computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

RELIABILITY OF RSA ALGORITHM AND ITS COMPUTATIONAL COMPLEXITY

Mykola Karpinskyy [1]**, Yaroslav Kinakh** [2]

1) Professor, Universytet Bjelsku-Bjala, Poland, E-mail: mk@yahoo.com
2) Assistant, Ternopil Academy of National Economy
Institute of Computer Information Technologies, Department of Information Technologies Security
3 Peremoga Square, 46004, Ternopil, Ukraine, E-mail: kinakh@lycos.com

Abstract: *This article deals with the RSA encryption algorithm. Its safety is analyzed using the number field sieve method. The algorithm work results allow to define a define a secret key in a simple way.*

Keywords: *computer networks, RSA algorithm, encryption, number field sieve, factorization*

1 INTRODUCTION

The system of information encryption RSA was made in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman in fact, has become a world standard, which is realized as separate program products and within the products of applied program provision. RSA is used in the world bank circles, for example, for working with credit cards. It is used in such standards: SSL, S-HHTP, S-MIME, S/WAN, STT and PCT.

It is the best of times for the game of factoring large numbers into their prime factors. In 1970 it was barely possible to factor “hard” 20-digit numbers. In 1980, in the heyday of the Brillhart-Morrison continued fraction factoring algorithm, factoring of 50-digit numbers was becoming commonplace. In 1990 my own quadratic sieve factoring algorithm had doubled the length of the numbers that could be factored, the record having 116 digits. By 1994 the quadratic sieve had factored the famous 129-digit RSA challenge number that had been estimated in Martin Gardner’s 1976 Scientific American column to be safe for 40 quadrillion years (though other estimates around then were more modest). But the quadratic sieve is no longer the champion. It was replaced by Pollard’s number field sieve in the spring of 1996, when that method successfully split a 130-digit RSA challenge number in about 15% of the time the quadratic sieve would have taken. Today there are many people interested in factoring, recognizing it not only as a benchmark for the security of cryptographic systems, but for computing itself. In 1984 the Association for Computing Machinery presented a plaque to the Institute for Electrical and Electronics Engineers (IEEE) on the occasion of the IEEE centennial. It was inscribed with the prime factorization of the number 2251 - 1 that was completed that year with the quadratic sieve. The president of the ACM made the following remarks: About 300 years ago the French mathematician Mersenne speculated that 2251 - 1 was a composite, that is, a factorable number. About 100 years ago it was proved to be factorable, but even 20 years ago the computational load to factor the number was considered insurmountable. Indeed, using conventional machines and traditional search algorithms, the search was estimated to be about 1020 years. The number was factored in February of this year at Sandia on a Cray computer in 32 hours, a world record [7].

The amount of time it takes to factor a number of x bits is asymptotically the same as the time it takes to solve a discrete log over a field of size x bits. However in practice solving discrete log problems has been more difficult than factoring equivalent numbers. While there are several reasons for this, the main reason is that solving the matrix for a discrete log must be done using multi-precision integer arithmetic while the matrix for factoring is solved mod 2 and thus one can use simple bit operations. It has been estimated that for large x, one can break a discrete log of size x-30 in about the same time as factoring an x-bit number. Throughout this paper we shall assume that solving the two problems are equivalent under NFS and henceforth shall only discuss factoring.

The text is coded in the number system, suitable for work before encryption. The ciphered text is divided into the blocks $B_i \in Z_n$ transformed into the blocks according to the rules:

$$E(B_i) = B_i^e (\bmod n) \qquad (1)$$

Where e is an open key. Such is that e < $\phi(n)$ and the biggest common divisor (e, $\phi(n)$) = 1, $\phi(n)$ - Eiler's function, n is module of transmission, which is the multiplication of two, possibly "strong" prime numbers p and q, of which are big, as it is possible.

As a result we have received cryptotext, which is formed from the blocks $P_i = E(B_i)$. It is clear that $P_i \in Z_n$. The process of decryption is done according to the rule:

$$D(P_i) = P_i^d (\bmod n) \qquad (2)$$

d is a secret key here, which is connected with a secret key by such correlation:

$$ed \equiv 1(\bmod \phi(n)) \qquad (3)$$

During practical work with cryptotext the task of decryption occurs, when secret key d is unknown. As far as a secret and public keys are connected by such correlation, we may know a secret key with the help of public by calculating the function $\phi(n)$. It is known that $\phi(n)$ = (p -1) (q - 1) and the given task is put; as the task of p and q calculation, where pq=n, i.e the factorization problem is put. According to condition [1], having calculated tow square roots y and y' from x with module n, we can prove, that the biggest common divisor ($y + y', n$) is one of the divisors of p or q of number n from the condition, that $y \neq \pm y'(\bmod n)$.

Thus having solved the congruence $y^2 = (y')^2 (\bmod n)$, we can solve the given task. Among the most effective method of last congruence solution are quadratic sieve and number field sieve method. The last one is the most perspective one at present. The time of algorithm work, that carves the number field sieve method is estimated with the help of

$$\exp\left((c + o(1))(\ln n)^{1/3} (\ln \ln n)^{2/3}\right)$$

expression , c=1,526 [6].

The means of information protection within calculation system are analyzed in the dissertation the algorithms of protection lock construction and confidential information organization are described.

Thesis deals with the definition of the safety level of the cryptoalgorithm RSA and El-Gamala based on the usage of the algorithm Number Field Sieve (NFS) which provides confidentional and authentic information in computer networks. It suggests the choice polynom of the algorithm NFS for big numbers [5]. In the work the algorithm NFS is divided into algorithmic structures. Their decomposition is done and the optimization rulers of decomposition are formulated. The calculated system is offered which allows the algorithm NFS to be applied parallel. It shortens the full filament time of the algorithm in order to define the level of the reliability of the algorithm RSA, El-Gamala and use the safe keys in practice.

For the lattice sieve, a factor base bound of 16 777 216 (2^{24}) was chosen, both for the rational and for the algebraic side. Two large primes were allowed on both sides.

Most of the line sieve was carried out with two large primes on both the rational and the algebraic side. The rational factor base consisted of the primes 44000000 and the algebraic factor base of the primes 110000000.

After removing 55054854 duplicates from the siever output, we had 254033792 relations. After the pruning step, we had 110636597 useful relations.

The basic criteria for estimation of efficiency of parallel linear algebraic equations solving of great sizing HX=Q on N - processor computing system with single and conveyer command and data processor, when *i*-processor processes i submatrix as well as one of the processors solves the equitation subsystem *H*i, we can use the efficiency i of algorithm acceleration. The estimation is fulfilled according to the quantity of multiplicative operations, when dispersed matrix of coefficients *H* of output system of linear algebraic equations can be simplified to the block form on the basic of usage of the formalized operations of row and column processing. The spreading operation is the opposite operation to the general of compounding of the Gaus and makes an appropriate element processing of output coefficient matrix *H*. To use effectively the operation of spreading to the quasiblock matrixes, in which all non zero elements are grouped in the middle of allocated blocks of the main diagonal. Such matrix structures exist during practical use of methods of quadratic sieve of numerical field (NFS), when for forming of mathematical models canonic homogeneous or nonhomogenous basses are used. For the efficiency research of the algorithm of the parallel system processing of the linear algebraic equations such mathematical model is used

$$\xi_{Np} = \frac{N^3}{k_1N^5 - k_2N^5 + k_3N^4 - k_4N^3 + k_5N^2 - k_6N + 1} \cdot T^*_{ieуз}$$

де $k_1 = 8\alpha^3$;

$k_2 = 24\alpha^3$;

$k_3 = 3\alpha^2(8\alpha + 1)$;

$k_4 = 2\alpha^2(4\alpha + 3)$;

$k_5 = 3\alpha(\alpha + 1)$;

$k_6 = 3\alpha$;

$\alpha = \frac{c_i}{N}$ - coefficient of the connection of subsystem of linear algebraic equations;

c_i – width of i-block line and i-block column;

$i = \overline{1, N}$;

T^*_{step} - simple average of values of computing superblocks.

On the fig. 2 there are graphic results of numerical experiments, dependence of efficiency to the quantity of N i coefficient α using matrix operations of the method of general sieve of the numerical field. When N encrease acceleration of the algorithm increases, after which – it decrease. That is why every to coefficient value α corresponds the sane subsystem quandary, according to which the further increasing of processor quantity N does not give great advantage due to block paralleling of the system solving.

V

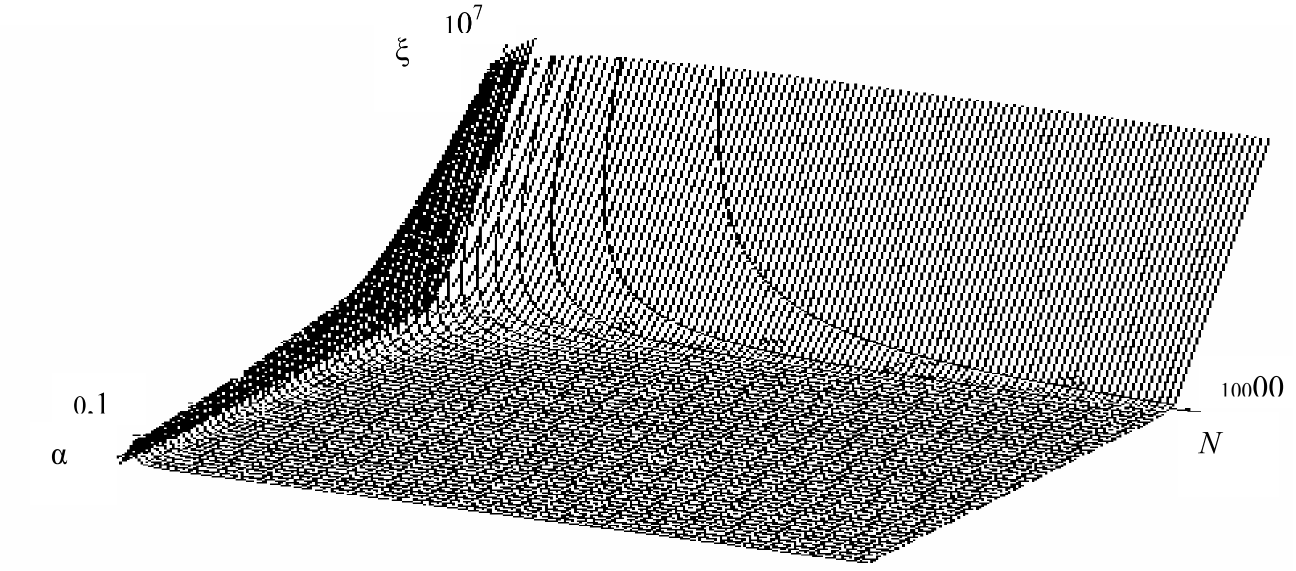

Fig.1 – The effectiveness function Graph.

$$\zeta\ 2 \cdot 10^{-4} \leq \alpha \leq 0.1$$

The main factors that cause the deviation of maximal acceleration, is the absence of maximum parallelism in the algorithm, which is used, as well as non balancing of processor loading of the computing system, and also are connected with delay data exchange; conflicts of storage devices and synchronization.

Let us consider which computing recourses are needed for the realization of the algorithm of general sieveof numerical field (NFS). In table 1 there are keys of coding system RSA and El Gamala and correspionding ammount of arythmetic operations, size of the factor base, needed minimal memory for the bolting stepi, as well as matrix operation memory volume .

The basis material efficiency is presented in fable 1.. It is impossible to present the operation time because it depends of hardware. For the key 1024 it is necessary to fulfill 52 mln. arithmetic operations more, and increase the accessible memory in 7200 times.

Table 1. The resources of computational system for NFS algorithm

Key length	Arithmetic operations	Factor base volume	Bolting step	Matrix operations
428	5,3·1017	550 Kbytes	23 Mbytes	130 Mbytes
465	2,4 1018	1,1 Mbytes	62 Mbytes	830 Mbytes
512	1,5 1019	3,2 Mbytes	130 Mbytes	2,1 Gbytes
768	1,2 1023	245 Mbytes	10,4 Gbytes	158 Gbytes
1024	1,4 1026	7,3 Gbytes	260 Gbytes	10,2 Tbytes

Some line sieving allowed three large primes instead of two on the algebraic side. In that case the rational factor base consisted of the primes 8000000 and the algebraic factor base of the primes 2500000.

The total amount of CPU-time spent on sieving was 35.7 CPU years estimated to be equivalent to approximately 8000 mips years.

The algorithm work results allow to define a secret key in a simple way.

REFERENCES

[1]R. L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public key cryptosystems, Commun. ACM. V.21. No 2. 1978. P. 120-126.

[2]H. Riesel. Prime numbers and computer methods for factorization, Birkhauser, 1985.

[3] H. Cohen. A course in computational algebraic number theory. *GraduateTexts in Math.* V. 138. New York, Springer, 1993.

[4] A. K. Lenstra, H. W. Lenstra (jr.). The Development of the Number Field Sieve, *Lect. Notes in Math.* V. 1554. Springer, 1993.

[5] M. Gardner. A new kind of cipher thet woud take millions of years to break, *Sci. Amer.* 1977. P. 120-124.

[6] On line access: http://www.nfsnet.org/.

[7] On line access: http://www.rsasecurity.com/.

[8] P. Montgomery. Parallel Implementation of the Block-Lanczos Method, *RSA-2000 Cryptographers Track*.

[9] R. Silverman, & S. Wagstaff Jr.. A Practical Analysis of the Elliptic Curve Factoring Method, *Mathematics of Computation,* vol. 61, 1993, pages [445-463].

[10] A. Odlyzko. The Future of Integer Factorization, *CryptoBytes 1* (1995) pp. 5-12.

[11] R. Silverman, & S. Wagstaff Jr.. A Practical Analysis of the Elliptic Curve Factoring Method, *Mathematics of Computation,* vol. 61, 1993, pages [445-463].

__Mykola Karpinsky__ was born on May 14'1958 in Baley, Chita region, Russia. Has graduated Electrical and Mechanical Department of Lviv Polytechnics Institute in 1980. Candidate of technical sciences since 1989, doctor of technical sciences since 1995, professor since 2001. Head of Department of Information Technologies Safety since 1991.

Scientific areas include: computer systems with emphasis on their defence, investigation in cryptography methods of information defence. Besides there were fulfilled researches in computer engineering, electrical engineering and industrial electronics made to order of Ministry of electrical industry.