UDC 681.3.06.

ALGORITHMIC PROCESSES OF THE LARGE NUMBERS FACTORIZATION BASED ON THE THEORY OF ELLIPTIC CURVES

G. Vostrov, I. Dermenji

Odessa national polytechnic university

Abstract. *In this article we consider the problem of the composite numbers factorization. Various methods for solving this problem were described and also their comparative characteristics were given. The Lenstra method algorithm was analyzed and described in detail. The ways of its optimization were given.*

Key words: *cryptosystem, factorization, elliptic curve, smooth numbers, composite numbers, pseudo-prime numbers, pseudo-curve, finite field.*

Introduction

Several approaches to solving the problem of decomposition of numbers into prime factors have been formed at the present time [1]. The methods variety can be divided into two classes: exponential and subexponential algorithms. Significant efforts of researchers are focused on this problem because its solution is important for both from the theoretical and applied points of view. Many areas of mathematics and computer science have been brought to bear on the problem, including elliptic curves, algebraic number theory, and quantum computing.

The solution of many mathematical problems in the mathematical theory of numbers, algebra, function theory and the theory of dynamical systems is associated with the assumption that the decomposition of the studied classes numbers is already known. This applies to the large Fermat theorem and its generalizations to the problem of solving the discrete logarithm problem, the theory of recursion in algebra, the theory of finite fields, and the theory of finite groups [2, 3]. An important trait is that a 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. On the other hand, the need of effective factorization methods follows from the modern theory of modeling complex dynamic systems [2], from the methods for constructing pseudo-random number generators and also from the deepening Monte Carlo methods. It is especially relevant in modern theoretical and applied cryptography [3, 4].

The need of numbers factorization often arises in cases of solving many mathematical problems. It determines the mathematical significance of creating an algorithmic effectiveness method. In the elementary number theory a number of problems, the solution of which is connected with the necessity of expansion of natural numbers into prime factors can be identified. This applies to pseudo-prime numbers. Number n is usually called pseudo-prime on the base a, if the following condition is satisfied: $G.C.D.(a,n)=1$ and $a^{n-1}\equiv 1(\mathrm{mod}\, n)$, i.e. the small Fermat theorem holds. Such numbers are also called Carmichael numbers [1].

Factorization problem due to pseudo-prime number theory

According to Manin's definition every pseudo-prime number n is odd and is a product of odd primes $n=\prod_{i=1}^{k} p_i^{\alpha_i}$, where p_i belong to odd classes of numbers and $p_1 > 2,..., p_k > 2$, and also the following conditions are satisfied:

1. $G.C.D(a,n)=1$;
2. $a^{n-1}\equiv 1(\mathrm{mod}\, n)$;
3. $a^{\varphi(n)}\equiv 1(\mathrm{mod}\, n)$;
4. $n=\prod_{i=1}^{k} p_i^{\alpha_i}$, $p_i > 2, i=1,...,k$

Conditions 1-3 correspond to definition and condition 4 corresponds to the Euler theorem [2]. In this case it is always necessary to be able to calculate the Euler function $\varphi(n)$ [5].

$$\varphi(n) = (p_1^{\alpha_1} - p_1^{\alpha_1 - 1_1}) \ldots \left(p_1^{\alpha_k} - p_1^{\alpha_k - 1_1}\right).$$

It is proved that there is the infinity of pseudo-prime numbers. Moreover, on page 23 of [3], the author notes that for the number n that is free from squares, the condition that $p-1$ divides the value $n-1$ (for all p coprime to n) – is equal to the fact that n is the Carmichael's number. This is an important property of pseudo-prime numbers. However, the concept of pseudo-primality depends on the value a. All pseudo-prime numbers in the base of a are in odd classes due the classification of numbers that based on the Fermat's theorem. However, the law of their distribution between classes requires to be analyzed.

Simultaneously, the pseudo-primality format is considered by Euler pseudo-prime numbers n in the base of a as pseudo-primality format, where $G.C.D(a,n)=1$ and the following condition is fulfilled:

$$a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) (\mathrm{mod}\, n) \quad (1),$$

where $\left(\frac{a}{n}\right)$ - the Legendre symbol by definition:

$$\left(\frac{a}{n}\right) = \begin{cases} 0, if\ a \equiv 0 (\mathrm{mod}\, n) \\ 1, if\ a \equiv b^2 (\mathrm{mod}\, n), b \in (Z/nZ)^* \\ -1, else \end{cases} \quad (2).$$

It was found that if for all a such that $a \in (Z/nZ)^*$, where $(Z/nZ)^*$ - is a multiplicative group of integers by modulo n, and the condition (1) is true, then the number n - is prime in contrast to pseudo-prime Carmichael numbers. This fact follows from the Chinese remainder theorem applied to the multiplicative group $(Z/nZ)^*$. The evidence is found in favor of the fact that the smallest number a, for which condition (1) is not satisfied does not exceed $2\log n \log\log n$ [4].

For these reasons (1) is a base of algorithms for distinguishing the primality of a number n. The decomposition of n into prime factors is much more complicated in contrast to the primality test algorithms of natural numbers.

Factorization problem in cryptography

Factorization of large odd composite numbers is an extremely significant part of cryptography due to the fact that numerous algorithms are based on the complexity of this task. This circumstance is fundamental for a variety of cryptographic algorithms and serves as a defense against potential attempts of hacking: for creation and multiplication of two prime numbers of large digit capacity no significant expenses are required, while their factorization is much more time-consuming and labour-intensive process. This fact defines the factorization operation to be the candidate for the one-way functions. For a long time there was no known method of factoring integer numbers that consists of more than 30 digits. Due to this, the use of natural number as the value of n, that consisting of more than 100 decimal places, provides a guarantee for the security of cryptosystem encryption based on the RSA algorithm. The algorithm is used in a variety of cryptographic applications among which: PGP, S/MIME, TLS/SSL, IPSEC/IKE and others [5].

Currently, the RSA-key length of 1024 bits or more per length should be considered as the trustworthy encryption system. The keys of 1024 bits in length will no longer be safe in the next three years [6].

The RSA algorithm is based on the difficulty of obtaining a secret key by an attacker with a known public key. Keys are obtained as follows:

1. Two random primes are generated p and q.
2. Calculating $n = pq$.
3. Calculating $\varphi(n) = (p-1)(q-1)$, where $\phi(n)$ - Euler function to n.
4. Choosing integer e, such that $1 < e < \varphi(n)$, e - is called an open exponent.
5. Calculating number d, multiplicatively inverse to the number e modulo n, i.e: $de \equiv 1 (\mathrm{mod}\, \varphi(n))$, d - is called a closed exponent.

An attacker can easily find a secret exponent and thereby hack RSA by knowing the decomposition of the module n into the product of two prime numbers. At the present time any effective non quantum algorithm for factoring integers is unknown. However, there is also no proof that this problem can't be solved in polynomial time. Conse-

quently, the creation of new one and the improvement of already existing factorization algorithms is an extremely important task of cryptography.

Classification of factorization methods

Typically, the number $n \in N$ that fed into the factorization algorithms must be decomposed into prime factors consisting of $N = \lfloor \log_2 n \rfloor + 1$ digits (if n represented in binary form). The algorithm organizes a search for one prime divisor, after which it is possible to start the algorithm again with new data for the purpose of further factoring, and so on until we expand the given number finally. Also, before starting the factorization of a large number it should be checked whether it is not prime. There are many primality test algorithms to do this [7]. This problem is solvable deterministically in polynomial time [7].

Despite a longer working time, exponential algorithms still have a place to be considered, and their analysis and improvement are important tasks of modern mathematics. There are a number of reasons for this such as:

- Exponential algorithms are used for factoring small numbers. So, for example, in some subexponential methods, small auxiliary numbers are decomposed by using exponential methods [8].
- Subexponential algorithms are often direct descendants of exponential ones. Exponential algorithms are important for further research in order to develop both new and improve existing algorithms [8].
- Exponential algorithms are strictly analyzed and determined in contrast to subexponential [8].
- In solving the problem of the discrete logarithm one of the basic operations is the factorization, thus factoring algorithms are their basis. For numerous algebraic groups, only exponential algorithms are known to calculate the discrete logarithms [8].

The aim of the algorithm is to decompose a large odd composite number into factors in a minimal average time. An important point is the computational complexity of the algorithm, the complexity of its implementation, and the amount of resources spent in its work in relation to the complexity of the problem.

Pollard algorithm and its analysis

It is important to consider the exponential Pollard's $(p-1)$ method, which in fact, is the ancestor of ECM method. The Lenstra algorithm is a complete analog of $(p-1)$ Pollard algorithm, where the operation of raising to the power of a prime number p is replaced by the operation of multiplying the point of the elliptic curve by a factor p. Otherwise, the organization of the first and second steps may be performed completely analogously to the $(p-1)$ – method.

His idea is described by the following algorithm. Let n — is the number that is factorized and $1 < p < n$ – its primal divisor. According to Fermat's little theorem, for any a, $1 \le a \le p$, the condition $a^{p-1} \equiv 1 (\mathrm{mod}\, p)$ is satisfied. It is also satisfied if instead of the degree $p-1$ take an arbitrary natural number m multiple to $p-1$, so that if $m = (p-1)k$, then $a^m = (a^{p-1})^k \equiv 1^k \equiv 1 (\mathrm{mod}\, p)$. The last condition is equivalent to $a^m - 1 = p^\alpha$ for some integer α. For this reason, if p is a number divisor of n, so then p is a divisor of the greatest common divisor $G.C.D.(n, a^m - 1)$ and coincides with it, if $a^m - 1 < n$. Let $p-1 = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_t^{\alpha_t}$. Idea of $(p-1)$ Pollard method is to select m in the form of a product of the greatest possible number of prime factors or their degrees so that m be divided by each factor $p_i^{\alpha_i}$, included in the expansion. Then, $G.C.D(n, a^m - 1)$ gives the desired divisor. The algorithm consists of two stages.

At the first stage, it is important to use the concept of a "smooth" number. 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, i.e. none of the prime factors of the number n does not exceed a number b. Further, the number $m(b) = \prod_i p_i^{k_i}$ must be calculated, where the product is over all simple p_i in the maximum degrees $k_i : p_i^{k_i} < b$. In this case, the required divisor is $q = (d-1, n)$, where $d = a^{m(b)} (\mathrm{mod}\, n)$ [7].

The second stage fixes boundaries $b_1 = b, b_2 >> b$, usually $b_2 \le b^2$ [7]. Next step is the search dividers n, such that $p - 1 = qa$, where a - is b-smooth, and q - primal, such that $b_1 < q < b_2$. Then we use a vector of primes q_i from b_1 till b_2. Then sequentially calculated $c_0 = d_1 (\mathrm{mod}\, n)$, $c_i = c_{i-1}^{\delta_i} (\mathrm{mod}\, n)$, where $d_1 = a^{m(b_1)} (\mathrm{mod}\, n)$, that was calculated at the first stage, at each step counting $g = (c_i - 1, n)$ and

$\forall \delta_i \in \Delta$, where Δ - finite set. When $g \neq 1$, calculations stop.

According to calculations of the algorithm running time from work [8], provided that the parameters b_1 and b_2 that are the limits of the first and second stages are chosen. The execution time of the first stage depends on the number of prime numbers and their degrees on the interval $[2;b]$. For each degree p^{α}, less than b, produced α modular exponentiation and require $\log_2 p \le \log_2 b_1$ squaring operations and multiplications modulo the number n. The total number of operations is estimated by the amount $O(b_1 \log b_1 \log^2 N)$. The method finds prime factors of small and medium size (less than 20-25 decimal digits) very quickly. According to Montgomery, the second stage of the algorithm requires $O(\log^2 b_2) + O(\log q_{\pi(b_1)}) + 2(\pi(b_2) - \pi(b_1))$ multiplications by modulo n and computing G.C.D. with n. Without considering terms of smaller order, the estimate will be $O(\pi(b_2))$ [8].

Quadratic sieve and numeric field sieve methods overview

The quadratic sieve method, the numeric field sieve method and the elliptic curve method are the most effective among the subexponential factorization algorithms.

The scheme of the algorithm for the Pomerance quadratic sieve is presented in the following form. The first step is constructing $x^2 = y^2 (\mathrm{mod} n)$, then checks the validity of inequality: $1 < G.C.D.(x \pm y, n) < n$. For this purpose we consider the polynomial $Q(x) = \left(x + \left[\sqrt{n}\right]\right)^2 - n = H(x)^2 (\mathrm{mod} n)$, where $H(x) = x + \left[\sqrt{n}\right]$. Values of $Q(x)$ at integer points are squares modulo n. Coefficients $Q(x)$ small, close to $n^{1/2}$. As a factor base S is considered $p_0 = -1$ and all primes $p_i, p_i \le b$ such that $\left(\frac{n}{p_i}\right) = +1$. After that, calculating the values x_i using some sieving, wherein x_i is the value for which $a_i = Q(x_i) = \prod_{p \in S} p^{\alpha_i}$ is true, thus $Q(x_i)$ decomposes in the factor base [7]. Consequently, denoting $b_i = H(x_i)$, concluding $b_i^2 = a_i (\mathrm{mod}\, n)$, accumulated a sufficiently large number of such relations, the elimination of variables is performed and the equation $x^2 = y^2 (\mathrm{mod} n)$ can be defined [7].

The method of a quadratic sieve with the use of several polynomials is an efficient and easily implemented computer algorithm. It is the best known algorithm for the factorization of arbitrary numbers $n \in N$, $n < 10^{110}$, except for the method of factorization by means of elliptic curves, which in some cases can work faster [7]. Algorithm requires $O\left(e^{\sqrt{(1+o(1))\ln n \ln\ln n}}\right)$ arithmetic operations by adopting a number of hypotheses about the distribution of prime numbers as a function of the number n [7].

The number field sieve is the most effective factorization method for large numbers at the current time. In fact, the number field sieve is not an algorithm. This is a calculation method that consists of several stages, which are served by several algorithms. Detailed description of SNFS and GNFS are very extensive and all the principles and basics of these algorithms are described in details in the [9]. The complexity rating is equal to $L_n[1/3;c]$ for some constant c [7].

Lenstra method overview

We can distinguish the method of elliptic curves, which is the most perspective among the subexponential algorithms. It is based on the theory of elliptic curves. Lenstra has created this method of decomposition into prime factors and algorithms obtained on it suggest that for its further improvement it is possible to create a much more efficient algorithm than the base ECM. An important feature of this algorithm is that its performance depends not on the number n itself but on the smallest factor only. This point is crucial when using a method of elliptic curves, since it opens new opportunities to use it in complex with other factorization algorithms. Such as the quadratic sieve method, which is also subexponential, but it works faster for numbers which divisors are large enough. The Lenstra method is the best algorithm for finding simple divisors of 20-25 characters per length [4]. However, in the case of using RSA, this fact is a significant drawback of this method, due to the fact that the RSA algorithm is based on multiplying of two large primes. Accordingly, the Lenstra method in this case is not optimal for the cryptosystem based RSA attack.

From the ideological point of view, the method of elliptic curves was created for the solution of certain classes of problems in classical mathematical analysis. From a conceptual point of view, elliptic curves can be considered over finite fields F_p, when

$p > 3$ - is primal odd number. The first results in this area were focused on solving problems in abstract and numerical number theory. Later they were generalized to the case of fields of the form F_{p^k}, when $k > 1$. A lot of important results were obtained for fields F_{2^k}.

One of the valuable and laborious problems in the theory of elliptic curves is the problem of computing the order of the group of an elliptic curve over a finite field. The set of points of an elliptic curve $E_{a,b}(F_p)$ [10] for any finite field is given by:

$$\left| E_{a,b}(F_p) = p + 1 + \sum_{x \in F_p} \left(\frac{x^3 + ax + b}{p} \right) \right| \quad (3).$$

The computation of the elements of this set for large p represents in the computational sense as a very labor-consuming problem. In the above expression $\left(\frac{x^3 + ax + b}{p} \right)$ is a Legendre's symbol. The set $E_{a,b}(F_p)$ consists of points $(x, y) (\mathrm{mod}\, p)$ that lie on the given elliptic curve and one another correspond to an infinitely distant point [10]. The Hasse theorem is an important result of the theory of elliptic curves. According to this theorem, the following assertion is correct: power of $E_{a,b}(F_{p^k})$ satisfies the inequality:

$$p + 1 - 2\sqrt{p} < \# E_{a,b}(F_{p^k}) < p + 1 + 2\sqrt{p},$$

where $\# E_{a,b}(F_{p^k})$ - the number of points lying on a given elliptic curve, or in other words the power of a given curve, or the order of this curve.

The arithmetic of elliptic curves allows us to state that if n - is a prime number the point at infinity means a 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 pseudo-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:

$$m = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, if\ x_1 \neq x_2 \\ \dfrac{3x_1^2 + 2cx_1 + \lambda}{2y_1}, if\ x_1 = x_2 \end{cases},$$

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. 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.

The curve in this case has the following form $E(Z_n): y^2 = x^3 + ax + b$. 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 divisors of n, thus, computing the greatest common divisor G.C.D.$(n, x_2 - x_1)$, there is a divisor of the given composite number [9]. Lenstra's algorithm is to select an arbitrary base point and pseudo curve $E_{a,b}(F_p)$ P_0 and to multiply it subsequent by various prime numbers and their degrees until they get:

$$kP_0 = \infty (\mathrm{mod}\, p), \quad (4)$$

where p – is one of the dividers of n.

Since, none of the divisors of n is known in advance, it is not possible to check whether condition (2) is satisfied. According to this fact, the sign of successful completion of the algorithm is the ful-

fillment of condition $G.C.D.(n,c)=d>1$ in the doubling operation or addition of points in the calculation of the next multiple c point P_0 when calculating the angular coefficient.

Many cryptographic applications are based on the theory of elliptic curves. Typically, the general idea in these applications is that a known algorithm which makes use of certain finite groups is rewritten to use the groups of rational points of elliptic curves. Elliptic-curve Diffie–Hellman algorithm, Elliptic Curve Digital Signature Algorithm, EdDSA, Dual EC DRBG, Elliptic curve primality proving, Supersingular isogeny key exchange and many other algorithms using elliptic curves.

ECM algorithm

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 chosen.
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 $g=G.C.D.(n,4a^3+27b^2)$. Further, if $g=n$, then we have to return to the curve generation and 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 [10].

Remarks to ECM

The computation continues until all the prime numbers that are less than b_1 will be passed, or until there will be no step on which condition $G.C.D.(n,P)=d>1$ is satisfied, that is signaling that $g=G.C.D.(n,d)$ is nontrivial divisor. If the last condition is fulfilled, then the desired divisor of n is found [11]. In another case, we increase b_1, or change the elliptic curve and repeat all over again.

With the help of implemented in practice program by using the computer-programming language Java, the following results were achieved: with the smallest divisor 4 decimal digits long, the average work time is equal to 6.985 seconds, in 3 decimal places 1.4 seconds, in 2 decimal places 0.166 seconds. In general, the results correspond to the subexponential estimation of the algorithm.

There are different cases in the operation of elliptic multiplication by pseudo-curve. The concrete case depends on the used chain of additions. So the point $5P$ can easily be calculated, when it is calculated by means of a chain $P\to 2P\to 4P\to 5P$. However penultimate elliptic addition may not be feasible, even if the calculate of this point provides by a chain $P\to 2P\to 3P\to 5P$. However, if two different chains that obtain kP by additions are achievable, then an identical result is obtained in both cases [10].

Until now, all calculations have been performed by modulo n, i.e. after each operation when the result exceeds the value of n, the remainder of the number dividing obtained by n was taken as a result. 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:

$$kP=\infty,\ k=\prod_{p_i^{\alpha_i}\le B_1} p_i^{\alpha_i} \quad (4)$$

and in this case the curve $y^2=x^3+ax+b$ is constructed over a finite field F_p. Let $l=\#E(F_p)$ is the number of points of this curve. According to Hasse theorem $l\in\left[p+1-2\sqrt{(p)},\ p+1+2\sqrt{(p)}\right]$. 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 (4) is divided by the order of the curve l. In the case when, all dividers l do not exceed the boundaries b_1, the last condition is satisfied. For the successful completion of the algorithm, it is required that all dividers l of kind p^{α}, except for the last one, were less than the border b_1, and the greatest divisor p^{α} had degree $\alpha=1$ and was less than the border b_2 [11].

ECM analysis

The required boundary for the degrees of divisors of l depends on the $\#E_{a,b}(F_p)$, which is determined by the coefficients of the elliptic curve a and b. At the moment, there is no known reliable

algorithm for choosing a curve with the smallest value of the maximum degree of the divisor of $\#E_{a,b}(F_p)$.

The number of performed arithmetic operations is estimated by the value: $e^{\sqrt{(2+o(1))\ln p;n(\ln p)}}\ln^2 n$, or $L_p\left[\frac{1}{2};\sqrt{2}\right]$ at L-notation, where p - the smallest divisor of a number n. L-notation, this is an asymptotic notation analogous to O-notation, that is used for the approximate estimation of computational complexity of the algorithm and is determined by the formula: $L_p[\alpha,c]=e^{(c+o(1))(\ln p)^{\alpha}(\ln\ln p)^{1-\alpha}}$, by $p\to\infty$, where $c=const$, $c\in(0;\infty)$ and $\alpha=const$, $\alpha\in[0;1]$.

Since the value of the factor p is unknown the choice of the value b_1 is performed empirically. That definitely worsens the reliability of the practical estimate of the method convergence. Adding the second stage of computations to algorithm preserves the general asymptotic estimate and at the same time provides a large practical increase in the rate of convergence of the method [11].

It is important to research [7] the probability of finding a certain b-smooth number in the Hasse interval (in this case there are curves with a smooth number order). At the moment, it is not known whether there is always a smooth number in the interval. The L-notation that based on the heuristic probabilistic methods of the Canfield-Erdoes-Pomerance theorem [8], gives an estimate that in order to obtain a smooth order of the group it is sufficient to take $L\left[\sqrt{2}/2;\sqrt{2}\right]$ curves.

According to frequent using of a random number generation (clauses 1 and 2 of the algorithm), it is critically important to use efficient pseudo-random number generators.

In addition, the algorithm uses a set of primes from the interval $[2;b_1]$. Proceeding from this, the question of finding prime numbers in a given interval remains to be opened (in the practical implementation of the method, sieve of Eratosthenes was used).

The question of correct curve generation is remains to be opened. Due to the algorithm, it generation has random a character and the condition $b=y^2-x^3-ax(\mathrm{mod}\,n)$ is satisfied. However, such an approach cannot be considered as acceptable. Firstly it is probable that the given curve will not be — b_1-smooth. It is known that the base algorithm will not be able to detect the divisor in that case. This disadvantage can be compensated by using the second stage of the extended ECM algorithm. However, such "optimization" leads to computational complexity increase. An important task is to identify the optimal classes of elliptic curves for a given algorithm. It is important to investigate the possible relationship between elliptic curves and modular forms. It is still unknown if they are similar or fundamentally different structures.

In comparison of the three considered subexponential methods (ECM, quadratic sieve QS method and NFS numerical field method), the dimension of the smallest divisor of the composite n is the main trait. In the case when the number n is chosen by the RSA method, which means that it is represented by the product of two prime numbers of the approximately same length, then the elliptic curve method has a similar estimate with the quadratic sieve method, but is inferior to the sieve method of the numeric field. In case when n has a dimension that exceeds the record for QS and NFS methods, the only way to find a divisor of n is factorization by Lenstra method [14].

Ways of the ECM optimization

Due the fact that the Lenstra method is a child of the $(p-1)$ in some way it also has an extension in the form of a second stage. It is assumed that the order of $\#E_{a,b}(F_p)$ is not equal to the smooth b_1, no matter what choice of the value b_1 is made, on the basis of this, the base algorithm will not be able to detect the divisor. However, it is quite possible that

$$\#E_{a,b}(F_p)=q\prod_{P_{i1}^{\alpha_i}\le B_1}p_i^{\alpha_i} \quad (5),$$

where q — is a prime number that greater than b_1. In the case where one out-of-bound prime number is a part of the unknown order factorization, there is no need to multiply the current point by every prime number from the interval $(b_1,q]$. Instead of this it's permitted to use point

$$Q=\left[\prod_{P_{i1}^{\alpha_i}\le B_1}p_i^{\alpha_i}\right]P \quad (6),$$

which, in fact, remains after the first stage of the ECM algorithm, and the points

$$[q_0]Q,[q_0+\Delta_0]Q,[q_0+\Delta_0+\Delta_1]Q,..., \quad (7),$$

are tested, where q_0 — is the smallest prime that exceeds b_1, and Δ_i — are differences between adjacent prime numbers after q_0. Some points

$$R_i = [\Delta_i]Q \quad (8),$$

are stored to be used repeatedly, after which the specified prime numbers greater than b_1 are verified, by successive elliptic additions using the corresponding R_i. The main advantage is that the multiplication of a point by a prime number q requires $O(\ln q)$ elliptic operations, while addition by using a pre-computed point R_i is the only operation [11].

Pollard $p-1$ algorithm is in fact the progenitor of the elliptic curves method [11], as was said earlier. The algorithm factorizes p, such that $(p-1)$- is a b-smooth for some small b. For any e, that is multiplicity to $(p-1)$, and for any a, mutually prime with p according to a small theorem of Fermat, $a^e \equiv 1 \bmod p$. In this case $G.C.D.(a^e-1, n)$ is more likely to be a divisor of n [11]. However, the method does not work [11], if p has large prime divisors. ECM in this case will work correctly, because the group of a random elliptic curve over a finite field F_p is used instead of considering a multiplicative group, over F_p, order of which is always equal $p-1$,.

In addition to this optimization in the form of a "second stage" and its variants, there are a number of other ways to optimize the Lenstra method proposed by Crandall and Pomerance [14] such as:

1. Special parameterization, in order to quickly obtain random curves.

2. Selection curves which orders are divided by 12 or 16 [15].

3. Optimizing the arithmetic of large integers and in particular elliptic algebras, as an option, by using a fast Fourier transform (FFT).

4. Fast algorithms applied to the second stage, for example "extended FFT", that is a scheme for calculating the values of a polynomial applied to sets of previously computed x-coordinates.

In addition, due to the parallel implementation of ECM with distributed memory [16], an almost linear acceleration can be obtained. Thus, it becomes possible to obtain a large amount of computing power with the help of cloud computing provided by a variety of services, such as Amazon [17].

Also, the correct choice of boundaries $b_1 \ldots b_n$ is important. In practice it makes possible to get the fastest running time of the algorithm. For the correct choice of such boundaries, the Brent's table [15] is used. In this table the recommended boundary values for close numbers of a certain digit are indicated.

Elliptic curves over finite fields are described in this work. The question about optimal structure of curve for this algorithm remains open. There are a great number of elliptic curve alternative representations. Such as: Hessian curve Edwards curve, Twisted curve, Twisted Hessian curve, Twisted Edwards curve, Doubling-oriented Doche–Icart–Kohel curve, Tripling-oriented Doche–Icart–Kohel curve, Jacobian curve, Montgomery curve and many others. Each of them should be considered.

The construction of families of elliptic curves over the rational numbers Q which have simultaneously nontrivial torsion and nontrivial rank is described in work [18]. These curves are then used to speed up the ECM algorithm. There also indicated a limited use of elliptic curves with complex multiplication.

However, the question about the security of cryptosystems based on the complexity of factorization remains to be opened. It does due to existence of quantum computers that realize the substantiated quantum Shore factorization algorithm that can solve the factorization problem in a polynomial time [19].

Conclusion

In total, you can get a significant reduction in the algorithm running time. Proceeding from the foregoing, it is quite obvious that, in spite of a fairly good estimate of the operation of the basic ECM algorithm. It has a truly great potential for improvement and is quite worthy of the title of one of the most promising factorization algorithms. At the same time, it is fairly simple to implement and clear to understand. Although the ideas of each of these methods are clear, however, their exact implementation is a very complex process. Moreover, it is important to combine all the proposed methods so that the algorithm to be the most effective. This work is the first step towards the creation and implementation of such an algorithm combining various ideas for optimizing the basic ECM method.

Summing up, we can say that the ECM algorithm, due to its subexponential nature, is well appli-

cable in practice and it should be investigated in more detail, due to its prospects in terms of optimization. It is necessary to direct as much effort as possible to the development of the Lenstra method for the reasons that the factorization problem is one of the fundamental in modern mathematics and number theory. It is important both in their theoretical aspects and in the applied sense.

References

1. Koblitz, N. (2001), A Course in Number Theory and Cryptography. – M.: Scientific Publishing House "TVP", – P. 188-200. – ISBN 5-85484-014-6.
2. Manin, Yu., Panchishkin, A., (2009), Introduction to the modern theory of numbers, [Vvedenie v sovremennuyu teoriyu chisel] - Moscow: MSC-MO.
3. Wagon, S. (1986), Primality Testing // Math. Intel, Vol.8, Num. 3, p. 58–61.
4. Vasilenko, O. (2003), Theoretical-numerical algorithms in cryptography, [Teoretiko-chislovyye algoritmy v kriptografii] - Moscow: MCCME, – ISBN 978-5-94057-103-2.
5. Bakhtiari, M., Maarof, M. (2012), Serious Security Weakness in RSA Cryptosystem // IJCSI — Vol.9, Iss. 1, No 3. – P. 175–178. – ISSN 1694–0814.
6. https://eprint.iacr.org/2010/006.
7. Ishmukhametov, Sh. (2011), Methods for the factorization of natural numbers: Textbook - Kazan: Kazan. UN. – 190 pp.
8. Vinogradov, I. (1952), Fundamentals of number theory [Osnovy teorii chisel] – 5-th ed. – M.–L.: State technical publishing, – 180 p.
9. Cohen, A. (2000), Course in Computational Algebraic Number Theory – 4th Print Edition – Berlin, Heilldberg, New-York: Springer. – 550 p. – (Graduate Texts in Mathematics) — ISBN 978-3-540-55640-4 — ISSN 0072–5285.
10. Crandall, R. E., Pomerance, C. B. (2001), Prime numbers: A Computational Perspective. — New York: Springer-Verlag, – 545 p. – ISBN 0-387-94777.
11. Brent, R. (1998), Some integer factorization algorithms using elliptic curves // Australian Computer Science Communications 8, 149–163 p.
12. Lenstra, H. (1987), Factoring integers with elliptic curves. *Annals of Mathematics*.
13. Lenstra, H. (1986), Elliptic Curves and Number-Theoretic Algorithms // Proceedings of the International Congress of Mathematicians. Thorsten Kleinjung, Kazumaro Aoki, Jens F.
14. Canfield, E., Erdős, P., Pomerance, C. (1983), On a Problem of Oppenheim concerning Factorisatio Numerorum // Journal of number theory. – Vol. 17.
15. Brent, R. (1999), Factorization of the tenth Fermat number. Mathematics of Computation 68.
16. Makarenko, A., Pykhteev, A., Efimov, S., (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.
17. http://www.ec2instances.info.
18. Atkin, A., Morain, F. (1993), Finding suitable curves for the elliptic curve method of factorization. Mathematics of computation vol. 60, number 201.
19. Shor, P. (1997), Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // Foundations of Computer Science: Conference Publications. – P. 1484.

АЛГОРИТМІЧНІ ПРОЦЕСИ ФАКТОРИЗАЦІЇ ВЕЛИКИХ ЧИСЕЛ, ЗАСНОВАНІ НА ТЕОРІЇ ЕЛІПТИЧНИХ КРИВИХ

Востров Г. М., Дерменжи І. Д.

Одеський національний політехнічний університет, Одеса, Україна

Анотація. *В даній роботі розглядається проблема факторизації великих складових чисел та її місце серед математичних та інформаційних наук, а також їх прикладних аспектів. Докладно описаний взаємозв'язок між теорією псевдопростих чисел і задачею розкладання числа на прості множники. Чітко відображена залежність сучасної криптографії від вирішення задачі факторизації, зокрема, фундаментальність даного питання для криптографічного алгоритму RSA на основі, якого створено велике число прикладних криптографічних програм. Дана класифікація сучасних методів декомпозиції чисел. Описані причини для дослідження кожного с класів. Приведено алгоритм метода Полларда і його аналіз, оскільки він являється деякого роду прародителем методу*

еліптичних кривих (метода Ленстри), який безпосередньо розглядається у статті. Серед субекспоненційних методів виділені: «Метод квадратичного решета», та «Метод решета числового поля», як одні з найшвидших, дана їх коротка характеристика, оцінка обчислювальної складності та порівняльний аналіз між собою та методом Ленстри. Детально описані основи метода Ленстри, а також ідеї на яких він базується, названі головні особливості математичних операцій на еліптичних кривих і властивості еліптичних кривих як математичних об'єктів, які надають можливість використовувати їх з цілю факторизації. Детально по крокам описаний сам алгоритм методу. Приведені результати розкладу великих складових чисел, отримані з допомогою реалізованої на практиці програми. Метод ретельно проаналізований, дана його обчислювальна оцінка, описані умови його збіжності. Названі фундаментальні проблеми алгоритму, які підлягають обов'язковому та найскорішому вирішенню, важливе місце серед яких займають: проблема вибору кривої, проблема генерації псевдовипадкових послідовностей, проблема пошуку гладких чисел. Викладені можливі варіанти оптимізації, зокрема, оптимізація аналогічна тій, що приводиться в методі Поларда у якості другої стадії. Поставлено питання щодо взаємодії таких способів оптимізації алгоритму та можливої реалізації. Підкреслена та обґрунтована перспективність методу еліптичних кривих в порівнянні з іншими сучасними методами факторизації. Описані пріоритетні шляхи вирішення проблеми факторизації.

***Ключові слова**: криптосистема, факторизація, еліптична крива, гладкі числа, складові числа, псевдопрості числа, псевдо-крива, кінцеве поле.*

АЛГОРИТМИЧЕСКИЕ ПРОЦЕССЫ ФАКТОРИЗАЦИИ БОЛЬШИХ ЧИСЕЛ, ОСНОВАННЫЕ НА ТЕОРИИ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Востров Г. Н., Дерменжи И. Д.

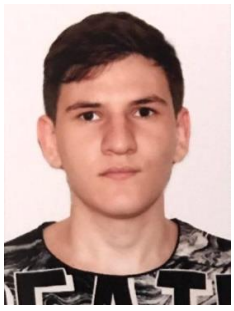
Одесский национальный политехнический университет, Одесса, Украина

***Аннотация.** В данной работе рассматривается проблема факторизации составных чисел. Были описаны различные методы решения этой проблемы, а также приведены их сравнительные характеристики. Алгоритм метода Ленстры был проанализирован и подробно описан. Приведены результаты его работы. Даны способы его оптимизации.*

***Ключевые слова:** криптосистема, факторизация, эллиптическая кривая, гладкие числа, составные числа, псевдопростые числа, псевдокривая, конечное поле.*

Received: 05.04.2018.

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