. By taking χ on both sides of our assumption it follows that $\prod_{(a,b)\in S_2}\left(\frac{a+bs}{q}\right) = 1$. □

The proposition states another necessary condition for squareness, not a sufficient one. But if we have k odd primes that do not divide $N(a + b\theta)$ for any $(a, b) \in S_2$ and $s_i \in R(q_i)$, for $1 \le i \le k$, where $f'(s_i) \not\equiv 0 \pmod{q_i}$, then with high probability (4.5) and (4.8) will imply that

$$\prod_{(a,b)\in S_2} (a + b\theta) = \gamma^2 \text{ for some } \gamma \in \mathcal{O}$$

It is conjectured that $k = \lfloor 3 \ln n \rfloor$ primes chosen as small as possible will be enough.

If we now make the associated exponent vectors of the factorization of the smooth $N(a + b\theta)$'s and include the quadratic characters it will be sufficient to construct a square [4, 11].

To do so the multiplicative Lagrange symbol group $\{\pm 1\}$ is changed into an additive group over $\mathbb{Z}_2$ by

$$\left(\frac{a+bs_i}{q_i}\right) = \begin{cases} 1 & \text{enter } 0 \\ -1 & \text{enter } 1 \end{cases}$$

We name the base of the first degree primes ideals $\mathfrak{q}$ the quadratic character base $\mathcal{Q}$ and the function χ corresponding to each $\mathfrak{q}$ the quadratic character [1].

To sum up the algebraic side, we want to find (4.7). From the sieving we have a set T_2 of B-smooth elements (a, b) over the factor base $\mathcal{A}$. This means that the norm $N(a + b\theta)$ factors completely over the primes occurring in the (p, r) pairs corresponding to the first degree prime ideals in the algebraic factor base.

We select a quadratic character base $\mathcal{Q}$ with a finite number of first degree prime ideals $\mathfrak{q}$ where the corresponding pairs (q, s) satisfy Proposition 11.

When we have more (a, b) pairs than the number of first degree prime ideals in the algebraic factor base and in the quadratic factor base, it is possible to find a subset $S_2 \subset T_2$ such that (4.4) is satisfied for all $\mathfrak{p} \in \mathcal{A}$ and (4.8) for all $\mathfrak{q} \in \mathcal{Q}$. Since satisfying (4.4) and (4.8) is sufficient, we will find a subset satisfying (4.7).

The Linear System

In the above we outlined the rational and the algebraic use of the smooth numbers separately, but the linear algebra is performed on the set T. We have

$$T = T_1 \cap T_2 = \{(a, b)|(a, b) \in \mathcal{U}, a + bm, N(a + b\theta) \text{ is } B\text{-smooth}\}$$

From T we locate the corresponding exponent vectors and they are divided into three parts of respectively $y + 1 = \pi(B) + 1, y' = |\mathcal{A}|$ and k elements as follows.

The rational factor base $\mathcal{B}$ consists of y primes $p_1, p_2, \ldots, p_y$, all below the limit B. The algebraic factor base the y' pairs (p_i, r_i), with $p_i \in \mathcal{B}$ and $r_i \in R(p_i)$, $1 \leq i \leq y$. And the quadratic character base is the first k pairs (q_i, s_i) such that $q_i > B$ and $s_i \in R(q_i)$ with $f'(s_i) \not\equiv 0 \pmod{q_i}$, $1 \leq i \leq k$.

Define

$$\widehat{\mathbf{v}} : T \to \mathbb{Z}_2^{1+y+y'+k}$$

Let $(a, b) \in T$. The first coordinate of $\widehat{\mathbf{v}}(a, b)$ is the sign of $a + bm$. It is 1 if $a + bm < 0$ and 0 if $a + bm > 0$. We found in Section 4.2 that $a + bm = \prod_{i=1}^{y} p_i^{e_i}$, with $p_i \in \mathcal{B}$, $e_i \in \mathbb{Z}$, $1 \leq i \leq y$, and the associated exponent vector gives the next y coordinates of $\widehat{\mathbf{v}}$ as $\mathbf{v}(a + bm) = (e_1, e_2, \ldots, e_y)$.

The next y' coordinates of $\widehat{\mathbf{v}}$ are given by $e_{p,r}(a + b\theta) \bmod 2$ as (p, r) runs through $\mathcal{A}$, as outlined in Section 4.2.

The last k coordinates of $\widehat{\mathbf{v}}$ are determined by $\left(\frac{a+bs}{q}\right)$ as (q, s) goes through $\mathcal{Q}$. It is 0 if $\left(\frac{a+bs}{q}\right) = 1$ and 1 if $\left(\frac{a+bs}{q}\right) = -1$.

If $|T| > 1 + y + y' + k$, then the vectors $\widehat{\mathbf{v}}$ with $(a, b) \in T$ are linearly dependent. Then there exist a subset $S \subset T$ such that $\sum_{(a,b)\in S} \widehat{\mathbf{v}}$ is the zero vector in $\mathbb{Z}_2^{1+y+y'+k}$. This set will satisfy (4.6) and (4.7) simultaneously [4]. There are various ways to solve the linear algebra step, but we will not outline any as they are out of scope for this thesis.

4.3 The Square Roots

We have $X^2 \in \mathbb{Z}$ and $\beta^2 \in \mathbb{Z}[\theta]$ such that $\varphi(\beta^2) \equiv Y^2 \pmod{n}$ and now we need the square roots to be able to calculate the $\gcd(Y \pm X, n)$.

As always, the rational side is fairly easy. We just look at the adjoint rational exponent vector to the product $X^2 \in \mathbb{Z}$. The vector gives us the factorization of X^2 and by dividing through it with 2 we obtain the prime factorization of X. Remark that we do not want to nor need to calculate the huge number X, since we seek its residue modulo n. We therefore reduce all the prime power divisors of X and then multiply them together to obtain $X \bmod n$, instead of multiplying them together and then reducing it.

There is more work to do on the algebraic side. We have the factorization of β^2 as an ideal, it is the prime ideals found in Section 3.4 together with Lemma 10. However, it is not true in general that all prime ideals have generators and computing the units is still hard. We will instead try to compute β as a polynomial combined with Hensel's lifting and the Chinese Remainder Theorem.

Let β^2 be expressed as $\sum_{i=0}^{d-1} b_i\theta^i$ for $b_i \in \mathbb{Z}$. Then for some unique $a_i \in \mathbb{Z}$, $\beta = \sum_{i=0}^{d-1} a_i\theta^i$ and by using the homomorphism (3.1) we get $\varphi(\beta) = \sum_{i=0}^{d-1} a_i m^i$.

To gain this we make a new homomorphism with a prime $\tilde{q}$, where $f(x)$ is irreducible modulo $\tilde{q}$

$$\begin{aligned} \eta_{\tilde{q}} :\ & \mathbb{Z}[\theta] \to \mathbb{Z}_{\tilde{q}}[x]/(f(x)) \\ & \sum_{i=0}^{d-1} \alpha_i\theta^i \mapsto \sum_{i=0}^{d-1} \alpha_i x^i + (f(x)) \end{aligned}$$

The prime $\tilde{q}$ lies below a prime ideal $\tilde{\mathfrak{q}}$ and $\mathbb{Z}_{\tilde{q}}[x]/(f(x)) \cong \mathbb{Z}[\theta]/\tilde{\mathfrak{q}}$.

We have $\eta_{\tilde{q}}(\beta^2) = \sum_{i=0}^{d-1} b_i x^i + (f(x))$ and call it b. We want to find $\eta_{\tilde{q}}(\beta) = \sum_{i=0}^{d-1} a_i x^i + (f(x)) = a$, so the square root of b must be calculated. Luckily, it is easy to calculate it in the finite field $\mathbb{Z}_{\tilde{q}}[x]/(f(x))$ using for instance the Tonelli-Shanks algorithm. We express the root as

$$\sqrt{b} = c = \sum_{i=0}^{d-1} c_i x^i + (f(x))$$

The algorithm gives $c \equiv \pm a$, that is, $c_i \equiv \pm a_i \pmod{\tilde{q}}$. We now use Hensel's Lemma which is a construction method that allows us to "lift" a solution mod $\tilde{q}$ to a solution mod $\tilde{q}^2$ and so on. We give an example before stating the theorem.

We have $g(x) = x^2 - e$ and x_1 such that $x_1^2 \equiv e \pmod{\tilde{q}}$. Let $x_2 \equiv (x_1^2+e)/2x_1 \pmod{\tilde{q}^2}$. This particular x_2 is located with Newton's method as follows

$$x_2 = x_1 - \frac{g(x_1)}{g'(x_1)} = x_1 - \frac{x_1^2 - e}{2x_1} = \frac{x_1^2 + e}{2x_1}$$

Then $x_2^2 \equiv e \pmod{\tilde{q}^2}$ and hence x_2 is a solution to $g(x) \bmod \tilde{q}^2$.

Theorem 12 (Hensel's Lemma)**.** *Let $\tilde{q}$ be a prime, $g(x) \in \mathbb{Z}[x]$ with $x_1 \in \mathbb{Z}$ such that $g(x_1) \equiv 0 \pmod{\tilde{q}^k}$ for some k and $g'(x_1) \not\equiv 0 \pmod{\tilde{q}}$. Then there is a unique solution $x_2 \bmod \tilde{q}^{2k}$ such that $g(x_2) \equiv 0 \pmod{\tilde{q}^{2k}}$ and $x_2 \equiv x_1 \pmod{\tilde{q}^k}$, given as $x_2 \equiv x_1 - \frac{g(x_1)}{g'(x_1)} \pmod{\tilde{q}^{2k}}$.*

Proof. $x_2 = x_1 + t\tilde{q}^k$, for some t, since $x_2 \equiv x_1 \pmod{\tilde{q}^k}$. We get

$$\begin{aligned} g(x_2) = g(x_1 + t\tilde{q}^k) &= g(x_1) + g'(x_1)t\tilde{q}^k + \text{higher terms} \\ &\equiv g(x_1) + g'(x_1)t\tilde{q}^k \bmod \tilde{q}^{2k} \end{aligned}$$

The first line is the Taylor series of $g(x)$ around x_1 or simply the finite expansion and regrouping of terms around x_1, since $g(x)$ is a polynomial.

For $g(x_2) \equiv 0 \pmod{\tilde{q}^{2k}}$ to hold, we need $g(x_1)+g'(x_1)t\tilde{q}^k \equiv 0 \pmod{\tilde{q}^{2k}}$. We are given $g(x_1) = s\tilde{q}^k$, for some s, so we get $(s + g'(x_1)t)\tilde{q}^k \equiv 0 \pmod{\tilde{q}^{2k}}$, which is equivalent to $s + g'(x_1)t \equiv 0 \pmod{\tilde{q}^k}$. Since $g'(x_1) \not\equiv 0 \pmod{\tilde{q}}$ it has an inverse in $\mathbb{Z}_{\tilde{q}}$, and $t \equiv -sg'(x_1)^{-1} \pmod{\tilde{q}^k}$. This t is uniquely given in $[1, \tilde{q}^k - 1]$ and we get the unique $x_2 \bmod \tilde{q}^{2k}$.

Now $g(x_1)+g'(x_1)t\tilde{q}^k \equiv 0 \pmod{\tilde{q}^{2k}}$ give $-t\tilde{q}^k \equiv g(x_1)/g'(x_1)$ and by rephrasing $x_2 = x_1 + t\tilde{q}^k$, $x_2 \equiv x_1 - \frac{g(x_1)}{g'(x_1)} \pmod{\tilde{q}^{2k}}$ follows. □

We could now choose one of the square roots and lift it to a square root mod $\tilde{q}^2, \tilde{q}^4, \tilde{q}^8 \dots$ until the modulo are above an estimate of the coefficients a_i of $\eta_{\tilde{q}}(\beta)$. However, the size of the last modulo will be extremely large and in a worst-case scenario it will use as much time as the rest of the number field sieve. The solution is that we want the answer modulo n.

Locate j different primes $\tilde{q}_i$ such that $f(x)$ is irreducible mod $\tilde{q}_i, \forall i$, and calculate the square root of β^2 in the various fields $\mathbb{Z}_{\tilde{q}_i}[x]/(f(x))$, for all $\tilde{q}_i$. Now we choose one of the square roots and lift it to a square root mod $m_i, \forall i$, where $m_i = \tilde{q}^{k_i}$ for suitable k_i's for every $\tilde{q}_i$ [2]. We name the different square roots $\beta_i \bmod m_i$. The various square roots will be much smaller than the single one above. If we assume we picked the right square root for all i, we get

$$\begin{aligned} \beta &\equiv \beta_1 \pmod{m_1} \\ \beta &\equiv \beta_2 \pmod{m_2} \\ &\vdots \\ \beta &\equiv \beta_j \pmod{m_j} \end{aligned}$$

If we have the product $m_1 m_2 \dots m_j$ bigger than the a_i estimate, we obtain the unique solution β by the Chinese Remainder Theorem.

The sign of the different square roots should be the same as the sign of β. If the degree of $f(x)$ is odd, we can use $N(-\alpha) = -N(\alpha)$ to test all the β_i's, since $N(-\alpha) = \sigma_1(-\alpha) \dots \sigma_d(-\alpha) = (-1)^d \sigma_1(\alpha) \dots \sigma_d(\alpha) = -N(\alpha)$.

Now we use φ to obtain the solution in $\mathbb{Z}_n$, $\varphi(\beta) \equiv Y \pmod{n}$, and hopefully $\gcd(Y \pm X, n)$ is a nontrivial factor of n.

4.4 The Algorithm

In summary, we have n, an odd integer which is not a power of a prime integer, and we want to split it.

1. Choose m and d. Find a monic polynomial $f(x) \in \mathbb{Z}[x]$ of degree d such that $f(m) \equiv 0 \pmod{n}$ and find a root θ of $f(x)$. Test to see if $f(x)$ is reducible. If so, return the splitting and we are done.

2. Compute $\gcd(f'(m), n)$. If this is a nontrivial factor, we are done.

3. For $\forall p \leq B$, calculate $R(p) = \{\forall r \in \mathbb{Z}_p | f(r) \equiv 0 \pmod{p}\}$.

4. Locate the first k pairs (q_i, s_i) such that $q_i > B$, $s_i \in R(q_i)$ and $f'(s_i) \not\equiv 0 \pmod{q_i}$, for all $i \in [1, \ldots, k]$.

5. Find $T = \{(a,b)|a, b \in \mathbb{Z}, \gcd(a,b) = 1, |a|, b < U, a{+}bm, N(a{+}b\theta)\ B\text{-smooth}\}$ using the sieves described in Section 3.4.

6. Make the matrix consisting of the vectors $\widehat{\mathbf{v}}(a,b)$, $\forall(a,b) \in T$ as outlined in Section 4.2.

7. Find a nonempty subset $S \subset T$ such that $\sum_S \widehat{\mathbf{v}}(a,b) \equiv 0 \pmod{2}$, using linear algebra.

8. Compute the residue X mod n from $X^2 = f'(m)^2 \prod_{(a,b)\in S}(a + bm)$.

9. From $f'(\theta)^2 \prod_{(a,b)\in S}(a + b\theta)$, find a square root $\beta = \sum_{i=0}^{d-1} a_i\theta^i$ as described in Section 4.3. Compute $\varphi(\beta) \equiv Y \pmod{n}$.

10. Calculate $\gcd(X \pm Y, n) = D$.

If D is trivial, go back to the linear algebra and find another linear dependencie in the matrix. If no such dependencie is located, expand the boundaries and sieve until more smooth values are found.

The boundaries and the overall complexity of the NFS are discussed in Chapter 6.

CHAPTER

FIVE

THE NUMBER FIELD SIEVE AND THE DISCRETE LOGARITHM PROBLEM

We will now explore an algorithm to solve the discrete logarithm problem using the number field sieve (NFS-dlog). As the name reveals, it derives from the number field sieve algorithm for splitting composite numbers. It is also a great modification of the index-calculus method, with crossover point at approximately 66-digit primes.

There are given different variations of the NFS-dlog with the desired running time and some of these are more heavily based on the ICM. Daniel M. Gordon gives one in [3], where he calculates the discrete logarithms of the primes in the factor base, then finds the discrete logarithms of "medium-sized" primes and combines these into the discrete logarithm. Oliver Schirokauer improved Gordon's algorithm and we will mainly follow his presentation given in [12, 13]. We also found some inspiration in [16].

A recent record using the number field sieve to compute the discrete logarithm in a prime field is held by Thorsten Kleinjung who in 2007 managed to compute a discrete logarithm in a prime field where the prime had 160 digits [12]. As the record proclaims, the development in the discrete logarithm algorithm lacks behind the improvements done in the factorization algorithms.

So, we are given $\mathbb{F}_q$, a field of q elements, q prime. We also have two elements $g, t \in \mathbb{F}_q^*$ such that t is in the subgroup of $\mathbb{F}_q^*$ generated by g. The discrete logarithm problem of t with respect to the generator g is finding the least nonnegative integer z such that $g^z \equiv t \pmod{q}$. It is written as $z \equiv \log_g t \bmod q-1$.

The rest of this chapter is as follows. The next section contains a strategy for solving the problem and an overview on how the algorithm works. Then in Section 5.2 we see the smooth numbers in action, before we build the linear equations in Section 5.3 and solves the problem. In Section 5.4 we study some complications concerning smooth generator, smooth t, ramified primes and prime powers and lastly, in Section 5.5 we present the algorithm.

5.1 Strategy

Given the problem in the introduction, we could go rock hard and try to directly locate the discrete logarithm modulo the order of the group. However, several problems would arise, the most severe of them is that the complexity would collapse due to the sizes of the estimates and numbers used. So a bit more clever approach is preferred. We will instead find the discrete logarithm modulo primes l, where $l|(q-1)$, and glue the solutions together to obtain the $z \equiv \log_g t \bmod q-1$.

We start by choosing a polynomial $f(x)$ of degree d such that $f(m) \equiv 0 \pmod{q}$ and construct the number ring $\mathbb{Z}[\theta]$ of a root $\theta \in \mathbb{C}$ of f as described in Chapter 3. We want f to be a monic, irreducible polynomial. We repeat the idea, let $m = \lfloor q^{1/d} \rfloor$ and write q in base m with $c_i \in \mathbb{Z}$ such that

$$q = m^d + c_{d-1}m^{d-1} + \ldots + c_0$$

The demands will most likely be satisfied by the polynomial $f(x) = x^d + \ldots + c_0$, otherwise one of its irreducible polynomial factors having m as a root mod q will work. An estimate of the degree d is presented in Chapter 6. We also want the prime divisors of $q-1$ to be unramified in $\mathbb{Q}(\theta)$. This is discussed in Section 5.2.

The polynomial defines the number ring $\mathbb{Z}[\theta]$ and the homomorphism (3.1) as

$$\begin{aligned} \varphi : \ \mathbb{Z}[\theta] &\to \mathbb{Z}_q \\ \theta &\mapsto m \pmod{q} \end{aligned}$$

Together with the norm map $N : \mathbb{Z}[\theta] \to \mathbb{Z}$ from Section 3.3, we have our setup.

To find the discrete logarithm modulo l we combine the smooth elements into a multiplicative relation of lth powers. That is, obtain lth powers $\omega \in \mathbb{Z}[\theta]$ and $\tau \in \mathbb{Z}$ with the property that $\varphi(\tau) = g^{y_l} t \varphi(\omega)$ for some y_l. The sieve defined in Section 3.4 will help us with the task. We construct a set T of B-smooth integers $a + bm \in \mathbb{Z}$ and $a + b\theta \in \mathbb{Z}[\theta]$ and from it we locate a subset S such that

$$\tau = g^{y_l} t \prod_{(a,b)\in S} (a+bm)^{y_{(a,b)}} \text{ is an } l\text{th power in } \mathbb{Z} \tag{5.1}$$

$$\omega = \prod_{(a,b)\in S} (a+b\theta)^{y_{(a,b)}} \text{ is an } l\text{th power in } \mathbb{Z}[\theta] \tag{5.2}$$

To be able to include the g and t in (5.1), they must have smooth preimages. This small complication is discussed in Section 5.4.

When the set S is found, we get the equation

$$\varphi(\tau) = g^{y_l} t \varphi(\omega)$$

Since both τ and ω are lth powers, we get that $g^{y_l} t$ also is an lth power in $\mathbb{F}_q^*$ and hence $y_l \equiv -\log_g t \pmod{l}$. Lastly, combining the different l solutions via the Chinese Remainder Theorem will reveal the $z \equiv \log_g t \pmod{q-1}$.

5.2 Smooth Numbers in Action

In Section 3.4 we gave a description on how to locate a set $T = T_1 \cap T_2$ of coprime integers (a, b) such that both $a+bm$ and $a+b\theta$ are B-smooth. To locate the subset S such that (5.1) and (5.2) are satisfied, we use a similar method as we used in the NFS. We begin the presentation with a brief repetition of the setup for the sieve.

The smoothness bound B provides two factor bases, the rational and the algebraic factor base

$$\mathcal{B} = \{p | p \leq B, p \in \mathbb{Z}, p \text{ prime}\}$$
$$\mathcal{A} = \{(p, r) | p \in \mathcal{B}, r \in R(p)\}$$

The set $R(p)$ we defined in Section 3.4 as all pth roots of $f(x)$. We have the universe $\mathcal{U} = \{(a, b) | a, b \in \mathbb{Z}, \gcd(a, b) = 1, |a| \leq U, 0 < b \leq U\}$, where we sieve with the two factor bases to locate the set

$$T = \{(a, b) | (a, b) \in \mathcal{U}, a + bm, N(a + b\theta) \text{ is B-smooth}\}$$

Both the parameters B and U are discussed and estimated in Chapter 6.

The fact that we are able to use a sieve to detect the smooth values in the NFS-dlog is one of the reasons why the algorithm is faster than other methods.

Recall that at this stage the strategy in the NFS is to do linear algebra on the exponent vectors corresponding to the elements in T together with the quadratic characters to locate a subset S with the squared property. Now seeking the lth powers, we will see that the rational side is again rather straightforward. The work is on the algebraic side, because it is not sufficient to know that the norm of an algebraic element is an lth power, we need to know that the element itself is an lth power. Sounds familiar? It was Oliver Schirokauer who in his paper [13] came with the major theoretical breakthrough of the following maps to replace the quadratic characters in the NFS.

The Character Maps

Let $\Gamma = \{\omega \in \mathcal{O} | N(\omega) \not\equiv 0 \pmod{l}\}$ be a multiplicative subgroup of $\mathcal{O}$ and find the least integer ε such that for all ω in Γ we have $\omega^{\varepsilon} \equiv 1 \pmod{l}$. We claim that the optimal ε is the least common multiple of the orders $|(\mathcal{O}/\ell)^*|$, where ℓ are all prime ideals above l. We will prove the statement in a moment, but recall that we wanted l to be unramified.

Definition 8. *We have a finite extension $\mathbb{Q}(\theta)/\mathbb{Q}$. A prime ideal $\langle l \rangle \in \mathbb{Z}$ generates the ideal $l\mathcal{O}$ of $\mathcal{O}$ and it has a unique representation*

$$l\mathcal{O} = \prod_i \ell_i^{f_i}$$

The ℓ_i's are distinct prime ideals of $\mathcal{O}$ above l and the powers f_i are called the ramification indices. If $f_i = 1, \forall i$, then the prime l is said to unramify *in $\mathcal{O}$.*

Lemma 13. *Let l be unramified in $\mathcal{O}$. The least nonnegative integer ε such that for every element $\omega \in \Gamma$ we have that $\omega^\varepsilon \equiv 1 \bmod l$ is*

$$\varepsilon = \mathrm{lcm}\{|(\mathcal{O}/\ell)^*| | \ell \textit{ all prime ideals dividing } \langle l \rangle\}$$

Proof. Let $\ell_1, \ell_2, \dots, \ell_k$ denote the k prime ideals dividing $\langle l \rangle$.

The ε will be divisible by $|(\mathcal{O}/\ell_1)^*|$, because then $\varepsilon = k_1|(\mathcal{O}/\ell_1)^*|$ for some integer k_1 and for all $\omega + \ell_1 \in \mathcal{O}/\ell_1$ we get $(\omega + \ell_1)^\varepsilon = 1 + \ell_1 \in \mathcal{O}/\ell_1$.

Hence ε is divisible by every $|(\mathcal{O}/\ell_i)^*|$, because for all elements ω in $\mathcal{O}$ and for all ℓ_i, we get that $\omega^\varepsilon = 1 + \omega'$, where $\omega' \in \ell_i, \forall i$. Since l is unramified, we can write the ideal as $\langle l \rangle = \prod_{i=1}^k \ell_i = \bigcap_i \ell_i$. But if $\omega' \in \ell_i, \forall i$, then $\omega' \in \langle l \rangle$ and hence $\omega^\varepsilon \equiv 1 \pmod{l}$. The least integer dividing all the orders is precisely the least common multiple. □

If l ramified in $\mathcal{O}$, we could not find such an ε. How do we test whether l ramifies or not?

A prime l ramifies if it divides the discriminant. To calculate the discriminant of $\mathbb{Q}(\theta)$ we use a basis for $\mathcal{O}$ as a $\mathbb{Z}$-module, say $\{a_1, a_2, \dots, a_d\}$, and the d embeddings from $\mathbb{Q}(\theta)$ into $\mathbb{C}$ defined in Theorem 8. The discriminant will be the square of the determinant of the matrix given as

$$\Delta_{\mathbb{Q}(\theta)} = \det \begin{bmatrix} \sigma_1(a_1) & \dots & \sigma_1(a_d) \\ \sigma_2(a_1) & \dots & \sigma_2(a_d) \\ \vdots & \ddots & \vdots \\ \sigma_d(a_1) & \dots & \sigma_d(a_d) \end{bmatrix}^2$$

As the determinant, the discriminant is invariant to the basis for $\mathcal{O}$ as a $\mathbb{Z}$-module.

Say we found that l is uramified and have located ε. Now define

$$\begin{aligned} \lambda : \Gamma &\to l\mathcal{O}/l^2\mathcal{O} \\ \omega &\mapsto \omega^\varepsilon - 1 + l^2\mathcal{O} \end{aligned}$$

The mapping λ is a well-defined homomorphism since

$$\begin{aligned} \lambda(\omega\widehat{\omega}) &= (\omega\widehat{\omega})^\varepsilon - 1 + l^2\mathcal{O} \\ &= (\omega^\varepsilon - 1) + (\widehat{\omega}^\varepsilon - 1) + (\omega^\varepsilon - 1)(\widehat{\omega}^\varepsilon - 1) + l^2\mathcal{O} \\ &= \lambda(\omega) + \lambda(\widehat{\omega}) \end{aligned}$$

$(\omega^\varepsilon - 1)(\widehat{\omega}^\varepsilon - 1)$ will lie in $l^2\mathcal{O}$ since $\omega^\varepsilon - 1 = lk'$ and $\widehat{\omega}^\varepsilon - 1 = lk''$, for some $k', k'' \in \mathcal{O}$, so $(\omega^\varepsilon - 1)(\widehat{\omega}^\varepsilon - 1) = l^2k'k'' \in l^2\mathcal{O}$.

The group $l\mathcal{O}/l^2\mathcal{O}$ is a $\mathbb{Z}$-module and $l\mathbb{Z}$ lies in $\mathrm{Ann}(l\mathcal{O}/l^2\mathcal{O})$, so $l\mathcal{O}/l^2\mathcal{O}$ is a $\mathbb{Z}/l\mathbb{Z}$-module. If we now represent $l\mathcal{O}$ as a direct sum $l\mathbb{Z} \oplus l\mathbb{Z} \dots \oplus l\mathbb{Z}$ of d copies of $l\mathbb{Z}$ and the same for $l^2\mathcal{O}$ as the direct sum of d copies of $l^2\mathbb{Z}$, we see that $l\mathcal{O}/l^2\mathcal{O} \cong l\mathbb{Z}/l^2\mathbb{Z} \oplus \dots \oplus l\mathbb{Z}/l^2\mathbb{Z}$. Hence $l\mathcal{O}/l^2\mathcal{O}$ is a free $\mathbb{Z}/l\mathbb{Z}$-module of rank d and we can make the basis $\{b_i l + l^2\mathcal{O}\}_{i=1}^d$ for $l\mathcal{O}/l^2\mathcal{O}$ over $\mathbb{Z}/l\mathbb{Z}$.

We have the following important theorem from Galois theory.

Theorem 14. *Let $l\mathcal{O}/l^2\mathcal{O}$ be a free $\mathbb{Z}/l\mathbb{Z}$-module with rank d. Then*

$$l\mathcal{O}/l^2\mathcal{O} \cong (\mathbb{Z}/l\mathbb{Z})^d$$

The two ways to represent $l\mathcal{O}/l^2\mathcal{O}$ by are isomorphic. We see it easily with the map $\eta : l\mathbb{Z} \to \mathbb{Z}/l\mathbb{Z}$ sending lk to $k + l\mathbb{Z}$, for $k \in \mathbb{Z}$. The map η is onto with kernel $\{l^2k\} = l^2\mathbb{Z}$.

From the theorem we get that λ consists of d maps $\lambda_i : \Gamma \to \mathbb{Z}/l\mathbb{Z}$. They are determined by the congruence

$$\omega^{\varepsilon} - 1 \equiv \sum_{i=1}^{d} \lambda_i(\omega) b_i l \bmod l^2$$

The λ_i's are well-defined homomorphisms by a similar argument as for λ. We call them the character maps and they will help us construct lth powers in the number ring $\mathbb{Z}[\theta]$, because any lth power is mapped to 0 by the λ_i's. How?

Let ω be an lth power in $\mathbb{Z}[\theta]$. Then there exists a $\bar{\omega} \in \mathbb{Z}[\theta]$ such that $\omega = \bar{\omega}^l$. We know $\lambda(\omega) \equiv \sum_{i=1}^{d} \lambda_i(\omega) b_i l \bmod l^2$. Also $\lambda(\omega) = \lambda(\bar{\omega}^l) = l\lambda(\bar{\omega}) \equiv 0$. Hence all $\lambda_i(\omega) = 0, \forall i$, when ω is an lth power.

The Linear System

We have all the tools to make the corresponding exponent vectors to the elements in T. As with the NFS, the vectors have three parts consisting of $y = \pi(B)$, $y' = |\mathcal{A}|$ and d elements defined as follows.

The rational factor base $\mathcal{B}$ consists of the y primes, $p_1, p_2, \ldots, p_y$ and the algebraic factor base $\mathcal{A}$ the y' pairs (p_i, r_i), where $p_i \in \mathcal{B}$ and $r_i \in R(p_i)$, $0 \leq i \leq y$, and the last d values we defined in the section above as the character maps λ_i.

For $(a, b) \in T$, we have

$$a + bm = \prod_{i=1}^{y} p_i^{e_i}$$
$$N(a + b\theta) = \prod_{(p,r)\in\mathcal{A}} p^{e_{p,r}(a+b\theta)}$$

The function $e_{p,r}$ was defined in (4.3).

The corresponding exponent vector $\widehat{\mathbf{v}}(a, b)$ consist of the values $(e_1, e_2, \ldots, e_y)$ as the first y coordinates, then the next y' coordinates are given by $e_{p,r}(a + b\theta)$ as (p, r) runs through $\mathcal{A}$. The last d values are $(\lambda_1(a+b\theta), \lambda_2(a+b\theta), \ldots, \lambda_d(a+b\theta))$.

If $|T| > y + y' + d$, we get more rows than columns in the matrix consisting of the vectors just defined and could use linear algebra to find the subset S of lth powers. We formalize the statement in the next section.

5.3 Constructing The Solution

Every pair in T has a fitting exponent vector of length $y + y' + d$. In Section 5.1 we urged the need for smooth preimages of t and g. Assume there exists some $k_g, k_t \in \mathbb{Z}$ such that

$$g + k_g q = \prod_{i=1}^{y} p_i^{g_i} \in \mathbb{Z} \tag{5.3}$$

$$t + k_t q = \prod_{i=1}^{y} p_i^{t_i} \in \mathbb{Z} \tag{5.4}$$

The corresponding vector $\mathbf{v}(g)$ has $(g_1, g_2, \ldots, g_y)$ as first y coordinates and zero on the last $y' + d$ places. The $\mathbf{v}(t)$ is defined in the same way.

We make the system $Y_l^{\mathsf{T}} A \equiv -\mathbf{v}(t) \pmod{l}$ where

$$A = \begin{bmatrix} \mathbf{v}(g) \\ \widehat{\mathbf{v}}(a_1, b_1) \\ \widehat{\mathbf{v}}(a_2, b_2) \\ \vdots \\ \widehat{\mathbf{v}}(a_{|T|}, b_{|T|}) \end{bmatrix}$$

Linear algebra solves the system and returns

$$Y_l^{\mathsf{T}} = \begin{bmatrix} y_l & y_{(a_1,b_1)} & y_{(a_2,b_2)} & \cdots & y_{(a_{|T|},b_{|T|})} \end{bmatrix}$$

Some of the entries in Y_l will be zero, and the desired $S \subset T$ is found. Actually, we have solved our problem and the solution is

$$y_l \equiv -\log_g t \pmod{l}$$

To see this, express the solution to the matrix as

$$-\mathbf{v}(t) \equiv y_l \mathbf{v}(g) + y_{a_1,b_1} \widehat{\mathbf{v}}(a_1, b_1) + \ldots + y_{a_{|T|},b_{|T|}} \widehat{\mathbf{v}}(a_{|T|}, b_{|T|}) \pmod{l}$$

Now make the corresponding τ and ω

$$\tau = g^{y_l} t \prod_{(a,b)\in S} (a + bm)^{y_{(a,b)}} \tag{5.5}$$

$$\omega = \prod_{(a,b)\in S} (a + b\theta)^{y_{(a,b)}} \tag{5.6}$$

Since g, t and $a + bm$ are all B-smooth in $\mathbb{Z}$, we can rewrite (5.5) as

$$\tau = \prod_{i=1}^{y} p_i^{e_i}$$

Clearly τ is an lth power of an element in $\mathbb{Z} \subset \mathbb{Z}[\theta]$, since $e_i \equiv 0 \pmod{l}, \forall i$.

Similarly, $e_{(p,r)} \equiv 0 \pmod{l}, \forall (p,r) \in \mathcal{A}$, and $\delta_i(\omega) \equiv 0 \pmod{l}$ for $i \in [1,d]$ in (5.6), so ω is an lth power in $\mathcal{O}$. To make ω an lth power of an element in $\mathbb{Z}[\theta]$, we revisit Lemma 11 and see that $f'(\theta)^l\omega$ and $f'(\theta)^l\tau$ are lth powers of some elements in $\mathbb{Z}[\theta]$.

Finally, using (3.1) and that $\varphi(m) = \varphi(\theta)$ we get

$$\begin{aligned}
\varphi\left(f'(\theta)^l\tau\right) &= \varphi\left(f'(\theta)^l g^{y_l} t \prod_{(a,b)\in S} (a+bm)^{y_{(a,b)}}\right) \\
&= g^{y_l} t \varphi\left(f'(\theta)^l \prod_{(a,b)\in S} (a+b\theta)^{y_{(a,b)}}\right) \\
&= g^{y_l} t \varphi\left(f'(\theta)^l\omega\right)
\end{aligned}$$

Hence we find that $g^{y_l}t$ is an lth power in $\mathbb{Z}_q$ as $\varphi(f'(\theta)^l\tau)$ and $\varphi(f'(\theta)^l\omega)$ are and the solution is $y_l \equiv -\log_g t \pmod{l}$.

After repeating for all prime divisors of $(q-1)$, the Chinese Remainder Theorem will glue the solutions modulo the different prime divisors together and we find the $z \equiv \log_g t \pmod{q-1}$.

5.4 Some Complications

Smoothness

We require g and t to be smooth in $\mathbb{Z}$ to be able to include them in (5.5) and (5.6). It will be sufficient to know two elements $\mathfrak{g}$ and $\mathfrak{t}$ being the preimage of g and t in $\mathbb{Z}[\theta]$ such that the ideal they generates $\langle\mathfrak{g}\rangle$ and $\langle\mathfrak{t}\rangle$ factorizes completely over the algebraic factor base $\mathcal{A}$. Then we define the vectors $\mathbf{v}(\mathfrak{g})$ and $\mathbf{v}(\mathfrak{t})$ to contain the exponents of the prime ideals in the factorization and the values $\lambda_i(\mathfrak{g})$ and $\lambda_i(\mathfrak{t})$ replace $\mathbf{v}(g)$ and $\mathbf{v}(t)$ in $Y_l^{\mathsf{T}} A \equiv -\mathbf{v}(t) \pmod{l}$. Then linear algebra would produce the lth powers

$$\begin{aligned}\tau =& \prod_{(a,b)\in S} (a+bm)^{y_{(a,b)}} \\ \omega =& \mathfrak{g}^{y_l}\mathfrak{t} \prod_{(a,b)\in S} (a+b\theta)^{y_{(a,b)}}\end{aligned}$$

The solution $y_l \equiv -\log_g t \pmod{l}$ would follow.

If we cannot find smooth preimages of g in neither $\mathbb{Z}$ nor $\mathbb{Z}[\theta]$, there is a third solution. If we manage to find another generator g' with a smooth preimage in either $\mathbb{Z}$ or $\mathbb{Z}[\theta]$, the logarithm of t is easily located by

$$\log_g t \equiv \frac{\log_{g'} t}{\log_{g'} g} \bmod (q-1) \tag{5.7}$$

There are other ways to get around the difficulty of smooth preimages of t and g, see [12] for more details.

Ramification

If l ramified in $\mathcal{O}$, Lemma 13 cannot be used and the character maps fails. Then we need a different map to do the same work [13]. Let $\mathfrak{l} = \prod_i \ell_i$, where ℓ_i are all prime ideals of $\mathcal{O}$ above l. Then

$$\omega^{\epsilon} \equiv 1 \pmod{\mathfrak{l}}$$

The character map λ can now be replaced by

$$\begin{aligned}\Lambda : \Gamma &\to \mathfrak{l}/l\mathfrak{l} \\ \omega &\mapsto \omega^{\varepsilon} - 1 + l\mathfrak{l}\end{aligned}$$

The rest of the steps in the algorithm must also be modified to fit the new map. The complication is minor, since it is only the few primes dividing the determinant which ramifies.

The Primes Dividing the Order

There are two issues concerning the primes dividing the order. The small primes and the larger prime powers.

We can use the NFS or other methods to get that $q-1=\prod_i l_i^{e_{l_i}}$, some $e_{l_i}\in\mathbb{Z}$ and l_i primes.

Now, we know that the NFS is the optimal choice to split n when n is big, say more than 110 digits, and that the algorithm is too extravagant for smaller n's and then other methods should be used. The same rule holds for the NFS-dlog. The algorithm is to cumbersome for small numbers. Hence, other methods should be applied, for instance the already discussed ICM. As mentioned in the introduction the crossover point between the NFS-dlog and the ICM is at approximately 66-digit primes. There exists a lot of methods to solve the discrete logarithm modulo small primes, but in this thesis we have only cared for the large primes.

The NFS-dlog will be the best choice for the big primes and one should proceed as we have outlined in this chapter. However, there is extra work for the prime powers, that is, if $l^{e_l}|(q-1)$ for $e_l>1$ and l a large prime.

The first issue is the character maps. In Section 5.2 we defined the group $\Gamma=\{\omega\in\ \mathcal{O}|N(\omega)\not\equiv 0 \pmod l\}$ and the map

$$\begin{aligned}\lambda:\Gamma&\to l\mathcal{O}/l^2\mathcal{O}\\ \omega&\mapsto\omega^{\varepsilon}-1+l^2\mathcal{O}\end{aligned}$$

This will only help us to determine whether ω is an lth power or not. Now seeking the l^{e_l}th power, further use of the λ is required. Let $\Gamma=\Gamma_1$ and $\lambda=\lambda_1$. Define $\Gamma_2=\{\omega\in\Gamma_1|\lambda_1(\omega)=0\}$ and the map

$$\begin{aligned}\lambda_2:\Gamma_2&\to l^2\mathcal{O}/l^{2^2}\mathcal{O}\\ \omega&\mapsto\omega^{\varepsilon}-1+l^{2^2}\mathcal{O}\end{aligned}$$

With the same argument as in Section 5.2 we get that $l^2\mathcal{O}/l^{2^2}\mathcal{O}$ is a free $\mathbb{Z}/l^2\mathbb{Z}$-module of rank d and can make the basis $\{b_i'l^2+l^{2^2}\mathcal{O}\}_{i=1}^d$. We use Theorem 14 and get that λ_2 is given by the d maps $\lambda_{2,i}:\Gamma_2\to\mathbb{Z}/l^2\mathbb{Z}$. They are determined by the congruence

$$\omega^{\varepsilon}-1\equiv\sum_{i=1}^{d}\lambda_{2,i}(\omega)b_i'l^2 \bmod l^{2^2}$$

So λ_2 helps us decide which values are l^2th powers. The next map is given as $\lambda_3:\Gamma_3\to l^{2^2}\mathcal{O}/l^{2^3}\mathcal{O}$ with $\Gamma_3=\{\omega\in\Gamma_2|\lambda_2(\omega)=0\}$ and so on. We repeat until we have reached a power of l greater than e_l. Since l is a large prime divisor of the order of the group, e_l will most likely not be a big integer and we will only need a few new maps.

Now the last d $\lambda_{j,i}$'s from above are included in the exponent vectors instead of the d λ_i's from Section 5.2 and the usual linear algebra will reveal a solution

$A^{\intercal}Y_{l,1} \equiv -\mathbf{v}(t)^{\intercal} \pmod{l}$ as in Section 5.3. But again the prime powers complicates the situation, since we want to find $A^{\intercal}Y_{l,k} \equiv -\mathbf{v}(t)^{\intercal} \pmod{l^{e_l}}$, for the power e_l. So we need to lift the already found solution $Y_{l,1} \pmod{l}$ to this higher power. We begin by writing $A^{\intercal}Y_{l,1} - (-\mathbf{v}(t)^{\intercal}) = l\mathbf{v}(Y_{l,1})$ for some integer vector $\mathbf{v}(Y_{l,1})$ and solves $A^{\intercal}Y_{l,2} \equiv \mathbf{v}(Y_{l,1}) \pmod{l}$. Then it follows that $A^{\intercal}(Y_{l,1} - lY_{l,2}) \equiv -\mathbf{v}(t)^{\intercal} \pmod{l^2}$.

From the last equation we get that $A^{\intercal}(Y_{l,1} - lY_{l,2}) - (-\mathbf{v}(t)^{\intercal}) = l^2\mathbf{v}(Y_{l,2})$ for some integer vector $\mathbf{v}(Y_{l,2})$. Then there exist a $Y_{l,3}$ such that $A^{\intercal}Y_{l,3} \equiv \mathbf{v}(Y_{l,2}) \pmod{l}$ and hence $A^{\intercal}(Y_{l,1} - lY_{l,2} - l^2Y_{l,3}) \equiv -\mathbf{v}(t)^{\intercal} \pmod{l^3}$ [13].

We continue the process until the solution modulo the desired power is found.

Again, if l ramified, then $\Lambda_i : \Gamma_i \to l^{2^{i-2}}\mathfrak{l}/l^{2^{i-1}}\mathfrak{l}$ replaces $\lambda_i : \Gamma_i \to l^{2^{i-1}}\mathcal{O}/l^{2^i}\mathcal{O}$ and so on [13].

The issue with the prime powers is rather small, since in most cases the biggest primes occure once in the order and the smaller primes are handled with other effective algorithms.

Notice that if the order has a lot of small primes and only a few medium sized ones, then the discrete logarithm will be easier to find than if one took care of choosing the order properly. So if one wish to make the discrete logarithm problem as hard as possible one should make sure that $p = (q-1)/2$ is a prime.

We sum up the NFS-dlog in the next section and we do it for a single l, not considering the small complications discussed in this section.

5.5 The Algorithm

In summary, we want to find the least nonnegative z such that $g^z \equiv t \pmod{q}$

1. Find a prime l dividing the order $q-1$.
2. Choose m and d. Find a monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$ of degree d such that $f(m) \equiv 0 \pmod{q}$.
3. For $\forall p \leq B$, calculate $R(p) = \{\forall r \in \mathbb{Z}_p | f(r) \equiv 0 \pmod{p}\}$.
4. Find $T = \{(a,b) | a,b \in \mathbb{Z}, \gcd(a,b) = 1, |a|, b < U, a{+}bm, N(a{+}b\theta)\ B\text{-smooth}\}$ using the sieve described in Section 3.4.
5. Find smooth preimages of g and t.
6. Calculate the character maps given in Section 5.2 for all $a + b\theta$ in T.
7. Make the matrix A consisting of the vectors $\widehat{\mathbf{v}}(a,b)$, $\forall (a,b) \in T$.
8. Find a nonempty subset $S \subset T$ such that $Y_l^{\intercal}A \equiv -\mathbf{v}(t) \pmod{l}$, using linear algebra as explained in Section 5.3.
9. Locate the solution y_l mod l.

CHAPTER
SIX

THE ANALYSIS OF THE NFS AND THE NFS-DLOG

The topic of this chapter is an analysis of the two main algorithms, the number field sieve for factorization and the number field sieve for the discrete logarithm problem. As the analysis of the QS, it will be a rough and heuristic analysis.

We analyse the number field sieves together since their shared parts uses approximately the same amount of time. And at the end we will see that they actually have the same complexity.

Analysis

We begin with the estimate of the running time of the number field sieves and along the way we will also estimate the degree d of $f(x)$, look at the bounds B and U and locate the upper bound of the $(a+bm)N(a+b\theta)$'s. We will do it for the splitting algorithm and spesify when there are special cases for the discrete logarithm case.

The analysis will build upon the analysis of the QS from Section 2.2 since the number field sieve is an extended quadratic sieve. In the QS, we found the optimal limit for B to be approximately $L(n)^{1/2} = e^{\frac{1}{2}\sqrt{\ln n \ln \ln n}}$ and that the running time to split n was $L(n)^{1+o(1)} = e^{\sqrt{\ln n \ln \ln n}} = B^{2+o(1)}$.

Both algorithms have two main phases, the sieving and the linear algebra. They use approximately the same amount of time and one should divide the work between them to match the available software. The sieving can be parted and divided onto a lot of computers while the linear algebra needs a huge computer to manage the big matrix. This different software use should therefore make an impact on the boundaries when one wants to use the two algorithms. We will study the sieving and briefly mention the linear algebra.

Now being able to choose the polynomial and its degree will lower the running time compared to the QS. The polynomial should be chosen such that the coefficients are small, because then it is more likely that $N(a+b\theta)$ is small and hence smooth. In addition we want the polynomial to have many real roots, its Galois group should be relatively small and it should have many roots modulo small primes [14]. If we choose the polynomial randomly, it will most likely not have any of the quantities. It should therefore be spend some time on picking a polynomial. With that said, the time used to choose a good polynomial will still be small compared to the two most time-consuming parts in the algorithms.

Further in this section we will mainly follow the presentations given in [4, 11].

We will use Theorem 2 in the analysis and need therefore an estimate for the bound K of smooth elements in T. Since we want $a + bm$ and $N(a + b\theta)$ to be smooth simultaneously, we study them together.

The probability that an element below the limit K is B-smooth we get from Theorem 1 and it is still valid as $\Psi(K, B)/K$.

So, if we have a random list of integers bounded by K, how large must K be so that a subsequence returns a square? In the quadratic sieve, $K = n^{1/2+o(1)}$.

Take a random element $(a+bm)N(a+b\theta)$ from T. Recall from Section 3.4 that $N(a+b\theta)$ is equal to the homogenous form of $f(x)$. With $m \leq n^{1/d}$, all coefficients $c_i \leq n^{1/d}$ and $|a|, b \leq U$ we get an estimate for the bound of $(a + bm)N(a + b\theta)$

$$\begin{aligned}(a + bm)N(a + b\theta) &= (a + bm)F(a, b)\\ &= (a + bm)(a^d + c_{d-1}a^{d-1}b + \ldots + c_0 b^d)\\ &= a^{d+1} + c_{d-1}a^d b + \ldots + c_0 ab^d \ldots\\ &\ldots + a^d bm + c_{d-1}a^{d-1}b^2 m + \ldots + c_0 b^{d+1} m\\ &< 2(d + 1)U^{d+1} n^{2/d} \qquad (6.1)\end{aligned}$$

We call the bound $2(d + 1)U^{d+1}n^{2/d} = K$. In the NFS-dlog the coefficients are bounded by $q^{1/d}$, $m \leq q^{1/d}$ so it has the bound $2(d + 1)U^{d+1}q^{2/d}$. Anyway, from Theorem 2 we get that $L(K)^{\sqrt{2}+o(1)}$ pairs of a and b's will be sufficient. If we let the bounds B and U be equal we say that $U^2 = L(K)^{\sqrt{2}+o(1)}$. We put this assumption into (6.1), take the logarithm and get

$$\begin{aligned}K &= 2(d + 1)\left(L(K)^{1/\sqrt{2}}\right)^{d+1} n^{2/d}\\ \ln K &= \ln(2(d + 1)) + (d + 1)\sqrt{\frac{1}{2}\ln K \ln\ln K} + \frac{2}{d}\ln n \qquad (6.2)\end{aligned}$$

We will first make an observation about the degree d. Let d be fixed. The first term on the right side in (6.2) is now insignificant compared to the last term. Also $\ln K > (d + 1)\sqrt{\frac{1}{2}\ln K \ln\ln K}$, so we can simplify (6.2) into

$$\ln K \approx \frac{2}{d}\ln n$$

The running time for a fixed d is therefore

$$L(K)^{\sqrt{2}+o(1)} = L(n)^{\sqrt{4/d}+o(1)}$$

Hence, if one picks a polynomial purely based on the bounds of the smooth integers and Theorem 2, then the number field sieve will not do better than the quadratic sieve unless $d > 4$. We give a finite estimate later, but first we will finish the estimate of the bound K.

Since we are trying to find an optimal bound, let $n, d \to \infty$. Notice that d will have a much slower rate than n, so the first term in (6.2) can still be neglected, but the second term is no longer negligible

$$\ln K = (d+1)\sqrt{\frac{1}{2}\ln K \ln\ln K} + \frac{2}{d}\ln n \tag{6.3}$$

We want to find the d that minimizes the bound K in (6.3). To accomplish that we take the derivative with respect to d and get

$$0 = \sqrt{\frac{1}{2}\ln K \ln\ln K} + \frac{-2}{d^2}\ln n$$

Hence

$$d = \frac{\sqrt{2\ln n}}{(\frac{1}{2}\ln K \ln\ln K)^{1/4}} \tag{6.4}$$

Replace d in (6.3) with (6.4)

$$\begin{aligned}
\ln K &= \frac{2(\frac{1}{2}\ln K \ln\ln K)^{1/4}}{(2\ln n)^{1/2}}\ln n + \frac{(2\ln n)^{1/2}}{(\frac{1}{2}\ln n)^{1/4}}\sqrt{\frac{1}{2}\ln K \ln\ln K} \\
&= \frac{2}{\sqrt{2}}(\ln n)^{1/2}\left(\frac{1}{2}\ln K \ln\ln K\right)^{1/4} + \sqrt{2}(\ln n)^{1/2}\left(\frac{1}{2}\ln K \ln\ln K\right)^{1/4} \\
&= 2\sqrt{2}(\ln n)^{1/2}\left(\frac{1}{2}\ln K \ln\ln K\right)^{1/4}
\end{aligned}$$

Collect the $\ln K$ term on one side and take the logarithm

$$\begin{aligned}
(\ln K)^{3/4} &= 2\sqrt{2}(\ln n)^{1/2}\left(\frac{1}{2}\ln\ln K\right)^{1/4} \\
\frac{3}{4}\ln\ln K &= \ln 2\sqrt{2} + \frac{1}{2}\ln\ln n + \frac{1}{4}\ln(\frac{1}{2}\ln\ln K)
\end{aligned}$$

The first and last term on the right are small compared to the term on the left side, so we say that $\frac{3}{4}\ln\ln K \approx \frac{1}{2}\ln\ln n$ and hence

$$(\ln K)^{3/4} = 2\sqrt{2}\,(\ln n)^{1/2}\left(\frac{1}{3}\ln\ln n\right)^{1/4}$$

Consequently

$$\begin{aligned}\ln K &= \left(2\sqrt{2}\right)^{4/3}(\ln n)^{2/3}\left(\frac{1}{3}\ln\ln n\right)^{1/3}\\ &= \frac{4}{3^{1/3}}(\ln n)^{2/3}(\ln\ln n)^{1/3}\end{aligned}$$

This is an estimate of the bound K. We want to find the running time for the overall sieving, so we place the estimate for K into Theorem 2

$$\begin{aligned}L(K)^{\sqrt{2}} &= \exp\left(\sqrt{\ln K\ln\ln K}\right)\\ &= \exp\left(\sqrt{2}\sqrt{\frac{4}{3^{2/3}}(\ln n)^{2/3}(\ln\ln n)^{1/3}\ln\left(\frac{4}{3^{2/3}}(\ln n)^{2/3}(\ln\ln n)^{1/3}\right)}\right)\\ &= \exp\left(\sqrt{2}\frac{2}{3^{1/6}}(\ln n)^{1/3}(\ln\ln n)^{1/6}\left(\frac{2}{3}\ln\ln n\right)^{1/2}\right)\\ &= \exp\left(\sqrt{2}\frac{2}{3^{1/6}}\frac{\sqrt{2}}{3^{1/3}}(\ln n)^{1/3}(\ln\ln n)^{2/3}\right)\\ &= \exp\left((64/9)^{1/3}(\ln n)^{1/3}(\ln\ln n)^{2/3}\right)\end{aligned}$$

We made a simplification by using only the dominate term $\frac{2}{3}\ln\ln n$ from $\ln\left(\frac{4}{3^{2/3}}(\ln n)^{2/3}(\ln\ln n)^{1/3}\right)$. Since $L(K)^{\sqrt{2}}$ now only depends on n, we rename it $L(n)^{\sqrt{2}}$.

An estimate for the bound B is easily found, since $B^2 = L(n)^{\sqrt{2}}$

$$\begin{aligned}B &= L(n)^{\sqrt{2}/2}\\ &= \exp\left(\frac{1}{2}\cdot(64/9)^{1/3}(\ln n)^{1/3}(\ln\ln n)^{2/3}\right)\\ &= \exp\left((8/9)^{1/3}(\ln n)^{1/3}(\ln\ln n)^{2/3}\right)\end{aligned}$$

In the estimation we assumed that the a and b's were bounded by the same limit U. In the sieving we begin at the smallest b value and increase it until enough smooth values are found. It will typically be before we reach $b = U$, so the upper bound for the b is a good overestimate. This will not be a problem in practise, regardless of how many available computers for the parallel activity one have, since one just sieves the smallest b's first and increase until enough values are found.

So U is an overestimate for b. Now look at a. Could it be that the limit on the a's would benefit from being less rigid? For instance could different limits on the a's speed up the sieve by returning smooth numbers more frequently? If we look at the shape of the $a + bm$, we see that b contributes more to the total size of $a + bm$

than the a value. That means that increasing b values have a more decreasing probability to return a smooth number compared to the growing a values. The same could be said about the $N(a + b\theta)$. So if we let the interval for a for a b decrease as b increase, we will save time on the most unlikely numbers. This activity will change the sieving from a "rectangle" into a "trapezium".

An estimate for d satisfying the estimated running time remains. We put our estimate for K into (6.4) and use the same simplifications given above. Then

$$\begin{aligned} d &= \frac{\sqrt{2\ln n}}{\left(\frac{1}{2}4/3^{1/3}(\ln n)^{2/3}\right)^{1/4}\left(\frac{2}{3}\ln\ln n\right)^{1/4}} \\ &= \frac{\sqrt{2\ln n}}{4^{1/4}/3^{1/3}(\ln n)^{1/6}(\ln\ln n)^{1/3}} \\ &= \left(\frac{3\ln n}{\ln\ln n}\right)^{1/3} \end{aligned}$$

Our estimation on d does not consider the other qualities we mentioned that $f(x)$ should have. For instance the *odd* degree of $f(x)$ used to the test if we have chosen the right square root in the NFS. When implementations of the algorithms are done, polynomials of degree $3-6$ are often chosen. See [5] for an explanation to why the polynomial of degree 6 was chosen when they split the 232-digit number RSA-768 and how they located the square root.

The other huge part is finding the linearly dependent set in the matrix. For the purpose of this work, it was not implemented, but we will discuss some of its complexity.

Both algorithms requires solutions over large sets of linear equations over finite fields, and it has long been regarded as the possible bottlenecks of the algorithms. There exists different algorithms to solve the linear algebra, as the structured Gaussian elimination, the finite field versions of the Lanczos and conjugate gradient algorithms and the Wiedemann algorithm. They are all surveyed in [8].

The key is that the linear systems are sparse. It is therefore beneficial to first use the structured Gaussian elimination in both algorithms, since it uses the fact that the system is sparse to reduce the system. For the method to be optimal the linear system should contain a lot more equations than unknows. It is therefore worth spending extra time on collecting relations, since the sieving is a cheaper operation than the linear algebra. One downside of the method is that it may produce dense equations, which is harder to solve.

After the preliminary step with the structured Gaussian elimination a combination of the remaining methods solves the system.

In the NFS-dlog, the linear algebra is a more difficult problem, since it is done modulo a large prime, not modulo 2. Again, the matrix will be sparse, and in addition the entries in the matrix will be not very big, due to the fact that not many smooth integers contains a huge prime power. The methods to solve the matrix take advantage of this, but the overall time will still be bigger than in the NFS. The linear algebra in the NFS-dlog is the main reason why the algorithm is considered harder than the NFS.

Overall, we have that the linear algebra is likely to be a significant, but not an unmasterable problem when computing discrete logarithms modulo a large prime. As already said, the sieving can be distributed among a lot of computer resources, but this is not possible for the linear algebra. It requires either a fast processor or could use a closely related set of processors. It is therefore beneficial to overdo the sieving part and obtain many excess equations. This will increase the time spend on the sieving, but be positive for the overall running time.

Using the methods mentioned it can be shown that the overall time spend on the linear algebra is approximately $B^{2+o(1)}$.

Now the overall running time for the number field sieve is

$$L(n)^{\sqrt{2}+o(1)} = \exp\left((64/9)^{1/3} + o(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}\right)$$

The $L(n)^{\sqrt{2}}$ shows the dominate term and the $L(n)^{o(1)}$ takes care of all smaller terms. If we now replace n with q in the estimation, we get the optimal boundaries and time for the NFS-dlog, as

$$\begin{aligned} B &= \exp\left((8/9)^{1/3}(\ln q)^{1/3}(\ln\ln q)^{2/3}\right) \\ d &= \left(\frac{3\ln q}{\ln\ln q}\right)^{1/3} \\ L(q)^{\sqrt{2}+o(1)} &= \exp\left((64/9)^{1/3} + o(1))(\ln q)^{1/3}(\ln\ln q)^{2/3}\right) \end{aligned}$$

The estimated boundaries considered as optimal must yield for the practical boundaries given by the available computers, but they are the reason why the NFS and the NFS-dlog are considered as the fastest algorithms to solve their respective problems as of today.

CHAPTER

SEVEN

CONCLUDING REMARKS

We have outlined the NFS and the NFS-dlog in this thesis, but we have not studied all the details in the implemented versions of them as used today. And there is also always potensial for further development. We briefly discuss some aspects in this final chapter.

There exists different versions of both sieves, depending on the given information about the composite number to split or the group in which to locate the discrete logarithm. The approaches also depend on the available computers and the purpose of the action. The last point refers to for instance seeking several discrete logarithms in a field, not just one. Some variations of the sieves uses different setup when it comes to the polynomial, for example using two polynomials instead of one. In the description of the QS in Chapter 2 we presented a variation of the polynomial known as the Montgomery's multiple polynomial variations which collected the smooth numbers faster. Could a family of polynomials improve the number field sieves?

The polynomial selection is in a trial and error phase for now and more guiding lines in finding the best polynomials would decrease the sieving time. Both the degree of the polynomial and the other parameters are chosen before the algorithms are implemented, which makes it hard to choose the most effective ones. The guiding lines as of today are the heuristic analysis and the implementer's experience. We cannot affect the last one, but the analysis could be improved. One of the reasons why it is heuristic, is the heuristic Theorem 1, stating the propability that a random number less than a limit is smooth.

We spent a lot of time on the sieving method and more or less built it from the ground and up. We briefly mentioned the lattice sieve in Section 3.4, where the main principle is to make a sublattice with a special prime and sieve in this smaller region, using either line sieving or vector sieving. The method is more effective than usual sieving methods, regardless of its time-consuming operations and collisions, and it is the method of choice when the algorithms are implemented today.

There is another interesting sieving approach that for now works in the NFS-dlog for fields of degree 6, namely the 3D sieve. It extends the sieving region to a box and expands either the line sieve or the lattice sieve to fit the new sieving area. A use of the 3D sieve seems too slow for gains yet because of its slow factor base transformation and norm estimation, but it has potensial. It is an open question whether there exists an algorithm to identify the special primes best suited for the sieve.

After a thorough description of one of the two main steps in both algorithms, a discussion of the implementation and the complexity of the linear algebra step was left out. The linear algebra is an important step in both algorithms, but also more intuitive than the sieving. However, as the numbers to split and the order of the fields in which to locate the discrete logarithm grows, the complexity of the linear algebra step will continue to increase, so an in-depth study of this highly nontrivial step is necessary.

The development in the discrete logarithm problem algorithms lies far behind the factorization algorithms, as spotted when comparing the record of the NFS-dlog to the NFS record. Some of this can be explained by the fact that not many people are working with the discrete logarithm problem. There is simply not as much work and energy being put into the problem as with the splitting. Also, there are a lot more variations due to the different fields. In particular the fields of order $q = p^k$, p a prime, and the fields of order 2^k are considered as interesting. And also some recent work in finite fields has mainly been in modifying, analysing and implementing different versions of the ICM.

Although there are different discrete logarithm algorithms of the desired running time developed for many fields, there still exists finite fields where the time bound does not hold. It is an open question whether or not it is possible to locate an algorithm for all fields with the desired running time.

Lastly we mention the computers. The development of the algorithms has been marvelous since the problems became interesting, and already the records have gone far beyond what was estimated to be possible at this point. The development of the algorithms is of course a major reason, but also the bigger, faster and stronger computers play a huge role. With the computers continuing to develope, the algorithms as they are today will also increase their limits.

BIBLIOGRAPHY

[1] Matthew E. Briggs. An Introduction to the General Number Field Sieve. Master's thesis, Virginia Polytechnic Institute and State University, 1998.

[2] Jean-Marc Couveignes. *Computing a square root for the number field sieve*, pages 95–102 in [4]. Springer-Verlag, 1993.

[3] Daniel M. Gordon. Discrete Logarithms in GF(p) using the Number Field Sieve. *SIAM J. Discrete Math*, 6, University of Georgia, February 1992.

[4] A.K. Lenstra & H.W. Lenstra Jr. *The development of the number field sieve.* Lecture Notes in Mathematics, Volume 1554. Springer-Verlag, Berlin, 1993.

[5] Thorsten Kleinjung. Factorization of a 768-bit RSA modulus. (1.4), EMC Corporation, February 2010.

[6] Neal Koblitz. *A course in Number Theory and Cryptography.* Springer, second edition, 1994.

[7] James S. Milne. Algebraic Number Theory (v3.00). pages 155 + viii, 2008. Available at www.jmilne.org/math/.

[8] B.A. LaMacchia & A.M. Odlyzko. Solving Large Sparse Linear Systems Over Finite Fields. AT&T Bell Laboratories, Springer, 1990.

[9] Carl Pomerance. A Tale of Two sieves. *Notices of the AMS*, (43), 1996.

[10] Carl Pomerance. Fast, Rigorous Factorization and Discret Logarithm Algorithms. *Discrete Algorithms and Complexity*, (9), Academic Press, 1987.

[11] Richard Crandall & Carl Pomerance. *Prime Numbers, A computational Perspective.* Second Edition, Springer, 2000.

[12] Oliver Schirokauer. The impact of the number field sieve on the discrete logarithm problem in finite fields. *Algorithmic Number Theory*, (44), MSRI Publications, 2008.

[13] Oliver Schirokauer. Discrete Logarithms and Local Units. *Philosophical Transactions: Physical Sciences and Engineering*, (345), The Royal Society, 1993.

[14] Nigel Smart. *Cryptography: An introduction.* McGraw-Hill College, 2003.

[15] William Stein. *Introduction to Algebraic Number Theory.* Springer, 2005.

[16] Chris Studholme. The Discrete Log Problem. *University of Toronto*, June 2002.

[17] Edwin Weiss. *Algebraic Number Theory.* McGraw-Hill College, 1963.