UDC 004.056

Informatics and Mathematical Methods in Simulation
Vol. 4 (2014), No. 4, pp. 342-348

FACTORIZATION ALGORITHMS FOR CRYPTOGRAPHIC ANALYSIS OF ASYMMETRIC CRYPTO SYSTEMS

L.M. Timoshenko[1], **K.V. Verbik**[1], **Ya.M. Nikolaichuk**[2], **S.V. Ivasiev**[2]

[1] Odesa National Polytechnic University,
1 Shevchenko Str., Odesa, 65044, Ukraine; e-mail: lmt0902@gmail.com
[2] Ternopil national economical university,
1, Lvivska Str., Ternopil, 46020, Ukraine; e-mail: stepan.ivasiev@gmail.com

Improvement of existing factorization methods is one of the most urgent problems of modern asymmetric cryptography. RSA factorization problems (from RSA-968 до RSA-2048) have not been solved yet. Any factorization problem has high computational complexity and requires seeking ways to reduce this complexity for factorization methods, including a well known Fermat's method. This paper shows an approach – using Chrestenson basis and Jacobi symbols – for such a reduction. The basis allows reducing the computational complexity by reducing the length of the numbers subjected to operations.

Keywords: asymmetric cryptography, RSA, factorization, Fermat's factorization method, Chrestenson basis, computational complexity.

Introduction

Currently, asymmetric encryption is used for identification and authentication of users, prevention of data intrusion, and protection of electronic documents against counterfeiting and copying.

RSA system, the most known and popular public-key encryption system, is based on the computational intensiveness of factorization problem for long integers. The cryptographic strength of RSA system is based on the following fact: an encrypted message cannot be easily revealed without knowledge of factors P and Q, and finding these factors and N is considered a hardly solvable problem [1].

Today's computer networks are high speed networks and ensure high capacity for data processing and storage. Therefore, they require more reliable protection than stand-alone computers, since with the increase in network speed, the time for parallel implementation of cryptanalytic algorithms decreases. This gives a chance for unauthorized access to computer networks. Paralleling of cryptanalytic algorithms will allow performing the cryptanalysis of user messages not at the expense of increased capacity of an individual computer, but due to distribution of personal computers and connecting them to computer networks, including Internet.

Extended key length should compensate for any improvement to factorization algorithms. The question of an asymptotic upper bound on the running time of a factorization algorithm remains open. Currently, determining the prime factorization of a number is considered a computationally intensive problem. However, this statement is not proved; and this ambiguity encourages specialists to refine the existing algorithms and to develop the new ones. The potential exists to prove the computational intensiveness of factorization. This would guarantee the security of RSA system within a domain of keys.

Factoring a three hundred digit number is at the limit of current capabilities, whereas factoring a number having more digits is still a task for the future. The problems of factoring RSA-896 to RSA-2048 remain open.

A polynomial-time algorithm for prime factorization on a quantum computer has been already developed. The question of existence of such an algorithm for a common computer remains one of the key open issues in number theory. Therefore, improvement in the speed of existing factoring methods is one of the most urgent tasks in information security [2].

Purpose and tasks of the study

To increase the reliability of data-flow security in computer networks, it is required to increase the cryptographic strength of asymmetric cryptographic security systems. The *purpose* of this paper is to reduce the computational complexity of the factorization algorithm for assessment of cryptographic strength of RSA compatible systems by improvement of Fermat's factorization method. To implement this purpose, the following tasks should be solved:

1. analysis of modern methods of solving cryptanalytic problems,
2. substantiation of the use of the system of remainder classes,
3. improvement of Fermat's factorization method,
4. assessment of computational complexity of the algorithm proposed, and
5. comparative analysis of efficiencies of the improved and common Fermat's factorization methods.

Main Body

Factorization and discrete logarithm approaches are commonly used to solve cryptanalytic problems for public-key encryption systems. They not only present a threat currently, but have a potential for increase in value in the future. In [8], the following methods for factorization of natural numbers have been discussed: Fermat's factorization method, Pollard's rho algorithm, Shanks square forms factorization method, Lehman's method, Dixon's algorithm, continued fraction method, quadratic sieve, elliptic curve method, and number field sieve algorithms (special number field sieve and general number field sieve).

Depending on the complexity, factorization algorithms are generally divided into two groups. The first group comprises exponential algorithms, with their complexity exponentially depending on the lengths of input parameters (i.e., on the length of a binary represented number). To denote their complexity, it is common practice to use «O notation». This notation allows taking into account in function $f(n)$ only the most significant elements and neglecting the less significant elements. The second group comprises subexponential algorithms that run in the time greater than polynomial time but less than exponential time. While studying the operation of the algorithms, the following has been revealed: Fermat's factorization method is the most efficient when near factors are present; elliptic curve method is the most efficient when a relatively small factor is present; Shanks and Pollard's methods are the two most effective methods when the factors are small. Therefore, Fermat's factorization method is the best available for common RSA compatible crypto systems [9].

The emergence of new cryptographic phenomena and methods resulted in the emergence of new cryptographic applications of number theory. Along with elementary and analytic number theory, algebraic number theory and analytic arithmetic geometry are becoming more and more commonly used. Currently, quadratic sieve algorithm is commonly used. It is an exponential and computationally intensive algorithm.

Let us consider the use of Fermat's factorization method for large numbers. As is known, Fermat's factorization method is an odd number factorization algorithm that was proposed by Pierre Fermat as early as 1643. It is effective when N is the product of two integer numbers that are near to each other [3]. The factorization of a natural number into a

product of primes exists and is unique up to rearrangement of the factors, as it follows from a basic theorem of arithmetic.

Fermat's factorization method is based on finding the two integers, x and y, satisfying $x^2 - y^2 = n$, that leads to factorization $n = (x - y)(x + y)$. Encoding schemes are specified by the theoretical-and-numeral bases used for representation.

Commonly used are unitary, Haar, Gray, Rademacher, Chrestenson and Galois bases.

Chrestenson basis produces a system of remainder classes and is successfully used to develop special data compression processors and to implement high-capacity processors in cryptographic security systems [4].

It is well known that parallel processing is the most promising approach to speed up computation [7]. This feature is pertinent to the system of remainder classes (SRC) [2]. Although it has some shortcomings, the system ensures efficient performance of some operations (addition, subtraction, multiplication, and exponentiation for large numbers), which is very important, particularly when solving asymmetric cryptography problems.

Representation of decimal number N in SRC conforms to the least remainders b_i of this number in the system of mutually coprime modules p_i, that is, $b_i = N \bmod p_i$. Here the calculation range should be within $0 \le N \le P-1$, where $P = \prod_{i=1}^{n} p_i$. SRC-to-decimal conversion is performed based on the Chinese remainder theorem: $N = \left(\sum_{i=1}^{n} b_i B_i \right) \bmod P$, where $B_i = M_i m_i$, $M_i = \dfrac{P}{p_i}$, and basis numbers m_i are found from $(M_i m_i) \bmod p_i = 1$ [11].

The need for calculating basis numbers $m_i = M_i^{-1} \bmod p_i$ significantly increases the computational complexity of SRC-to-decimal conversion. The complexity of this problem is reduced for a full SRC form (SRC FF), when all $m_i = 1$ [4]. In [5], a modified SRC FF (SRC MFF) has been proposed, with basis numbers $m_i = \pm 1$, which excludes the need of seeking for the reciprocal number. To reduce the computational complexity of Fermat's factorization method, in [10] it has been proposed to use numerical Chrestenson basis that represents a number as the remainder when this number is divided by a fixed module.

We shall apply SRC to equation

$$x^2 = y^2 - n, \tag{1}$$

and perform the following conversion

$$x^2 \bmod p = y^2 - n \bmod p. \tag{2}$$

to get $x^2 \equiv (y^2 - n) \bmod p$. In order to solve this equation, it is advantageous to use Jacobi symbols that allow specifying explicitly whether a square root modulo is computed. This approach reduces the computational complexity of the algorithm.

Let $n \ge 3$ be an odd integer, and it is known that $n = p_1^{k_1} p_2^{k_2} \ldots p_t^{k_t}$, where p_i are prime numbers. Jacobi symbol $\left(\dfrac{a}{n} \right)$ is defined as follows:

$$\left(\frac{a}{n} \right) = \left(\frac{a}{p_1} \right)^{k_1} \left(\frac{a}{p_2} \right)^{k_2} \cdots \left(\frac{a}{p_t} \right)^{k_t}.$$

It follows from the properties of Jacobi symbols [1] that if n is an odd integer, and a is represented as $a = 2^k a_1$, where a_1 is an odd integer, then

$$\left(\frac{a}{n}\right) = \left(\frac{2^k}{n}\right)\left(\frac{a_1}{n}\right) = \left(\frac{2^k}{n}\right)\left(\frac{n \bmod a_1}{a_1}\right)(-1)^{\frac{(a_1-1)(n-1)}{4}}.$$

This formula allows computing the Jacobi symbol without knowing the prime factorization of n.

To find x, we need to perform the following conversion:

$$x \equiv \sqrt{(y^2 - n)} \bmod p. \qquad (3)$$

Then the improvement of Fermat method consists in the following: we shall deliberately disregard the values of $(y^2 - n)$ for which the root is not defined, and reduce the resolution at the expense of modulo operation.

The flowchart of the improved factorization algorithm is shown in Fig.1.

In order to assess the complexity of the algorithm developed, we need to determine the number of basic operations. It is worth to mention that the main computationally complex operations of the algorithm developed are seeking for a Jacobi symbol, computation of a modular square root, and computation of a root of the number. Table 1 presents the estimations of complexity of the basic steps of the multidigit number factorization algorithm developed.

Table 1.

Complexity of the basic operations of the algorithm developed

Basic steps of the multidigit number factorization algorithm developed	Complexity values of the basic operations
sqstart =Sqrt(P0)	n
Difference = (sqstart+1) * (sqstart+1)	$n\log_2 n$
Difference = Difference- P0	n
sqstartm[i]= (sqstartm +2) mod Prime[i]	$\log_2 n$
If Jacobi symbol (sqstartm[i], Prime[i])≠1 then Step 8	$\log_2{}^2 n$
If Jacobi symbol (sqstartm[i], Prime[i])=1 then i++, Step 10	$\log_2{}^2 n$
If Sqrt(Difference) is fractional, then Step 6	n
Output of sqrt(Difference+sqstart*sqstart)+sqrt(Difference)	n

Therefore, the total complexity of the algorithm developed is calculated as $O(n\log_2 n + 2\log_2{}^2 n + 4n) \approx O(n\log_2 n)$.

Since conventional Fermat's factorization method is similar to the trial division algorithm, the complexity of this method is estimated as $O(n(\log_2 n)^2)$.

The results of our investigations show that the algorithm developed has lower computational complexity than the conventional one, which is well seen in a graph showing the algorithm complexity versus the length of the input number (Fig. 2).

Since the operating efficiency of a factorization algorithm is estimated through its computational complexity, the improved Fermat's factorization method developed is more efficient than the conventional one.

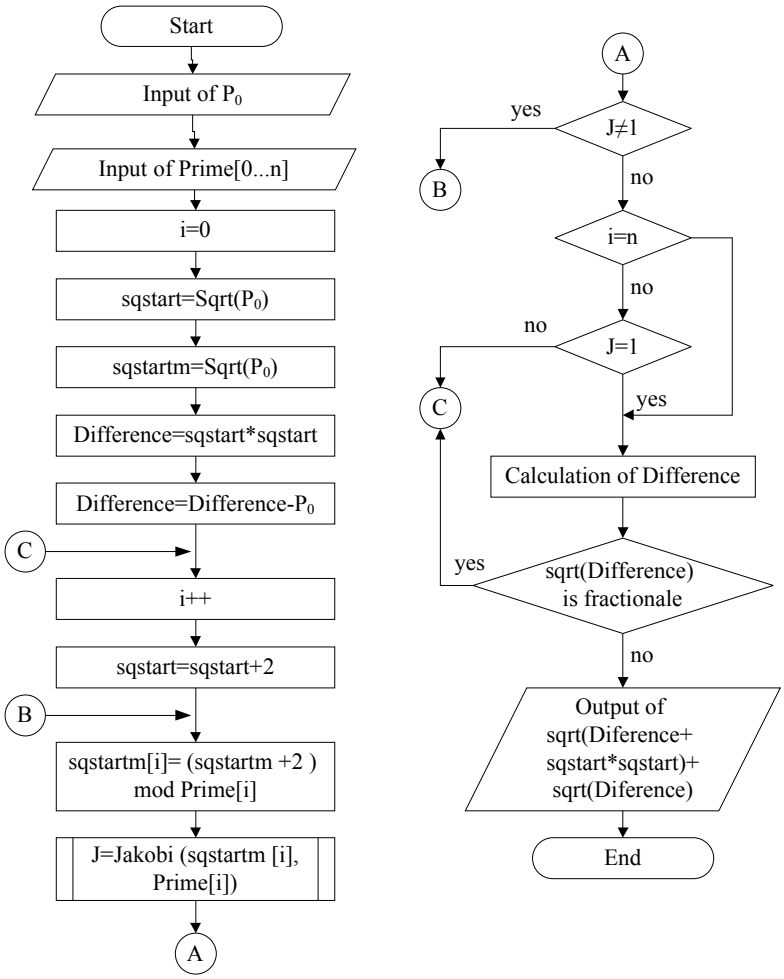

Figure 1. Flowchart of the improved Fermat's factorization algorithm

The operating efficiency gain of the algorithm developed compared to the conventional one is then defined as their computational complexity ratio

$$E(n) = \frac{n(\log_2 n)^2}{n(\log_2 n)} = \log_2 n.$$

Therefore, the efficiency gain of the algorithm developed increases with the increase in the bitlength of the number.

Based on the algorithm proposed and using C++ Builder 6.0 environment, we have developed a software product to perform factorization of multidigit numbers for specified module p.

For operations with extra long variables, a special A. Lenstra's library (involving Lip.h and Lip.c) was used. This library is optimized for use in crypto systems and incorporates a set of functions which we used in the application developed. The results of numerical experiments are in agreement with theoretical calculations and validate the scientific approach proposed.

Therefore, the efficiency of the Fermat's factorization method improved to reduce the computational complexity of factorization has been proved theoretically and experimentally.

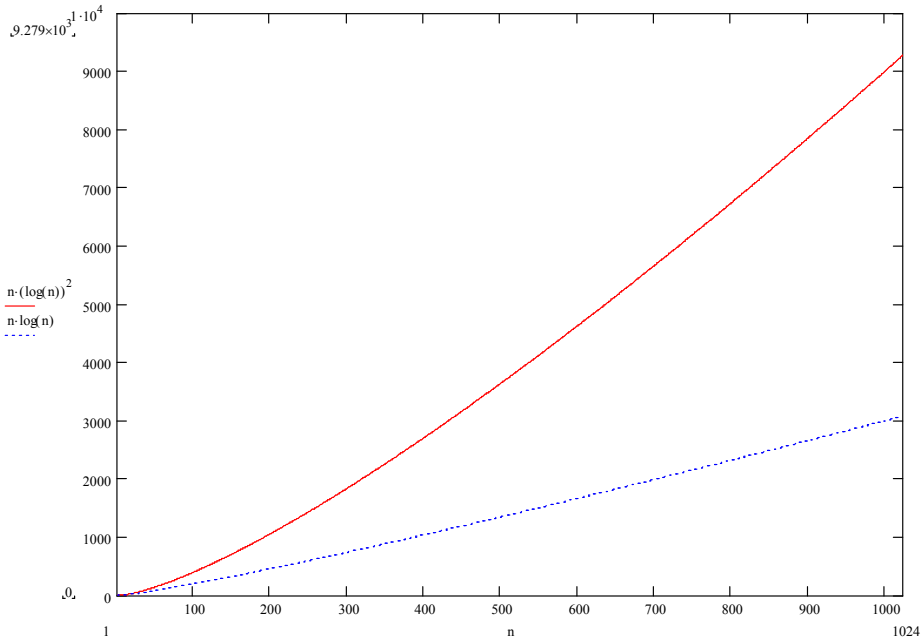

Figure 2. Computational complexity of the algorithm developed compared to that of the conventional one.

Conclusion

The basic factorization methods have been discussed and it has been revealed that Fermat's method is the most efficient when near factors are present. Additionally, the use of Chrestenson basis for the improvement of Fermat's factorization method for multidigit numbers has been substantiated.

Fermat's method has been improved using Chrestenson basis and Jacobi symbols, which allowed reducing its computational complexity.

Chrestenson basis allows reducing the computational complexity by reducing the length of the numbers subjected to operations,

Jacobi symbols allow not to use the numbers for which the square root modulo does not exist.

Thus, the operating efficiency gain of the algorithm developed compared to the conventional one equals $\log_2 n$.

Therefore, the improvement proposed in this paper for Fermat's method is a promising one for the assessment of the cryptographic strength of asymmetric cryptographic security systems.

References

1. Айерлэнд, К. Классическое введение в современную теорию чисел / К. Айерлэнд, М. Роузен. – М.: Мир, 1987. – 416 с.
2. Акушский, И.Я. Машинная арифметика в остаточных класах / И.Я. Акушский, Д.И. Юдицкий. – М: Сов.радио, 1968. – 440 с.
3. Вербіцький, О.В. Вступ до криптології / О.В. Вербіцький. – Львів: ВНТЛ, 1998. – 248 с.

4. Василенко, О.Н. Теоретико-числовые алгоритмы в криптографии / О.Н. Василенко. – М.:МЦНМО, 2003. – 328 с.
5. Касянчук, М.М. Теорія та математичні закономірності досконалої форми системи залишкових класів / М.М. Касянчук // Праці Міжнародного симпозіуму «Питання оптимізації обчислень (ПОО–XXXV)». – Київ–Кацивелі. – 2009.– Т.1. – С. 306-310.
6. Николаенко, О.В. Анализ методов факторизации в криптографических системах защиты информации / О.В. Николаенко, Л.Н. Тимошенко, К.В. Вербик // Матеріали ІІІ міжнародної науково-практичної конференції ІУСТ-Одеса-2014. – С. 173-175.
7. Николайчук, Я.М. Теорія джерел інформації / Я.М. Николайчук. – Тернопіль: ТзОВ «Терног раф», 2010. – 536 с.
8. Тимошенко, Л.М. Современные методы решения криптоаналитических задач / Л.Н. Тимошенко, К.В. Вербик // Матеріали ВШСМВС АСІТ’2014. – Тернопіль, 2014. – С. 220-222.
9. Тимошенко, Л.М. Шляхи удосконалення алгоритму факторизації чисел / Л.М. Тимошенко, К.В. Вербик // Тези доповідей X МНПК «Військова освіта і наука: сьогодення та майбутнє». – Київ, 2014. – С. 220-222.
10. Тимошенко, Л.М. Удосконалений метод Ферма факторизації чисел / Л.М. Тимошенко, К.В. Вербик, С.В. Івасьєв // Всеукраїнської НПК «Проблеми становлення інформаційної економіки в Україні». – Львів, 2014. – С.348-350.
11. Якименко, І.З. Алгоритм знаходження системи модулів модифікованої досконалої форми системи залишкових класів / І.З. Якименко, М.М. Касянчук, Л.М. Тимошенко, С.В. Івасьєв, Я.М. Николайчук // Матеріали МНПК СІЕТ. – Одеса, 2014. – С. 115-117.

АЛГОРИТМИ ФАКТОРИЗАЦІЇ ДЛЯ КРИПТОАНАЛІЗУ АСИМЕТРИЧНИХ КРИПТОСИСТЕМ

Л.М. Тимошенко[1], К.В. Вербик[1], Я.М. Николайчук[2], С.В. Івасьєв[2]

[1] Одеський національний політехнічний університет,
просп. Шевченка, 1, м. Одеса, 65044, Україна; e-mail: lmt0902@gmail.com
[2] Тернопільський національний економічний університет,
вул. Львівська, 1, м.Тернопіль, 46020, Україна; e-mail: stepan.ivasiev@gmail.com

Удосконалення існуючих методів факторизації є однією з найбільш актуальних задач сучасної криптографії. Задачі факторизації RSA від RSA-968 до RSA-2048 залишаються відкритими. На сьогодні задача факторизації має велику обчислювальну складність. Ставиться важливе завдання пошуку шляхів зменшення обчислювальної складності, у тому числі, і відомого методу Ферма. Дане дослідження пропонує один з напрямів його вирішення. Запропоновано скористатися теоретико-числовим базисом Крестенсона, який дозволяє зменшити обчислювальну складність за рахунок зменшення розрядностей чисел, над якими проводяться операції.

Ключові слова: факторизація, метод Ферма, базис Крестенсона, символ Якобі, обчислювальна складність, асиметрична криптографія, RSA.

АЛГОРИТМЫ ФАКТОРИЗАЦИИ ДЛЯ КРИПТОАНАЛИЗА АССИМЕТРИЧНЫХ КРИПТОСИСТЕМ

Л.Н. Тимошенко[1], К.В. Вербик[1], Я.М. Николайчук[2], С.В. Ивасьев[2]

[1] Одесский национальный политехнический университет,
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: lmt0902@gmail.com
[2] Тернопольский национальный экономический университет,
ул. Львовская, 1, г.Тернополь, 46020, Украина; e-mail: stepan.ivasiev@gmail.com

Усовершенствование существующих методов факторизации является одной из наиболее актуальных задач современной криптографии. Задачи факторизации RSA от RSA-968 до RSA-2048 остаются открытыми. На сегодняшний день задача факторизации имеет большую вычислительную сложность. Ставится важное задание поиска путей уменьшения вычислительной сложности, в том числе, и известного метода Ферма. Данное исследование предлагает одно из направлений его решения. Предложено воспользоваться теоретико-числовым базисом Крестенсона, который позволяет уменьшить вычислительную сложность за счет уменьшения разрядностей чисел, над которыми проводятся операции.

Ключевые слова: факторизация, метод Ферма, базис Крестенсона, символ Якоби, вычислительная сложность, ассиметричная криптография, RSA.