J. Shanghai Jiao Tong Univ. (Sci.), 2018, 23(1): 138-145
https://doi.org/10.1007/s12204-018-1919-8

A Note on the Behaviour of the Number Field Sieve in the Medium Prime Case: Smoothness of Norms

BENGER Naomi[1], *CHARLEMAGNE Manuel*[2*], *CHEN Kefei*[3] (陈克非)
(1. School of Mathematical Sciences, University of Adelaide, Adelaide 5005, Australia;
2. University of Michigan - Shanghai Jiao Tong University Joint Institute,
Shanghai Jiao Tong University, Shanghai 200240, China;
3. School of Science, Hangzhou Normal University, Hangzhou 311121, China)

Abstract: As we examine the behaviour of the number field sieve (NFS) in the medium prime case, we notice various patterns that can be exploited to improve the running time of the sieving stage. The contributions of these observations to the computational mathematics community are twofold. Firstly, we clarify the understanding of the true practical effectiveness of the algorithm. Secondly, we propose a test for a better choice of the polynomials used in the NFS. These results are of particular interest to cryptographers as the run-time of the NFS directly determines the security level of some discrete logarithm problem based protocols.
Key words: number field sieve (NFS), pairing friendly elliptic curves, polynomial selection
CLC number: O 17, O 23 **Document code:** A

0 Introduction

In cryptography, the computational behaviour of particular algorithms is used to determine the security level of a protocol. More specifically, as the security of cryptographic protocols is based on the hardness of underlying mathematical problems, it follows that the security is reliant on the computational effort required to solve those problems in the given implementation context. It is therefore important to gain a sound knowledge of the behaviour of the algorithms used to solve these problems in practice so as to implement cryptographic protocols with reliable security settings.

The discrete logarithm problem (DLP) is the basis of security in numerous cryptographic protocols instantiated in either the additive group of points on an elliptic curve DLP (ECDLP) or a multiplicative group of a finite extension field, DLP.

The most efficient algorithms known to solve the ECDLP have been thoroughly examined and their computational behaviour in practice is well understood[1-5]. In contrast, the DLP in medium prime extension fields has received comparably less attention. As a result, the behaviour of the algorithm to solve it remains limited. Hence, there is a lack of implementation guidelines for protocols whose security is dependent on the hardness of this DLP instance. In particular, in pairing-based cryptography (PBC) the underlying problems that form the basis for security are usually related to variations of both ECDLP and DLP. The hardness of the first problem is well understood, while the one of the second still remains elusive; this complicates parameter selection.

As various known algorithms solve the DLP, the most appropriate depends on the context of the problem. According to the complexity analysis in Ref. [6], the number field sieve (NFS) in the medium prime case is the most efficient algorithm to solve the instances of finite field DLP occurring in PBC.

Little is known about the practical effectiveness of the NFS in the medium prime case, which limits our understanding of the security of PBC protocols. In this work, we examine one aspect of this NFS version.

The ultimate goal of this continuing project is to increase the general understanding of the behaviour of the NFS. The motivation for this work is threefold. Firstly, the behaviour of other NFS versions is known to vary widely[7-8]. Secondly, implementation of cryptographic protocols on small devices requires accurate security estimates as memory and processing power are limited. Finally, recent advances in solving the DLP in binary fields[9-11] promote the use of medium prime fields in PBC; it is therefore necessary to investigate the problem in this increasingly utilized setting.

The analysis of the NFS algorithm in Ref. [6] is given for finite fields $\mathbb{F}_{p^n}$ as both p and n tend to infinity which necessitates loss of detail and generalisation to

Received date: 2017-07-29
∗**E-mail:** charlem@sjtu.edu.cn

limiting cases. We wish to fine-tune this analysis and model the behaviour of the NFS in the context of PBC, while retaining as much detail as possible. In this endeavour, we perform statistical analysis on experimental results and outline important practical observations. This lead to the development of a "pre-NFS" polynomial test, that offers hints on the effectiveness of a particular NFS instance, and to the introduction of a variation in the NFS polynomial selection. The results of our work are of interest to implementers of the NFS and pairing-based protocols alike.

1 Framework

We first introduce all the necessary background to understand the deeper aspects of this work.

1.1 NFS in a Nushell

The NFS can be considered as a sequence of stages whose complexity is assessed separately. The two main parts of the NFS are the sieving stage and the linear algebra stage, followed by a descent stage. Suppose we wish to solve a DLP instance in the finite field $\mathbb{F}_{p^n}$, where p is a prime and n is a strictly positive integer.

In the sieving stage, we aim to construct a set of linear equations in the logarithms of "small" elements. This is done with two number fields, say $\mathcal{K}_1$ and $\mathcal{K}_2$, both of which contain $\mathbb{F}_{p^n}$ as a subfield. We sieve through the elements of these fields, looking for "doubly-smooth" pairs. Two elements form a pair if the mappings from their respective number fields to $\mathbb{F}_{p^n}$ map them to the same element. An element is smooth if it can be written as the product of small elements, i.e. with norms below a given bound; the pair is doubly-smooth if both are smooth in their respective number fields.

The system of linear equations is solved in the linear algebra stage to find the discrete logarithms of all the small elements. In the descent stage, the specific element of interest is written as a product of small elements and thus its discrete logarithm is found.

In this work, we examine the complexity of the sieving stage. For the NFS variation relevant to the PBC context, the two number fields are constructed as $\mathcal{K}_i = \mathbb{Q}[\theta_i]$ for $i \in \{1, 2\}$. The element θ_i is a zero of the irreducible polynomial $f_i(x)$ with integer coefficients such that $n \mid \deg(f_i)$; f_1 and f_2 have a common root modulo p. The elements of these number fields are thus considered as polynomials of degree $t < n$ in θ_i. To compute the norm of an element $a = a(\theta_i)$ in $\mathcal{K}_i$, we consider a as a polynomial in x and take the resultant of $a(x)$ with $f_i(x)$. To find doubly-smooth pairs, we look at the factorisation of the norms of the elements.

1.2 Smoothness Probability of Norms

The run time of the sieving stage of the NFS is determined by the probability of finding doubly smooth relations. There exist varying methods in the literature to predict smoothness probabilities in different contexts. The number of integers less than x with no prime factor exceeding y is denoted by $\Psi(x, y)$; the probability of a random integer of size, approximately the size of x to be y-smooth, is $\Psi(x, y)/x$.

In Ref. [12], a comprehensive survey of estimates for $\Psi(x, y)$ is given; in the more recent Ref. [13], a method for computing Ψ within an arbitrarily tight bound is presented. The results of Ref. [14] are extensions of some of the theorems in Ref. [12] to the context of algebraic number fields, the relevant case for the NFS.

The formula for calculating the probability is directly obtained from the $\Psi_K(x, y)$ estimate (number of integral ideals with norm less than x, all of whose prime divisors have norm less than y in a number field K) (see Ref. [14], Satz 3). Computation of this estimate is not feasible; it relies on knowledge of the class number of K and a result of the Dedekind zeta function, both of which are known to be hard to compute in practice. This estimate thus cannot be used in the analysis of the NFS, a major setback when one trys to estimate the computational behaviour of the NFS. Indeed, the variability of the probability of smoothness of norms in given number fields (that is, for a given choice of f_1 and f_2) could be responsible for some of the behaviour noticed by Zajac[7]. The ability to compute this probability would aid in the selection of the polynomial pair (f_1, f_2).

The smoothness probability estimate for integers used for the complexity analysis in Ref. [6] is from a corollary[15]; this formula is the most appropriate choice, given that the algorithm is presented for extension degree k and prime p both tending to infinity. We will show in Subsection 2.1 that this estimate can be significantly improved in the context of PBC.

2 Analysis

In this section, we outline our experimental parameters and a variation of the NFS polynomial selection process (Table 1). As the contextual focus is PBC, the current field sizes suggested in the literature (from resources such as Refs. [16-18]) are used for the security levels 80, 128 and 192. This fixes the types of elliptic curves that can be used and therefore also fixes n. Following the complexity analysis of Ref. [6], we

Table 1 Parameter sizes for common pairing friendly elliptic curve families

Curve family	Security level/bit	Size of p/bit	n	t
MNT6	80	160	6	2
BN	128	256	12	3, 4
KSS18	192	512	18	4

compute the corresponding values for t (degree of elements sieved) for each of the three instances. We present our results using a working example of the pairing friendly elliptic curve family MNT6; the methods are directly adaptable to other families such as BN and KSS18.

2.1 Polynomial Selection

To have the NFS polynomials of simple structure, we find two integers a_1, a_2 of size about $\sqrt{p}$ and such that $a_1 a_2 = p + i$, where i is some small integer. We define the polynomials f_1 and f_2 as $f_1 = x^n + a_2$ and $f_2 = a_1 x^n + i$ when both are irreducible.

It is straightforward to show that these polynomials have a common zero modulo p (in fact have the same set of zeros) and are therefore appropriate for use in the NFS. Using this polynomial selection means that the sieving space does not need to be skewed to balance the sizes of the norms of the elements in each number field (see Ref. [6] for details). One important detail supporting the use of binomials in this context is that for efficient pairing implementation the prime is chosen to be $p \equiv 1 \mod n$ (or the prime factors of n)[19], so it is possible to find irreducible binomials for the NFS polynomial selection in this context. For the BN case, we use irreducible polynomials constructed as described for the tower constructions in Ref. [20].

The process is divided into three parts. Firstly we use empirical data to compute the cumulative distribution function of the norms and observe which parameters influence the distribution; then we illustrate how the cumulative distribution affects the smoothness probability. Secondly, we show how to determine an improved estimate of the smoothness probability using the cumulative distribution function and the results of Ref. [13]. Thirdly, we present a test for the effectiveness of the f_1 and f_2 polynomials against the expected outcome.

2.2 Cumulative Distribution of Norms

Once the polynomials f_1 and f_2 have been selected, the computation of the norms can be reduced to the evaluation of an integer polynomial. The arguments are taken from a fixed interval $[0, S]$ for a sieving bound S, computed following the instructions in Ref. [6], as is the smoothness bound $\mathcal{B}$. The arguments are considered as independent, uniformly distributed, random variables $X_0, X_1, \cdots, X_t$ from $[0, S]$. In that context the norm is also a random variable, N, taking values in the range of the function $\mathcal{N}(X_0, X_1, \cdots, X_t)$, defined by the determinant of the Sylvester matrix. We are interested in the cumulative distribution of $N = \mathcal{N}$.

Our focus is on the cumulative distribution since the estimates for the probability of smoothness are in fact cumulative probabilities, i.e., $\Psi(x, y)/x$. This cumulative distribution of N is used to weight the smoothness probability estimate of the integers and find a more accurate smoothness probability estimate for the norms.

In the MNT6 setting, the resultant of polynomials of the form $Ax^6 + B$ with $X_0 + X_1 x + X_2 x^2$ is given by the evaluation of the function

$$\mathcal{N}(X_0, X_1, X_2, X_3) = A^2 X_0^6 - 2ABX_0^3 X_2^3 + 9ABX_0^2 X_1^2 X_2^2 - 6ABX_0 X_1^4 X_2 + ABX_1^6 + B^2 X_2^6,$$

where $A = 1$ for f_1. Since numbers generated through this equation have much structure, their distribution is very different from the uniform one, the assumption used in the complexity analysis. Trying various theoretical methods to determine the probability distribution of this N from this function is unfruitful due to the complexity and large number of variables. We therefore generate empirical data to directly determine the cumulative distribution.

Figure 1 shows the cumulative distribution function as generated for the norms in the MNT6 case, where M is the largest norm generated. Intuitively, we expect a "clumping" effect in the centre of the probability distribution of the norms as we are repeatedly summing 6th degree products of uniform variables from a bounded interval: this increases the proportion, and therefore probability, of the low to middle range norm values. The result of this clumping effect in the probability distribution translates to a very steep cumulative distribution function over the low and middle range (plentiful) values. In turn this tapers down to very small slope over the larger (rarer) values. This is exactly the result that the data present. In particular, Fig. 1 highlights how our cumulative distribution function as generated for the norms in the MNT6 case differs from the uniformly distributed variable.

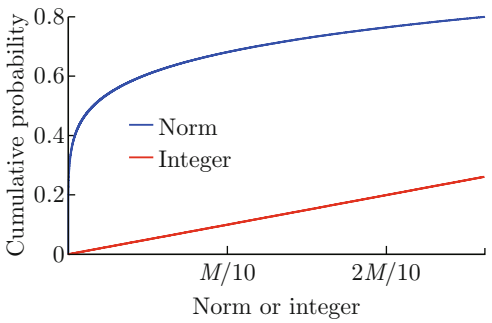

Fig. 1 Comparison of the cumulative distribution of norms and uniform integers in the MNT6 case, with a 160 bit prime

This central clumping clearly results in the "mid-range" values being more probable in the norm distribution than for the integers. The number of integers in a particular interval increases constantly with the upper bound of the interval whereas the norms have a higher concentration in the mid-range of the same interval, as a direct consequence of the range of X_0, X_1 and X_2. Indeed in this case, 80% of the norms are of size less than a quarter of the size of the maximum norm M generated during the experience.

Another crucial observation is that the distribution of the norms depends only on n and t. In fact, the size of p only dictates the placement of the distribution not its shape. This is not surprising, given that the shape of the elements of interest is the output of the above function $\mathcal{N}(X_0, X_1, \cdots, X_t)$, a degree n function in t variables. This remark allows us to use a smaller example case than would be used in practice: the MNT6 setting and a 50 bit prime (with $n = 6$ and $t = 2$). As a result, we can conduct more thorough and varied tests, only recalculating the sieving bound to correspond with the new field size[6], while preserving the value for t.

Fitting a Line to the Cumulative Distribution The line fitted to the cumulative distribution as presented in Fig. 1 is generated with 2 million norm values (X_0, X_1, X_2 chosen uniformly at random) for a range of primes and polynomials. To obtain the equation of this line, we perform curve fitting using **R**, a language package for statistical computing[21]. As the initial experiments indicate that this is not an exponential curve, this prompts us to use the Box-Cox method[22]. This method aids us in identifying a suitable power transformation on the response variable ($y =$ the proportion of norm values $\leqslant x$) to obtain a linear relationship between the explanatory variable x and the modified response variable; that is, we wish to find a value λ such that $y^\lambda = ax + b$. Often the initial test range for λ is $[-5, 5]$ and the default setting in **R** is $[-2, 2]$ incrementing in steps of 0.1. The graph in Fig. 1 represents the density of norm values, and as the norms are sums of monomials of degree 6 we expect λ to be close to 6 and therefore set our test range to be $[0, 10]$. Instead of using all 2 million points (which would have made the tests very slow) we repeat the test numerous times on random samples of 10^4 points. The results for each of the tests give clear indication that $\lambda = 6$.

We then proceed to fit a linear model to the full data set of points of the transformed data (x, y^6) to find that $Y^6 = aX + b$ where $a = 2.171 \times 10^{-106}$ and $b = -1.868 \times 10^{-5}$ with negligible variance. From a statistical perspective, the fit is incredibly accurate as there is no noise in the data. The exact output of the linear model fitted in **R** is given in Tables 2 and 3, where 1Q and 3Q represent the first and third quartiles, respectively.

Table 2 R output: residuals of the linear model

Min	1Q	Median	3Q	Max
-8.589×10^{-4}	-7.149×10^{-5}	3.580×10^{-6}	1.868×10^{-5}	7.928×10^{-4}

Table 3 R output: coefficients of the linear model

	Estimate	Standard error	t value	$P\{>\|t\|\}$
Intercept	-1.868×10^{-5}	1.701×10^{-7}	-109.8	$< 2 \times 10^{-16}$
x	2.171×10^{-106}	1.331×10^{-112}	1 630 932.3	$< 2 \times 10^{-16}$

How the Cumulative Distribution Affects the Smoothness Probability To illustrate why the probability of smoothness is affected by the differing probability distributions, we use the law of total probability. In terms of random variables, we define Z to be a random variable taking integer values from the interval $[0, U]$. As defined, N is a randomly selected norm (also from the interval $[0, U]$) and S_Z (resp. S_N) is a binomial random variable such that

$$S_Z = \begin{cases} 1, & Z = z \text{ is smooth} \\ 0, & Z = z \text{ is not smooth} \end{cases}.$$

We define S_N similarly where smooth in both cases is with respect to some fixed bound $\mathcal{S}$. Now the probability that a random integer z selected from the defined interval uniformly at random is smooth is expressed as

$$P\{S_Z = 1 | Z = z\}.$$

Following the law of total probability, we partition the interval $[0, U]$ into s equally sized sections, of width $U/s = a$, and write this probability as a summation

$$P\{S_Z = 1\} = \sum_{i=1}^{s} P\{S_Z = 1 | z \in [(i-1)a, ia]\} P\{z \in [(i-1)a, ia]\}.$$

Applying the same method, we can rewrite $P\{S_N = 1\}$ as

$$P\{S_N = 1\} = \sum_{i=1}^{s} P\{S_N = 1 | n \in [(i-1)a, ia]\} P\{n \in [(i-1)a, ia]\}.$$

Assuming the probability of a norm being smooth is equal to the probability of an integer of the same size being smooth (the validity of this assumption will be

discussed further in Subsection 2.3), the left probability value in each of the summations is identical for every value of i and can be computed by the method introduced in Ref. [13] (a free implementation is available at Ref. [23]). From different distributions of the norms and integers, we know that the right hand values are quite different and so the sums will clearly not be equal.

The probabilities on the right in the $P\{S_Z = 1\}$ expression are exactly the s-quantiles (s is the number of partitions); as $P\{Z \leqslant ia\} = \dfrac{ia}{U}$ and so for all i we have

$$P\{z \in [(i-1)a, ia]\} = P\{Z \leqslant ia) - P(Z \leqslant (i-1)a\} = \frac{(i-1)a - ia}{U} = \frac{a}{U}.$$

In other words, the right probabilities in the $P\{S_Z = 1\}$ expression are constant and the expression becomes

$$P\{S_Z = 1\} = \frac{a}{U}\sum_{i=1}^{n} P\{S_Z = 1 | z \in [(i-1)a, ia]\} = \frac{a}{U}\sum_{i=1}^{n} \frac{n_{\mathrm{e}} \in [(i-1)a, ia]}{a} = \Psi(\mathcal{S}, U)/U,$$

where n_{e} represents the number of smooth elements.

Examining Fig. 1, we see that the probabilities $P\{n \in [(i-1)a, ia]\}$ will be quite different. Through the law of total probability we not only highlight how the distribution of the norms affects the probability of smoothness, but also present a method for computing a better estimate for the smoothness probability of the norms.

2.3 Estimating the Smoothness Probability of Norms

To compute $P\{S_N = 1\}$ we use the same approach as above, though instead of having the partitions evenly spaced we use the s-quantiles of the distribution of the norms; that is, we fix the intervals such that the probability that n is in any interval is fixed at $1/s$. Fixed probability and even spacing are synonymous for the uniform distribution, but not for the distribution of the norms. Using **R** we easily compute the 100-quantiles of the norms, and find the values $a_0, a_1, \cdots, a_{100}$, such that $P\{n \in [a_i, a_{i+1}]\} = 0.01$ for $i \in [0, 99]$. As we assume $P\{S_N = 1 | n \in [a_i, a_{i+1}]\} = P\{S_Z = 1 | z \in [a_i, a_{i+1}]\}$, we still need to compute

$$0.01\sum_{i=0}^{99} P\{S_N = 1 | n \in [a_i, a_{i+1}]\},$$

so as to determine the probability of smoothness of the norms. Using the implementation[23], we compute the probabilities in the summation to obtain an estimate of the smoothness probability

$$P\{S_N = 1\} = 0.000\,169\,920\,1 = \hat{r}$$

whereas

$$P\{S_Z = 1\} = 0.000\,057\,778\,68.$$

The smoothness probability of the norms over this interval is higher by a factor of almost 3 (2.94). This is no surprise as, intuitively, smaller numbers are more likely to be smooth and the norms have a higher concentration of "small" to "medium" numbers than the integers do. These results are reflected in experimental results obtained by selecting integers at random, following the distribution of the norms, and testing for smoothness; almost three times as many smooth integers are obtained through this method as compared with selecting integers uniformly at random.

The probability above is an optimistic-case probability; we have made the assumption that the probability of a norm being smooth is equal to the probability that an integer of the same size is smooth. That is, for some integers b_0 and b_1 we assume that

$$P\{S_N = 1 | n \in [b_0, b_1]\} = P\{S_Z = 1 | z \in [b_0, b_1]\}.$$

This is, however, not necessarily the case. In any given interval, the integers are denser than norm values (approximately $1/n$) and the norms will not necessarily coincide with the smooth integers at the same rate as the actual proportion of smooth integers. As the number of quantiles grows, however, this assumption is better supported by the experimental results than the assumption used in the complexity analysis[6]. In the event that the norms fall on proportionally more smooth integers, the NFS sieving stage will execute faster. Table 4 shows examples of polynomials found for an example MNT6 prime for which the smoothness rate r is calculated from $1\,000/\hat{r} \approx 5 \times 10^6$ tests.

Table 4 Polynomials found for the MNT6 prime $p = 1\,125\,909\,838\,976\,401$

Polynomial f	n_{e}	r	β
$35\,374\,332x^6 + 11$	633	0.000 126 6	0.745 1
$11\,791\,444x^6 + 11$	1 370	0.000 274 0	1.612 5
$17\,687\,166x^6 + 11$	652	0.000 130 4	0.767 4
$x^6 + 8\,264\,689$	1 379	0.000 275 8	1.623 1
$1\,069\,214x^6 + 37$	1 320	0.000 264 0	1.553 7

Note that although all of our examples presented here have positive i values, we examine positive and negative values for i (that is, 4 complex roots and 6 complex roots respectively), both separately and all together. We observe strong evidence to support the hypothesis that the smoothness rate of the norms is the same in the two cases and thus proceed to examine all the data together.

The values of β, the factor difference from $\hat{r}$, in the fourth column can be compared with the value

$1/3 = 0.33\bar{3}$ which is used in the complexity analysis. We see the second, fourth and fifth polynomials produce smooth norms at a rate over 50% faster than the prediction (and so 4.5 times the estimate used in the complexity analysis). We are able to find many examples of polynomials yielding better than estimated performance. The average factor difference from $\hat{r}$ is $\bar{\beta} = 0.552\,699\,9$, as shown in Fig. 2. The long tail of the graph distorts the standard deviation, but after examining the quantiles, half of the values are between 0.28 and 0.71 with 1 being near the end of the main peak. This explains the use of the term optimistic-case probability.

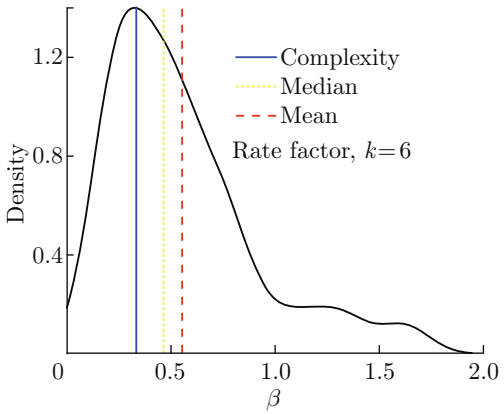

Fig. 2 The probability density function of the multiple β by which the rate of smooth relation collection differs from the estimated rate

In Fig. 2, the value given by the asymptotic complexity is shown; it is the mode of the distribution. It is clear that a much larger proportion of the cases perform significantly better than this case. An NFS implementer would always endeavour to launch the most efficient algorithm possible. Thus the average is a better reflection of the rate than the mode when one estimates the runtime of the sieving stage and consequently the security of a PBC protocol. The existing complexity analysis thus gives an underestimation of the probability of finding doubly-smooth pairs.

The goal of an implementer of the NFS is to find a polynomial pair such that the rate of smooth relation collection in each number field is maximal, so the implementer would aim to find pairs (f_1, f_2) such that $r_1 r_2 \approx \hat{r}^2$ (r_i is smooth rate of norms computed by f_i, $i \in \{1, 2\}$); this is not a straightforward task. We denote $r_1 r_2 = \delta \hat{r}^2$ and recognize that the possible values for δ will be the product of independently selected values from the distribution of β and will therefore have mean $\bar{\beta}^2$. From our empirical data we compute $0.332\,089\,8$, very close to the predicted $0.306\,289\,8 \approx 0.552\,699\,9^2$ (see Fig. 3). We can compare this with the estimate used in the complexity analysis which would give a value of $\delta = 1/9 = 0.11\bar{1}$. Even if we consider the median of this distribution, $0.211\,886\,5$, to minimize the influence of the large, rare cases, it is still approximately twice the probability used in the complexity analysis and reflects more closely the observed behaviour from our experiments.

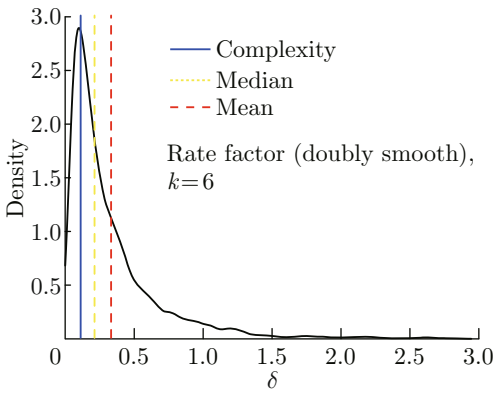

Fig. 3 The probability density function of the multiple δ by which the rate of smooth relation collection differs from the estimated rate (doubly smooth case)

2.4 Polynomial Selection Test

When our polynomials are selected for the NFS, the mean value of δ gives us a benchmark to obtain, or to improve on. For a given instance of the DLP in a finite field $\mathbb{F}_{p^n}$, we perform the following test before launching the NFS sieving stage.

Step 1 Compute $\hat{r}$, the upper estimate on the smoothness probability of norms.

Step 2 Find a pair f_1 and f_2 using the method from Subsection 2.1 or Ref. [6].

Step 3 Generate $1\,000/\hat{r}$ random norms for both f_1 and f_2. Test if they are smooth to approximate the smoothness rate for each number field.

Step 4 Compare the product of the approximate rates of smooth norms to $\hat{r}^2$. If it is "large enough", i.e. $\hat{\delta}$, the experimental δ value for the polynomial f_1, f_2, is larger than the mean value, then proceed with sieving; otherwise goto Step 2.

This differs to how the results in Ref. [7] would be used: we have given a benchmark to compare the rate of smooth norm collection against, whereas previous methods would have required the generation of numerous pairs of polynomials to mutually compare the rate of smooth relation collection. Thus, our method results in fewer tests necessary, to distinguish if a pair of polynomials is performing at least as well as the average case.

To illustrate, we have generated various f_1 and f_2 pairs for our example case of MNT6 with prime around 50 bit. Our experiments result in polynomial pairs for which smooth norms occur at a much higher rate than others. Tables 5 and 6 show some examples of polynomial pairs in the MNT6 case.

Table 5 Smoothness rates for various polynomials for the MNT6 prime $p = 1\,125\,909\,838\,976\,401$

Polynomial pair	n_e	$r_i \times 10^4$	$\hat{r}^2 \times 10^9$	$\hat{\delta}$
$f_1 = 35\,374\,332x^6 + 11$	633	1.266	7.545 36	0.261 3
$f_2 = x^6 + 31\,828\,441$	298	0.596		
$f_1 = 11\,791\,444x^6 + 11$	1 370	2.740	11.727 2	0.406 2
$f_2 = x^6 + 95\,485\,323$	214	0.428		
$f_1 = 17\,687\,166x^6 + 11$	652	1.304	5.424 64	0.187 9
$f_2 = x^6 + 63\,656\,882$	208	0.416		
$f_1 = x^6 + 8\,264\,689$	1 379	2.758	11.473 28	0.397 4
$f_2 = 136\,231\,362x^6 + 17$	208	0.416		
$f_1 = 1\,069\,214x^6 + 37$	1 320	2.64	4.963 2	0.171 9
$f_2 = x^6 + 1\,053\,025\,717$	94	0.188		

Table 6 Smoothness rates for various polynomials for the MNT6 prime $p = 1\,126\,169\,969\,103\,937$

Polynomial pair	n_e	$\hat{r}^2 \times 10^9$	$\hat{\delta}$
$f_1 = x^6 + 4\,747\,150$	476	10.148 32	0.351 5
$f_2 = 237\,230\,753x^6 + 13$	533		
$f_1 = x^6 + 186\,474\,550$	139	6.527 44	0.226 1
$f_2 = 6\,039\,269x^6 + 13$	1 174		
$f_1 = x^6 + 516\,240\,070$	118	5.121 2	0.177 4
$f_2 = 2\,181\,485x^6 + 13$	1 085		
$f_1 = x^6 + 20\,871\,667$	202	4.726 8	0.163 7
$f_2 = 53\,956\,877x^6 + 22$	585		
$f_1 = x^6 + 27\,201\,934$	251	6.686 64	0.231 6
$f_2 = 41\,400\,364x^6 + 39$	666		

Interestingly, observing that the rate of smooth relation collection for the first three polynomial pairs is vastly different, highlights that simply taking a small value for i (11 in this case) does not ensure a good polynomial pair. Though the rate is comparable in the number fields defined by f_2, using the first and third pairs we find smooth relations at less than half the rate of the second pair, for which the value $\hat{\delta} = 0.406\,2$ exceeds the mean value of 0.332 089 8.

Again Table 6 shows great variations in the rate of smooth element collection in the first three cases, for which the same value $i = 13$ is used to construct the polynomial pairs. For this prime, the value of $\hat{\delta} = 0.351\,5$ slightly exceeds the mean value of 0.332 089 8.

3 Conclusion

In complexity theory, constant factors are hidden in the O and L notations. As a result, the complexity does not properly reflect how an algorithm runs in practice.

In cryptography, the security of protocols is directly related to the practicality of solving hard mathematical problems. Until now the practical behaviour of the NFS in the medium prime case has received relatively little attention. Due to its relevance to the security of pairing-based protocols in particular, we have examined its run-time behaviour in the given context; the motivation of this work is to deepen the understanding of the NFS in general, in the hope that this would help determining more precise estimates for appropriate parameter sizes at a fixed security level for PBC protocols.

In this work, we present some observations on the behaviour of the NFS in practice. We focus on the smoothness probability of the norms of number field elements as it determines the practical run-time of the sieving stage. Our observations and analysis result in a pre-sieving test that can be performed on the selected polynomials. This test ensures as efficient a set up and execution of the sieving stage as possible. The new revelation about the probability of smooth norm occurrence is a step towards a more precise run-time estimate of the sieving stage. We also propose a variation of the polynomial selection method given in Ref. [6] and use it to conduct our experiments. It shows promising behaviour.

This work covers the initial progress in the examination of the NFS sieving stage; in order to compute the expected run-time of this stage, it remains to investigate the true cost of smoothness test.

References

[1] POLLARD J M. Monte Carlo methods for index computation (mod p) [J]. *Mathematics of Computation*, 1978, **32**: 918-924.

[2] TESKE E. Speeding up Pollards Rho method for computing discrete logarithms [C]//*Algorithmic Number Theory Symposium* (*ANTS IV*) *in LNCS*. Berlin: Springer-Verlag, 1998: 541-543.

[3] BAILEY D V, BALDWIN B, BATINA L, et al. The Certicom challenges ECC2-X [C]//*Workshop on Special Purpose Hardware for Attacking Cryptographic Systems*. [s.l.]: SHARCS, 2009: 1-32.

[4] BAILEY D V, BATINA L, BERNSTEIN D J, et al. Breaking ECC2K-130 [EB/OL]. (2017-07-29). https://eprint.iacr.org/2009/541.

[5] BAI S, BRENT R P. On the efficiency of Pollards Rho method for discrete logarithms [C]//*Fourteenth Computing*: *The Australasian Theory Symposium* (*CATS2008*). Wollongong: Australian Computer Society Inc., 2008: 125-131.

[6] JOUX A, LERCIER R, SMART N, et al. The number field sieve in the medium prime case [C]//*Advances in Cryptology* (*CRYPTO 2006*). Berlin: Springer-Verlag, 2006: 326-344.

[7] ZAJAC P. Remarks on the NFS complexity [EB/OL]. (2017-07-29). http://eprint.iacr.org/.

[8] BERNSTEIN D J. Predicting NFS time [R]. Nancy: Cado Workshop on Integer Factorization, 2008.

[9] GÖLOĞLU F, GRANGER R, MCGUIRE G, et al. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$ [C]//*Advances in Cryptology* (*CRYPTO 2013*). Berlin: Springer-Verlag, 2013: 109-128.

[10] GÖLOĞLU F, GRANGER R, MCGUIRE G, et al. Solving a 6120-bit dlp on a desktop computer [C]//*Selected Areas in Cryptography* (*SAC 2013*). Berlin: Springer-Verlag, 2013: 136-152.

[11] JOUX A. Faster index calculus for the medium prime case: Application to 1 175-bit and 1 425-bit finite fields [C]//*Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin: Springer-Verlag, 2013: 177-193.

[12] HILDEBRAND A, TENENBAUM G. Integers without large prime factors [J]. *Journal de Théorie des Nombres de Bordeaux*, 1993, **5**(2): 411-484.

[13] BERNSTEIN D J. Arbitrarily tight bounds on the distribution of smooth integers [J]. *Number Theory for the Millennium*, 2002, **1**: 49-66.

[14] KRAUSE U. Abschätzungen für die function $\psi_k(x, y)$ in algebraischen Zahlkörpern [J]. *Manuscripta Mathematica*, 1990, **69**: 319-331.

[15] CANFIELD E R, ERDÖS P, POMERANCE C. On a problem of Oppenheim concerning "factorisatio numerorum" [J]. *Journal of Number Theory*, 1983, **17**: 1-28.

[16] LENSTRA A K, VERHEUL E R. Selecting cryptographic key sizes [C]//*International Workshop on Public Key Cryptography*. London: Springer-Verlag, 2000: 446-465.

[17] LENSTRA A K. Unbelievable security: Matching AES security using public key systems [C]//*International Conference on the Theory and Application of Cryptology and Information Security*. London: Springer-Verlag, 2001: 67-86.

[18] SMART N. ECRYPT II yearly report on algorithms and keysizes (2011-2012) [R]. Kortrijk: University of Leuven, 2012.

[19] KOBLITZ N, MENEZES A. Pairing-based cryptography at high security levels [C]//*IMA International Conference on Cryptography and Coding*. Berlin: Springer-Verlag, 2005: 13-36.

[20] BENGER N, SCOTT M. Constructing tower extensions for the implementation of pairing-based cryptography [C]//*Arithmetic of Finite Fields, Third International Workshop* (*WAIFI 2010*). Istanbul: Springer, 2010: 180-195.

[21] The R Core Team. R: A language and environment for statistical computing [M]. Vienna: The R Core Team, 2010.

[22] BOX G E P, COX D R. An analysis of transformations [J].*Journal of the Royal Statistical Society*, 1964, **26**: 211-252.

[23] BERNSTEIN D J. Implementation of ψ estimation method of "arbitrarily tight bounds on the distribution of smooth integers" [EB/OL]. (2017-07-29). https://cr.yp.to/psibound.html.