Date: 13th December-2025

# ALGORITHMS FOR SOLVING THE FACTORIZATION PROBLEM

**Abdurakhimov Bakhtiyor Fayziyevich**
Mirzo Ulugbek National University of Uzbekistan, e-mail: bakhtiyor@mail.ru
**Akhadova O'giloy Chorshanbi qizi**
Mirzo Ulugbek National University of Uzbekistan, e-mail: oaxadova95@bk.ru

**Abstract:** In this article, Fermat's factorization method and Pollard's Rho method are proposed to solve the factorization problem. Experiments showed that Pollard's Rho method is more efficient in solving the factorization problem compared to Fermat's factorization method, particularly in terms of time and memory complexity. The results of the experiments are planned to be discussed in detail in the next articles.

**Keywords:** Prime numbers, factorization, Fermat's factorization method, Pollard's Rho method, Birthday paradox, Floyd's cycle-finding algorithm.

**Introduction:**

The factorization problem is the problem of decomposing a number into prime factors. There are exponential-type algorithms that solve the factorization problem. Their complexity depends on the length of the input parameters exponentially, specifically the binary length of the number N being factored.

**Main Body:**

**Fermat's Factorization Method:**

Fermat's factorization method is an algorithm for factoring an odd integer n into its prime factors, proposed by Pierre Fermat in 1643.

The RSA encryption algorithm is based on the complexity of the factorization problem. An RSA public key contains an integer (usually called N) that is the product of two prime numbers (usually p and q).

If the private key values p and q in the RSA encryption algorithm are close to each other, then the encrypted information can be decrypted using Fermat's factorization method.

The method is based on searching for integers a and b that satisfy the relation $n = a^2 - b^2$, leading to the expansion $n=(a-b)(a+b)$.

**Algorithm Description:**

First, it is required to find the integers a and b that satisfy $n = a^2 - b^2$ . Accordingly, $n=(a-b)(a+b)$, where $(a-b)$ and $(a+b)$ are divisors of the required number n.

Using the fact that $n = a^2 - b^2$ is equivalent to $b^2 = a^2 - n$, a is checked from the value 1 to $(n-1)/2$ . If such values are found, the divisors $a-b$ and $a+b$ are considered as the private key values p and q.

**Example:**

For n=144, the following calculations are performed:

- q=1: $\rightarrow$ $p^2 = n + q^2$ $\rightarrow$ $p^2 = 144 + 1$ $\rightarrow$ $p^2 = 145$ $\rightarrow$ (not valid).
- q=2: $\rightarrow$ $p^2 = n + q^2$ $\rightarrow$ $p^2 = 144 + 4$ $\rightarrow$ $p^2 = 148$ $\rightarrow$ (not valid).



International Conferences
Open Access | Scientific Online | Conference Proceedings
INTERNATIONAL CONFERENCES OF PRACTICE

- q=3: → $p^2 = n + q^2$ → $p^2 = 144 + 9$ → $p^2 = 153$ → (not valid).
- q=4: → $p^2 = n + q^2$ → $p^2 = 144 + 16$ → $p^2 = 160$ → (not valid).
- q=5: → $p^2 = n + q^2$ → $p^2 = 144 + 25$ → $p^2 = 169$ → p = 13 → (valid).

Then n = n = $p^2 - q^2$ 144 = 169– 25 144 = (13 – 5) * (13 + 5) 144 = 8 * 18

For the method to work faster, n should have two prime divisors that are close to each other. Therefore, in the RSA algorithm, it is required that the difference between the prime divisors of the public key n be large.

Since factorization is a critical problem in cryptography, mathematicians have developed many algorithms that allow factoring a number in a limited time with high efficiency. These include Pollard's P-1 method, Pollard's P- algorithm, Lehmann's method, Elliptic curve method (Lenstra's algorithm), Dixon's algorithm, Quadratic Sieve algorithm, among others.

**Pollard's Rho Method:**

John Pollard invented the Rho factorization algorithm in 1975. Pollard's Rho algorithm is one of the most widely used algorithms for factoring integers. This is because the time complexity of the algorithm is proportional to the square root of the smallest prime divisor, and the algorithm requires much less computer memory than other algorithms.

**Concepts Used in Pollard's Rho Algorithm:**

**Birthday Paradox:**

The probability of two people having the same birthday is unexpectedly high, even for a small group of people.

**Floyd's Cycle-Finding Algorithm:**

This algorithm, also known as the "Tortoise and the Hare" algorithm, is based on a fable about a race between a tortoise (slow pointer) and a hare (faster pointer). If the tortoise and the hare start from the same point and move in a circle where the hare's speed is twice that of the tortoise, it is necessary to determine when they meet at the same point.

Let the prime divisors of N be p and q. The algorithm considers a pseudo-random sequence $\{x_i\} = \{x_0, f(x_0), f(f(x_0)), \ldots\}$, where f is a polynomial function, usually $f(x) = (x^2 + c) \bmod n$, with c=1.

In this case, the sequence $\{x_i \bmod p\}$ is more important for the result than the sequence $\{x_i\}$. Since f is a polynomial function and all values lie in the interval [0;p), this sequence eventually converges to a cycle. The birthday paradox indicates that the expected number of elements before repetition begins is $O(\sqrt{p})$. If p is less than $\sqrt{n}$, repetition starts in $O(\sqrt[4]{n})$.

Without knowing the value of p in advance, the question arises as to how the properties of the sequence $\{x_i \bmod p\}$ can be used. If a cycle exists with indices $s,t \leq j$, then the equality $x_s \equiv x_t \bmod p$ holds, where $x_s - x_t \equiv 0 \bmod p$ is equal to p. For convenience, one can check only the indices 2t−t.

International Conferences
Open Access | Scientific Online | Conference Proceedings
INTERNATIONAL CONFERENCES OF PRACTICE

International Conferences
Open Access | Scientific Online | Conference Proceedings
INTERNATIONAL CONFERENCES OF PRACTICE

Thus, if two indices s and ttt are found with g= *gcd* ($x_s$-$x_t$, n)>, then g is a prime factor of n. If g=n, then the algorithm needs to be repeated with other parameters (a different initial value for $x_0$ and the constant c in the polynomial function f).

**Floyd's Cycle-Finding Algorithm:**

This algorithm detects a cycle by moving two pointers through the sequence at different speeds. During each iteration, the first pointer moves one element forward, and the second pointer moves two elements forward. If there is a loop, the second pointer will eventually meet the first one during the cycles. If the loop length is λ and μ the index at which the loop starts, the algorithm runs in O(λ+μ) time.

Using this algorithm, it is possible to determine the parameters λ and μ. When a loop is detected, the algorithm returns true. If the sequence does not have a loop, the function runs indefinitely. However, this can be avoided by using Pollard's Rho algorithm.

**Example:**

For n=1189 with an initial value $x_0 = 2$ and the polynomial function $f(x) = x^2 + 1$:

$x_1 = 5$
$x_2 = 26$ — *gcd* (26 - 5, 1189) = 1
$x_3 = 677$
$x_4 = 565$ — *gcd* (565 - 26, 1189) = 1
$x_5 = 574$
$x_6 = 124$ — *gcd* (124 - 677, 1189) = 1
$x_7 = 1109$
$x_8 = 456$ — *gcd* (456 - 565, 1189) = 1
$x_9 = 1051$
$x_{10} = 21$ — *gcd* (21 - 574, 1189) = 1
$x_{11} = 442$
$x_{12} = 369$ — *gcd* (369 - 124, 1189) = 1
$x_{13} = 616$
$x_{14} = 166$ — *gcd* (166 - 1109, 1189) = 41

So, one of the prime multipliers of 1189 is 41.

**Analysis of Results:**

**Factorization Algorithms with Exponential Complexity**

| Algorithm name | Basis | Efficiency Limit | Calculation complexity |
|---|---|---|---|
| Fermat's Factorization Method | Express the number N as the difference of the squares of two numbers. | < 16 bit | $O\left(N^{\frac{1}{3}}\right)$ |
| Pollard's ρ-Algorithm | Find the length of the cycle in a repeating function sequence and use the birthday paradox to derive certain results. | < 30 bit | $O\left(N^{\frac{1}{4}}\right)$ |

**Conclusion:**

In conclusion, the advantage of Pollard's Rho method is that it eliminates the need to test

all possible integers until one of the factors is found, thereby improving time complexity. This algorithm is more efficient than Fermat's factorization method in terms of time and memory complexity. Additionally, problems based on discrete logarithms on elliptic curves can be solved using Pollard's Rho algorithm. The results of the conducted experiments are planned to be discussed in detail in the following articles.

**REFERENCES:**

1. Ishmukhametov Sh. T. Methods of Factorization of Natural Numbers: A Textbook — Kazan: Kazan University, 2011. — 190 p.
2. Schneier B. Applied Cryptography. Protocols, Algorithms, and Source Code in C. — Moscow: Triumph, 2002. — 816 p. — ISBN 5-89392-055-4.
3. Bressoud, D. M. Factorization and Primality Testing. — N. Y.: Springer-Verlag, 1989. — 260 p. — ISBN 0-387-97040-1.

International Conferences
Open Access | Scientific Online | Conference Proceedings
INTERNATIONAL CONFERENCES OF PRACTICE