UDC 681.3.06.

MODELING THE FORMATION PROCESSES OF FACTORIZATION ALGORITHMS BASED ON THE ELLIPTIC CURVES THEORY

I. Dermenzhy, G. Vostrov
Odessa national polytechnic university

Abstract. *This paper describes the problem of factorization. It indicates the fundamental place of this problem in a number of purely mathematical and applied sciences. The structure of factorization classes methods is analyzed, the choice of the approach based on the elliptic curves theory is substantiated. The algorithm of elliptic curves is described and analyzed in detail. The paper describes the problem of the relationship between the generated curves number and the necessary boundary of the basic method of elliptic curves. The research of this problem are made by method software implementation. Results of this research are represented.*

Key words: *factorization, elliptic curve, one-way function, exponential complexity, subexponential complexity, smooth numbers, composite numbers, pseudo-curve, finite field.*

Introduction

Factorization problem is typical for a wide range of mathematical tasks. The solution of many mathematical problems is associated with the assumption that the result of the number decomposition is known in advance. It is typical for such sciences as mathematical number theory, the theory of functions, the theory of recursion in algebra, the theory of finite fields and the theory of finite groups. For example, the number smoothness test, first of all requires its decomposition into prime factors. Further analysis is carried out on the basis of this decomposition [1]. Also it is not necessary to bring the factorization process to the end for such kind of tests. The decomposition problem also arises for various fast methods of the multidimensional discrete Fourier transform, which are widely used and have significant meaning in the theory of signals [2]. For example, methods based on the matrix factorization approach and the polynomial transformations approach. The second is based on a priori information about the factorization of special polynomials [2]. The factorization problem is part of the methods for proving the primality and pseudo-primality of numbers. In addition to the above, the need for effective factorization methods follows from the modern theory of complex dynamic systems modeling, theory of pseudorandom number generators constructing, and used for the deepening of Monte Carlo methods. The simple quick and affordable multiplicative decomposition of composite numbers can become an arithmetic operation that is inverse to multiplication, and thus replenish the arsenal of the mathematics computational means.

An absolutely new view on probabilistic number theory is formed as a result of the Erdos-Kac theorem consideration. It connects the distribution of the different large numbers prime divisors with the limit laws formulas of probability theory. According to this theorem, for any integer $n \geq 1$, if $\omega(n)$ – is the amount of given number different prime divisors, then for any real numbers $a < b$:

$$\lim_{N \to +\infty} \frac{1}{N} |\{1 \leq n \leq N \mid a \leq \frac{\omega(n) - \log\log N}{\sqrt{\log\log N}} \leq b\}| = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx.$$

is satisfied. That is, the limit distribution of $\frac{\omega(n) - \log\log N}{\sqrt{\log\log N}}$, corresponds to the standard normal distribution. Hence the conclusion, that the amount of natural numbers n with a small number of dividers increases with growth of n value.

In addition to the question of the factorization problem belonging to the one-sided functions class, the factorization problem also arises in the construction of methods for solving the discrete logarithm problem [1]. Which is also a candidate for one way functions. Both of these problems are extremely important in computer algebra and at the same time occupy the fundamental position in modern cryptography. All modern cryptographic systems are based on the assumption that one-way functions exist and that these two problems belong to a given class. Thus, the whole inconsistency of the current cryptography will be exposed by solving the factorization problem.

All these factors together explain the extreme importance and huge interest in factorization problem. Also they inspire scientists through all around the world to search for ways of its solution.

1. Analysis of factorization classes structure

The choice of the factorization method is quite a simple task, sometimes more difficult than the process itself. Conventionally, all methods can be classified as follows:

1. Exponential methods.
2. Subexponential methods.

This classification is based on the computational complexity of the methods.

Exponential complexity – in this case the complexity of the problem is bounded by the exponent of the polynomial in the problem size, that is, it is limited by the function $\exp(P(n))$, where P – some polynomial, and n – is the size of the task.

There are algorithms that work in more than polynomial time ("super-polynomial"), but in less than exponential time ("sub-exponential"). Unfortunately, the strict definition of subexponential complexity has not yet been given. At the moment, there are two main definitions.

The first definition: the problem is solved in subexponential time, if it is solved by an algorithm, which logarithm of the operation time grows less than any given polynomial [3].

The second definition: the running time of the subexponential algorithm is equivalent $2^{O(n)}$ [4]. This definition allows more time costs than the first. An example of an algorithm with a subexponential time is the general number field sieve for integers factorization.

The inaccuracy, which lies in the general phrase that this computational complexity is intermediate between polynomial and exponential, cannot be acceptable. It is necessary to get the clear assessment of the difference measure between this complexity and the exponential one. Which would display how much faster algorithms of this class actually work. In the case of factorization algorithms, the subexponential character is expressed in L-notation recording of the computational complexity. In such case, the complexity of the algorithm is an exponent of the product of some constant by the natural logarithm of the task size to a power less than one multiplied by the double natural logarithm of task size to a degree less than one. Those, it is determined by the formula: $L_n[\alpha, c] = e^{(c+O(1))(\ln n)^{\alpha}(\ln\ln n)^{1-\alpha}}$, when $n \to \infty$, where, $\alpha = const$, $\alpha \in [0;1]$ [4]. L-notation is an asymptotic notation, similar to O-notation, which is used to approximate the computational complexity of an algorithm. As can be seen, the complexity is no longer an exponential of a polynomial (as with exponential complexity), but a product of a twice logarithmic function of the task size by a logarithmic function. In this case, due to the restriction on the constant, both of these functions have degree less than 1 (that is, a fractional degree). Thus, the growth rate of such a function is much less than the polynomial function, and even than the linear function. Due this difference, that the subexponential complexity is an exponent of a function whose complexity is well below the polynomial, it is allocated to a separate class that differs from the exponential. Thus $(c + O(1))(\ln n)^{\alpha}(\ln\ln n)^{1-\alpha} << Poly(n)$ when $n \to \infty$, based on this:

$\exp((c + O(1))(\ln n)^{\alpha}(\ln\ln n)^{1-\alpha}) << \exp(Poly(n))$

when $n \to \infty$.

It is natural to choose the most computationally efficient factorization method. In this case, it is obvious that the exponential methods can be discarded from consideration, because those are much worse than the subexponential methods by the given criterion. However, they should not be abandoned. There are a number of reasons for this, such as the fact that the ideas and principles of factorization derive precisely from these algorithms due to the fact that their emergence and development preceded the formation of subexponential ones. Moreover, with a small size of a composite number that is have to be factorized, it is often more expedient to use exponential methods.

Even discarding a huge number of methods by refusing to consider exponential methods, the problem of choice will still be a great challenge for the researcher. This statement is based, as well as on the nature of each subexponential method (the considered methods are heuristic, as well as their estimates), and on the specific features of each of them. However, there is another problem. The fact there are suspicions that for different classes of composite numbers, such algorithms behave differently. There is a certain probability that on some rare numbers, the speed of these methods can sometimes be less than the speed of exponential methods. This moment, in fact, partly reflects the heuristical character of all subexponential methods. This question, in addition to its unexplored character, is also characterized by extremely low attention from researchers.

Among the subexponential algorithms the following algorithms should be highlighted: Dixon's factorization method, continued fraction factorization method (CFRAC), the quadratic sieve method (QS), elliptic curve factorization method (the Lenstra's method, or ECM) and the numerical field sieve method (NFS). The NFS is considered the most effective algorithm for factoring large numbers (more than 10^{110}) [5]. Also, there are two methods

of the n numerical field sieve: general (GNFS) and special (SNFS). The special is obviously more effective than the general, however it can be used only to factorize the numbers of a special type: $r^e \pm s$, where $r \in N$, $s \in Z$, r and s are small. Due to this limitation, this algorithm will not be considered. In table 1, below are the computational complexity of each method in L-notation.

Table 1.

The computational complexity of subexponential factorization methods

Name of method	Computational complexity
Dixon's factorization method	$L_n(\frac{1}{2};2\sqrt{2})$
Continued fraction factorization method	$L_n(\frac{1}{2};\sqrt{2})$
Quadratic sieve method	$L_n(\frac{1}{2};1)$
Elliptic curve method	$L_p(\frac{1}{2};\sqrt{2})$
General number field sieve method	$L_n(\frac{1}{3};(\frac{64}{9})^{\frac{1}{3}})$

Where n – is a composite number to factorize, and p – is smallest divisor of this number.

An important feature is that none of the considered subexponential methods is strictly justified. [5]. But it is precisely due to the randomness of the methods a lower expected record complexity is achieved. Although the use of such methods is strange in some way, nevertheless, in practice it is not at all necessary to constantly fulfill any of them, it is only necessary that their fulfillment frequency is sufficient to be useful. [5]. Among the given methods, ECM is the closest to a strict justification. This is due to the Lenstra's hypothesis, on the distribution of smooth numbers in short intervals [6]. By accepting this hypothesis, he showed that the expected number of arithmetic operations with integers of order n, required to find the smallest divider p of the composite n by ECM is equal $\exp((2+O(1))\sqrt{\ln p \ln\ln p})$, where $O(1) \to 0$, when $p \to \infty$ [6]. Thus, there is only one heuristic gap in ECM, while QS and NFS have several similar gaps in their justification. [5]. The important moment here is a definition of a smooth number. General definition of this term is one of unsolved mathematic problems. Fortunately, in elliptic curve theory the term of b-smooth number, which is in some way ascendant of smooth number term, that have strict definition. According to the definition of Leonard Adleman, an integer is called smooth, if it consists of small prime factors. At this stage, it is assumed that n is b-smooth number, if none of the prime factors of the number n does not exceed a number b.

In comparing of this three most effective subexponential methods: ECM, QS and NFS the size of the composite number smallest divisor is the main criteria. In the case when the factoring number has a size that exceeds the record value for this methods, the only way to find a divisor is factorization using elliptic curves [5]. Thus, it has great sense to research it more deeply concentrating on it the most.

2. Development and generalization of algorithms based on elliptic curves theory

The possibility of the such structures as elliptic curves using for the purpose of factorization gave a new impetus to the search for a solution to this problem. The existing Lenstra's method, based on the theory of elliptic curves, provides subexponential computational complexity. That is, it rightfully takes its place among the most computationally efficient algorithms for decomposing numbers into prime factors. At the same time, it is inherent in a huge set for all sorts of optimizations and tasks for research.

An important feature of the elliptic curves method is that its performance does not depend on the factoring number itself, but on its smallest divisor value [5]. This moment is significant for the method, since it opens up new possibilities for its use in combination with other factorization algorithms. Such as the method of a quadratic sieve, which is also subexponential, but it works faster for numbers whose dividers have a greater bit capacity. The Lenstra's method is the best algorithm for finding simple divisors of 20-25 characters' length [5].

Turning directly to the description of the method, it is first necessary to clarify all the theoretical aspects that make it possible to effectively use elliptic curves for factorization purposes. Firstly, an elliptic curve is a set of solutions of the cubic equation which can be written into the general form as

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad (1)$$

where a_1, a_2, a_3, a_4, a_6 – coefficients from the field over which the curve is constructed [7]. Formally, a field is a set F together with two operations called addition $+$ and multiplication $\times$. Let $a, b \in F$, then an operation is a mapping that associates an element of the set to every pair of its elements. The result of the addition $a + b$ is called the sum. Similarly, the result of the multiplication $a \times b$ is called the product. These operations are required

to satisfy the following properties, referred to as field axioms:

1. Associativity of addition and multiplication $a+(b+c)=(a+b)+c$ and $a\times(b\times c)=(a\times b)\times c$.

2. Commutativity of addition and multiplication $a+b=b+a$ and $a\times b=b\times a$.

3. Additive and multiplicative identity: there exist two different elements 0 and 1 in F such that $a+0=a$ and $a\times 1=a$.

4. Additive inverses: for every a in F, there exists an element in F, $-a$, called additive inverse of a, such that $a+(-a)=0$.

5. Multiplicative inverses: for every $a\neq 0$ in F, there exists the element a^{-1}, called the multiplicative inverse of a, such that $a\times a^{-1}=1$.

6. Distributivity of multiplication over addition: $a\times(b+c)=a\times b+a\times c$.

The ring is a set similar to the field, with the difference that commutativity of multiplication, multiplicative identity and inverse axioms, are not satisfied.

In the case of constructing a curve over a set of rational numbers, or over fields whose characteristic is different from 2 and 3, this equation can be simplified to $y^2=x^3+ax+b$, this kind of equation is called the Weierstrass form. This curve must be nonsingular and include point at infinity. The arithmetic of elliptic curves allows us to state that if n – is a prime number the point at infinity means an unique additional projective point on an elliptic curve that does not correspond to any affine point. If n is composite number, then there are other projective points, which do not correspond to any affine point. Nevertheless, we will allow only one additional point that corresponds to the projective solution [0;1;0]. Due to this limitation in the definition of the elliptic curve group, the elliptic curve no longer forms a group with a composite n. It is easy to prove that there are pairs of points P and Q, for which the sum $P+Q$ – is undefined. This explains by the structure of the angular coefficient:

$$m=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1}, & if\ \ x_1\neq x_2\\ \dfrac{3x_1^2+a}{2y_1}, & if\ \ x_1=x_2\end{cases}, \qquad (2)$$

where $P=(x_1,y_1),\ Q=(x_2,y_2)$.

The above results carry over to the elements of the set $E_{a,b}(Z_n)$, which differ from elliptic curves in the case when n – is a composite number [5]. In this case, the concept of elliptical pseudo-curve is used. This curve defines by the conditions:

1. $a,b\in Z_n$;
2. $G.C.D.(a,b)=1$;
3. $G.C.D.(4a^3+27b^2,n)=1$;
4. $E_{a,b}(Z_n)=\{(x,y)\in Z_n\times Z_n : y^2=x^3+ax+b\}\vee\{O\}$, where O – infinitely corresponded point.

In a strict mathematical formulation, this curve is not considered as an elliptic curve (such a curve is also called pseudo-curve), since F_p is not a field according to it the operations of finding the inverse element that are necessary to find the sum of the points of the curve are not always feasible in it. It goes from the impossibility of calculating the sum of two points $P(x_1,y_1)$ and $Q(x_2,y_2)$. It turns out that the difference between the first coordinates x_2-x_1 must be equal to zero modulo one of the n divisors. Thus, computing the greatest common divisor G.C.D.(n,x_2-x_1), gives a divisor of the given composite number [8]. Lenstra's algorithm is based on an arbitrary base point P_0 and pseudo-curve $E_{a,b}(F_p)$ selection and on its subsequent multiplicity by various prime numbers and their degrees until get:

$$kP_0=\infty(\mathrm{mod}\ p), \qquad (3)$$

where p – is one of divisors of n.

In addition, there is the possibility of the divisor obtaining as G.C.D. of the curve discriminant and factorized number. However, while all the features of the discriminant that allow to receive the divisor in this way are unknown, this question remains an extremely important for research.

Since, none of the composite number divisors is known, it is not possible to check whether condition (3) is fulfilled. On this basis, a sign of the algorithm successful completion is that the condition G.C.D. $(n,c)=d>1$ is fulfilled when calculating the angular coefficient is done [5].

The idea of algorithm was proposed by Pomerance [5], but its realization in such form, was made for first time. The algorithm can be represented in the following form:

The input is a composite number n, which must be decomposed into prime factors.

1. The limit of the first stage b_1 is selected.
2. A random curve $E_{a,b}(Z_n)$ and a point on it with coordinates (x,y) are generated.

Moreover, $b=(y^2-x^3-ax)\bmod n$ and $g=G.C.D.\left(n,4a^3+27b^2\right)$. Further, if $g=n$, then we have to return to the curve generation. If $1<g<n$, then a divisor is found.

3. For every prime number $p<b_1$ the greatest degree is determined α_i such that $p_i^{\alpha_i}\le b_1$. Then a loop is executed for all $j=1:\alpha_i$, $P=p_iP$, as a result of which the point P multiplies by p^{α_i}. Each multiplication by p is performed using the elliptic multiplication algorithm: the addition-subtraction scheme [5].

3. Method analysis

Typically, the number $n\in N$ that fed into the number of performed arithmetic operations is estimated by the value: $L_p\left[\frac{1}{2};\sqrt{2}\right]$ at L-notation, where p – is smallest divisor of n [5].

The important result of the elliptic curves theory is the Hasse theorem. According to this theorem, the following assertion is true: the power of $E_{a,b}\left(F_{p^k}\right)$ is satisfying the inequality: $p+1-2\sqrt{p}<\#E_{a,b}\left(F_{p^k}\right)<p+1+2\sqrt{p}$, where $\#E_{a,b}\left(F_{p^k}\right)$ – is the number of elliptic curve points, or in other words the power of a given curve, or the order of this curve [7].

Until now, all calculations have been performed by modulo of factorized number. In case when the coordinates of the obtained points are calculated by modulo p, which is a divisor of n, we get the following condition for the successful completion of the algorithm: $kP=\infty$, $k=\prod_{p_i^{\alpha_i}\le B_1}p_i^{\alpha_i}$ .In this case the curve $y^2=x^3+ax+b$ is constructed over the field F_p [5]. Let $l=\#E\left(F_p\right)$ is the number of curve points. Then according to Hasse theorem $l\in\left\lfloor p+1-2\sqrt{(p)},p+1+2\sqrt{(p)}\right\rfloor$. According to the fact that for every point $Q(x,y)$ the condition $lQ=\infty$ is satisfied then, in order to ECM method to be successfully completed, it is necessary that factor k in equation (3) is divided by the order of the curve l. The last condition is satisfied in the case when, all dividers of l do not exceed the boundary b_1 [5].

Since the Lenstra's method is a direct descendant of the *(p-1)*-Pollard's method, it also has an extension in the form of the second stage [7]. Which is to use the points obtained in the first stage, with further multiplication by prime numbers, whose values exceed the boundary of the first stage [7].

For the successful completion of an algorithm with two stages, it is required that all dividers of l, except the greatest one, were less than the boundary of the first stage, and the greatest divisor had degree $\alpha=1$ and was less than the boundary of the second stage. This condition is less strict, but it is characterized by all the same problems as in the case of one stage algorithm. Besides, this optimization, in addition to increasing of the method convergence, leads to the increasing of the computational costs for each new generation of the curve [7].

Thus, the necessary boundary for the degrees of l divisors strongly depends on the value of $\#E_{a,b}\left(F_p\right)$, which is determinates by coefficients of elliptic curve a and b. At the moment the reliable algorithm for choosing a curve with the order divisor maximum degree smallest value is unknown [7].

It is important to research the probability of the certain b-smooth number in the Hasse interval finding. At the moment, it is not known whether there is always a smooth number in this interval. The L-notation that based on the heuristic probabilistic methods of the Canfield-Erdoes-Pomerance theorem, gives an estimate that in order to obtain a smooth order of the group it is sufficient to take $L_p[\frac{\sqrt{2}}{2};\sqrt{2}]$ curves [9].

On the other hand, there is a need to evaluate the order of the generated curve, to change curves until we get smooth one. The algorithm could be greatly speed up with an efficient curve order evaluation method. Since the generation of a curve is a low cost operation, all the computational complexity is related to the search for prime numbers at the given interval and then multiplying the points of the curve by the given primes and their degrees. The problem is that there is no algorithm for the pseudo-curve's order calculating. The existing Schoof's algorithm for the curve order calculating, in addition to its laboriousness and complexity in implementation, is intended for curves constructed over finite fields. Knowing the divisor of the composite number n, it is possible to compute the order of such pseudo-curve by using Schoof's algorithm. However, none of the divisors is known, moreover, the search for divisors is the goal of the Lenstra's method. Thus, the only way to solve this problem is a theoretical research of the elliptic curves structures and

their various classes. Also, according to empirical results, there is a definite relationship between the values of the curve parameters, for which the total required value of the first and second stage boundaries reaches its minimum, in case when using optimization in the form of the second stage [7].

Since the theoretical apparatus for the selection of these parameters has not been developed at the moment, the only way out is random generation of an elliptical curve by random selection of its parameters, and the most effective way of optimization is the parallel use of several curves. Because the value of the factor p is unknown, then the choice of boundary performed empirically that definitely degrades the reliability of the method practical convergence assessment.

In addition, it is important to remember about the possibility of obtaining a divider as a G.C.D. of curve discriminant and factoring number. Such cases for large n, are quite rare. However for $n < 10^{11}$, their frequency is sufficient for them to happen in a simple enumeration of curves. This approach for numbers of small dimension is often a much faster way of a number factoring. Nevertheless, it is also known almost nothing about the properties of an elliptic curve discriminant. In particular, the authors do not give clear estimates of the divider obtaining probability by a given method, and the rationale for this approach is not given. Nevertheless, the empirical results give reason to believe that for small n, such approach is more efficient in terms of numerical costs. This is partly due to the significant increase of the algorithm computational complexity with the increasing of boundary b_1. The main difficulties in this case are associated with an increase in the number of primes considered, and accordingly, number of operations during each cycle of curve and point on it generation with further point multiplication.

The most effective way to optimize the elliptic curves method is to use parallel implementation with distributed memory [10], when the same number is attempted to factorize by using many different randomly generated curves at the same time. Thus, it is possible to obtain almost linear acceleration [10]. Based on this, it becomes possible to use large computing powers based on cloud technologies provided by many services, such as Amazon, in order to speed up the factorization process.

4. The problem of the relationship between the generated curves number and the required boundary of the basic ECM

The aim of the work was to analyze the effective number of curves, after using of which the boundary that used in the elliptic curves method would increase. The question of the optimal number of curves after using which it would be necessary to increase the boundary remains open. Most authors do not give a clear indication of this, saying only that the boundary should be changed only if the process takes "too much" time [1]. Nevertheless, this moment should be investigated. The correct choice of this number can lead to a significant reduction in the time costs of the method software implementation, as shown further. This is due to the dependence of the computing costs amount on the value of used boundary.

It is important to estimate the dependence of the method calculation amount as a function of the chosen boundary. In the method it is necessary to look for prime numbers on the interval $[0, b_1]$. It requires $O(b_1 \log(\log b_1))$ arithmetic operations using the sieve Eratosthenes [11]. Then, according to the algorithm, for each found prime number p_i it is necessary to find the greatest degree α_i, such, that $p_i^{\alpha_i} \le b_1$, then multiply dot P by p_i in a cycle from 0 to α_i. The assessment of this phase is highly difficult. First, we need to know the number of primes on a given interval. This problem is fundamental not only for the given algorithm, but also for a set of mathematical sciences in general. The prime number theorem gives an approximate description of the distribution's asymptotic behavior, but this estimate is not exact. New problems arise, when try to estimate values of prime numbers degrees – α_i. Due to the uncertainty of the primes distribution and their values, this task is very labor-intensive. All this is complicated by the probabilistic nature of the algorithm, that is, in specific cases, the divisor can appear at any stage of multiplying the curve's dot by p_i, or not appear at all, because there is still a problem of b_1 - smoothness of the generated curve.

Proceeding from the foregoing, a theoretical assessment of the necessary relationship between the boundary and the number of curves that is necessary to increase it, even if it is possible, will be heuristic in view of the huge number of uncertainties. And because of the laboriousness of the definition, it makes sense to use an empirical estimate. Thus, in this work, it was decided to use the stochastic model of elliptic curve method for the research.

Software that modeling the factorization process based on the theory of elliptic curves was implemented. This software is based on the idea of the algorithm proposed by Pomerance and described

above. The input of the method software implementation is composite numbers consisting of the two prime numbers product with a size of $\sim 10^5$. For the representativeness of the obtained results, for each case, 10 such composite numbers were used as input, each of which was factorized 30 times. That is, for each choice of the curves number, after which it is necessary to increase the boundary, 300 tests were carried out. The cases with the initial boundary: 100, 30, 6, 2, 1 were considered. Such boundaries were chosen in the research process due to the gradual acceleration of the software with a decrease of the initial boundary.

Numbers were taken from the interval, in increments of 500, as the number of curves used, after which the boundary is increased. This is due to the fact that at a larger step the overall trend would not be displayed, since even the selected sample size does not provide a deterministic estimate of the efficiency of the algorithm. At step 500, obtained results fairly accurately describe the overall efficiency of the method, while the length of this interval is quite enough for a clear fixation of this trend.

The dependence of the time spent on the work of the software implementation, the final boundary and the number of curves used depending on the number of curves after which the boundary is increased was investegated. Results are reflected in Fig. 1, Fig. 2 and Fig. 3 respectively.

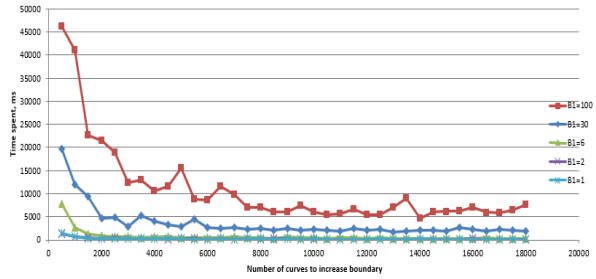

Figure 1 – The graph of the time consumed dependence on the number of curves, after which the boundary is increased

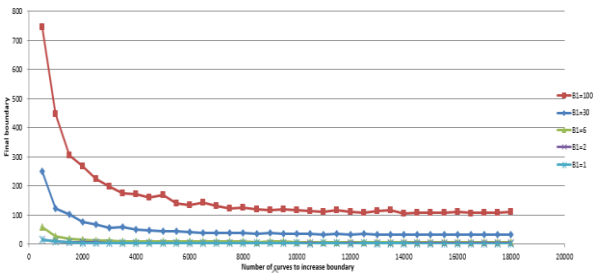

Figure 2 – The graph of the final boundary dependence on the number of curves, after which the boundary is increased

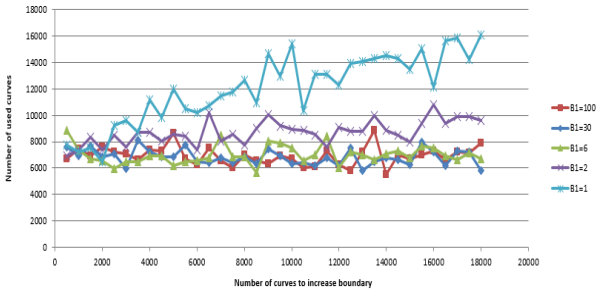

Figure 3 – The graph of the curves number used for factorization dependence on the number of curves, after which the boundary is increased

As can be seen from Figures 1 and 2, for all the cases considered, an increase in the number of curves needed to change the boundary led to a decrease in time costs, as well as to a decrease in the final boundary. In particular, the final boundary with the increase in the number of curves required for changing the boundary converged to its original value. That is, it did not increase when the program was executed. This indicates that, either or for numbers whose divisors consist of 5 decimal characters, it is almost always possible to get a curve of b_1-smooth order, or the fact that in this case the second condition for finding the divisor of n is satisfied. That is $1 < g < n$, where $g = G.C.D.\left(4a^3 + 27b^2\right)$. Thus the discriminant of the generated curve has a common divisor with a composite number n, different from 1.

It would be strange to obtain a curve of 1-, 2- or 6-smooth order, this is a very stringent condition, even taking into account the fact that the dividers have a relatively small value (for 1, it is impossible at all). Therefore, most likely for these cases, the second condition was satisfied. Based on this, the percentage of the cases number in which the second condition was satisfied was estimated (Figure 4).

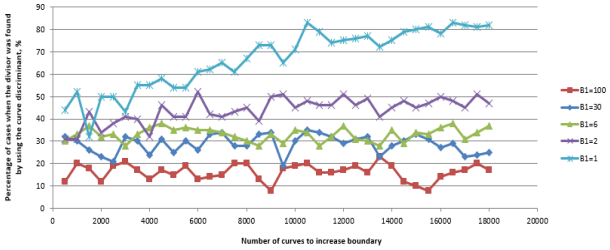

Figure 4 – The graph of the finding the divisor by a discriminant of curve dependence on the number of curves, after which the boundary is increased

Figure 4 shows, that for boundary $b_1 = 100$ percentage of such cases is small and fluctuates in a range from 8% to 20%. In case, when $b_1 = 30$ it fluctuates in a range from 19% to 34%, when $b_1 = 6$

– from 28% to 38%. Cases when $b_1 = 2$ and $b_1 = 1$ are special, because the probability of getting b_1-smooth order curve is the smallest, as evidenced by the highest percentage of cases when the divisor was detected using the discriminant of the curve. The percentage of such situations ranges from 31% to 52% and 32% to 83% respectively for $b_1 = 2$ and $b_1 = 1$. In addition, for the last two boundaries, it should be noted that the frequency of cases when the divisor was found using a discriminant increases with the number of curves, after which the boundary increases. This is due to the difficulty of obtaining a curve, the order of which will be b_1-smooth for given boundaries, with their permanence. If these boundaries remain unchanged, the probability that we will get a curve of b_1-smooth order, is much smaller, so we will rather find a curve with the suitable discriminant. This requires the use of a larger curves number, as shown in Figure 3. Thus, for cases $b_1 = 100$, $b_1 = 30$, $b_1 = 6$, the total number of curves used ranges from 6000 to 9000, and their average value is close for all three cases (6900, 6800 and 7000 curves respectively). In the last two cases, in addition to the fact that the average number of curves used is greater (8700 and 12000 for $b_1 = 2$ and $b_1 = 1$ respectively). There is also a direct relationship between the number of curves used and the curves number needed to increase the boundary.

Based on the obtained results, it is possible to form the following conclusions for numbers whose smallest divisor is less than 10^5. It is much more advantageous to increase the number of used curves than the boundary to reduce the time required for factorization process, even in cases of a minimal boundary. This is possible due to the probability that the discriminant of the curve has a common divisor with the specified composite number. Moreover, the best results were obtained, when the curves were generated until such situations happened. Thus, in cases where the initial boundary was taken as small as possible. This is indicated by the results obtained when $b_1 = 1$, was taken as initial boundary. In this case the percentage of these cases was about 80%, while the time costs were the lowest. This is due to a significant increase in the complexity of the algorithm with the growth of the boundary b_1, much greater than with increasing of the used curves number for given composite numbers.

However, we cannot exclude the cases when the curves whose order was b_1- smooth were received. Their results sometimes did not concede on the effectiveness of cases when a curve with the corresponding discriminant were received. These cases are also occured for $b_1 = 2$, and for $b_1 = 1$, providing a good executing time of program realization.

Also despite the result we can't really implement them very far. The reason is that we can't establish the fact that composite number contains divisors less than 10^5. For sure, even considering that this is a quite small step forward, it's still the step forward in the extremely difficult and fundamentally important scientific direction. Thus, it's difficult to underestimate the results of such kind of a research.

Conclusions

This work highlights the fundamental importance of the factorization problem for a number of both purely mathematical and applied sciences. The fundamental importance of this problem and its solution in number theory, signal theory, modern cryptography, in the construction of dynamic systems, etc. is substantiated. The paper gives a classification of existing approaches to solving this problem. Special attention is paid to the problem of a clear definition of "subexponential" complexity. Subexponential factorization methods are described and analyzed. The perspectivity of the method on the basis of the elliptic curves theory is substantiated, from which follows the significance of these objects and their research. It is indicated that the solution based on the elliptic curves theory is extremely effective. This is evidenced by both the subexponential character of the method, which depends on the smallest divisor of the number to factorize, and the existence of a many optimizations. Among these optimizations the most effective is the parallel implementation of the algorithm. Lenstra's method based on the theory of elliptic curves is described and analyzed in detail.

As a result of theoretical analysis and development of the elliptic curves theory a new algorithm was constructed. On the basis of this algorithm a software implementation of the elliptic curve factorization method was developed. The factorization process was simulated for numbers with divisors of a certain size by the created software. The purpose of this process was to research the dependence of the software implementation time costs, the final limit and the number of curves used, depending on the number of curves, after which the limit automatically increases. As a result of the algorithm computational analysis, it turned out that, due to the features of an elliptic curve constructed over a ring of an integers modulo composite number structure, for numbers of small size it is better to increase the

number of curves than the value of the boundary. This approach can significantly reduce the time costs. However, the important moment is that the size of composite number divisors a priori is unknown.

Despite all the advantages of this method, it also has gaps and difficulties. This work pays great attention to these moments. The problem of the method heuristic nature is analyzed in detail. This problem, in particular, is expressed in the random generation of curves. Further efforts will be directed to the research of the choosing a curve problem, and the influence of this choice on the process of different classes numbers factorization. In addition, an important task is the research of the curve discriminant features which allow to obtain the decomposition of a composite number.

References

1. Pomerance, C. B., (2008), Smooth numbers and the quadratic sieve, Cambridge University Press, New York: Algorithmic Number Theory MSRI Publications Volume 44, pp. 1–14.

2. Chernov, V. M., Chicheva, M. A, (2001), Algebraic arithmetic methods for the synthesis of fast algorithms for discrete orthogonal transformations [Algebraicheskie arifmeticheskie metody dlya sinteza bystryh algoritmov diskretnyh ortogonalnyh preobrazovanij], publishing house: Nauka, Moscow, pp. 301–384.

3. Complexity Zoo, (2008), Wayback Machine Class SUBEXP: Deterministic Subexponential-Time, available at: https://complexityzoo.uwaterloo.ca/Complexity_Zoo:S#subexp.

4. Regev, O., (2004), A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space, available at: https://arxiv.org/abs/quant-ph/0406151.

5. Crandall, R. E., Pomerance, C. B., (2001), Prime numbers: A Computational Perspective, Springer-Verlag, New York, pp. 293–420.

6. Lenstra, H. W., (1987), Factoting integers with elliptic curves, Annual of Mathematics Volume 126, New-Jersey, pp. 649–673.

7. Ishmukhametov, Sh. T., (2011), Methods for the factorization of natural numbers [Metody faktorizacii naturalnyh chisel], Kazan. UN., Kazan, pp. 83–103.

8. Koblitz, N., (1996), Introduction To Elliptic Curves And Modular Forms, Second Edition, Springer-Verlag, New York, pp. 9–56.

9. Canfield, E., Erdős, P., Pomerance, C. B., (1983), On a Problem of Oppenheim concerning "Factorisatio Numerorum", Elsevier, Amsterdam, Journal of number theory Volume 17, pp. 29–36.

10. Efimov, S. S., Makarenko, A. V., Pykhteev, A. V., (2012), Parallel implementation and comparative analysis of factorization algorithms with distributed memory, [Parallel'naya realizatsiya i sravnitel'nyy analiz algoritmov faktorizatsii s raspredelennoy pamyat'yu], Omsk State University. F. M. Dostoyevsky, Omsk, Mathematical structures and modeling Volume 26, pp. 94–109.

11. Pritchard, P., (1987), Linear prime-number sieves: a family tree, Elseiver, Amsterdam, Sci. Comput. Programming 9:1 pp. 17–35.

МОДЕЛЮВАННЯ ПРОЦЕСІВ ФОРМУВАННЯ АЛГОРИТМІВ ФАКТОРИЗАЦІЇ НА ОСНОВІ ТЕОРІЇ ЕЛІПТИЧНИХ КРИВИХ

Дерменжи І. Д., Востров Г. М.

Одеський національний політехнічний університет, Одеса, Україна

Анотація. *У даній статті описується проблема факторизації. Вказується на фундаментальне місце даної задачі в ряді чисто математичних і прикладних наук. Обґрунтовується важливість подальшого розгляду різних підходів вирішення даної проблеми. Аналізуються структурні класи методів факторизації. Дається їх умовна класифікація на експоненціальні і субекспоненціальні на основі їх обчислювальної складності. Описано основні субекспоненціальні методи, докладно аналізується проблема чіткого визначення субекспонеціальної складності. Дається визначення L-нотації, на основі якої даються оцінки обчислювальної складності субекспоненціальних алгоритмів. Обґрунтовується вибір підходу на основі теорії еліптичних кривих, як найбільш перспективного. Детально описано і проаналізовано алгоритм еліптичних кривих. Ідея методу ґрунтується на побудові псевдокривої над кільцем лишків складеного числа. Завдяки цьому вдається отримувати ситуації, коли неможливо знайти зворотний елемент в заданому кільці при складанні двох точок кривої, що сигналізує про знаходження дільника. Головною особливістю методу є залежність його обчислювальної складності від найменшого дільника числа, що факторизується, а не від безпосередньо нього самого. Описано основні проблеми методу та можливі шляхи його оптимізації. Окрему увагу приділено проблемі евристичного характеру алгоритму і випадкової генерації кривих. У статті*

описується проблема співвідношення між числом згенерованих кривих і необхідною границею базового методу еліптичних кривих. Дослідження цієї проблеми виконано за допомогою програмної реалізації методу, на основі описаного алгоритму для чисел чиї дільники не перевищують п'ять десяткових знаків. Представлені результати цього дослідження для різних випадків початкової границі. Отримані результати вказують на залежність витрачаємого часу, кінцевої границі при процесі факторизації і кількості кривих, що використовуються під час факторизації від кількості кривих після якого збільшується границя методу еліптичних кривих. Внаслідок аналізу цього методу, проведено дослідження кількості випадків отримання дільника складеного числа, за допомогою дискримінанту кривої.

***Ключові слова**: факторизація, еліптична крива, одностороння функція, експоненціальна складність, субекспоненціальна складність, гладкі числа, складені числа, псевдокрива, кінцеве поле.*

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФОРМИРОВАНИЯ АЛГОРИТМОВ ФАКТОРИЗАЦИИ НА ОСНОВЕ ТЕОРИИ ЭЛЛИПТИЧЕСКОЙ КРИВЫХ

Дерменжи И. Д., Востров Г. Н.
Одесский национальный политехнический университет, Одесса, Украина

***Аннотация.** В данной статье описывается проблема факторизации. Указывается на фундаментальное место данной задачи в ряде чисто математических и прикладных наук. Анализируются структурные классы методов факторизации, обосновывается выбор подхода на основе теории эллиптических кривых. Подробно описан и проанализирован алгоритм эллиптических кривых. В статье описывается проблема соотношения между числом сгенерированных кривых и необходимой границей базового метода. Исследование этой проблемы произведено с помощью программной реализации метода. Представлены результаты этого исследования.*

***Ключевые слова:** факторизация, эллиптическая кривая, односторонняя функция, экспоненциальная сложность, субэкспоненциальная сложность, гладкие числа, составные числа, псевдокривая, конечное поле.*

Received: 15.11.2018.

George Vostrov, Ph. D. of Technical Sciences, Associate Professor of the Department of Applied Mathematics and Information Technologies, Odessa National Polytechnic University. Shevchenko ave., 1, Odessa, Ukraine.
E-mail: vostrov@gmail.com, mob. +380503168776
Востров Георгій Миколайович, кандидат технічних наук, доцент кафедри прикладної математики та інформаційних технологій Одеського національного політехнічного університету. Проспект Шевченко, 1, Одеса, Україна.
E-mail: vostrov@gmail.com, тел. +380503168776
ORCID ID: 0000-0003-3856-5392

Ivan Dermenji, Student of the Department of Applied Mathematics and Information Technologies, Odessa National Polytechnic University. Shevchenko ave., 1, Odessa, Ukraine.
E-mail: ivandermenji97@gmail.com, mob. +380965824211
Дерменжи Іван Дмитрович, студент кафедри прикладної математики та інформаційних технологій, Одеського національного політехнічного університету. Проспект Шевченко, 1, Одеса, Україна.
E-mail: ivandermenji97@gmail.com, тел. +380965824211
ORCID ID: 0000-0003-0421-3372