Section 1

COMPARATIVE ANALYSIS OF FACTORIZATION ALGORITHMS

Oles Yudin
National Aerospace University «Kharkiv Aviation Institute»
Scientific adviser: Vladimir Pevnev

The relevance of this research is based on the fact that many modern cryptographic systems, especially those that protect the privacy of transmitted information, rely on complex mathematical problems that are hard to solve using standard methods [1]. One of these problems is breaking down large numbers into prime factors, which is the foundation of the RSA encryption algorithm [2]. Today, cloud systems can perform calculations faster using clusters of computers, which helps speed up the factorization process and brings us closer to solving such problems [3]. As a consequence, it is important to study how the sizes of two large prime numbers affect the strength of the keys used in the RSA system. Factorization has an interesting feature: when a number is the product of two prime numbers P and Q the difficulty of breaking it down depends on the size difference between P and Q [4]. For example, if the prime numbers have a big size difference, the efficiency of factorization algorithms can change. This makes it important to study not only how effective the algorithms are in general but also how changes in these parameters affect the time it takes to factorize the numbers.

The purpose of this work is to analyze current factorization algorithms and study how the ratio between large prime numbers influences the difficulty of factorizing their product.

Principal provisions. This work studies the efficiency of various algorithms for factoring large numbers, including Fermat's factorization, Pollard's rho algorithm, Lenstra's algorithm, Dixon's algorithm, and the Continued Fraction Method (CFRAC). A comparison of these algorithms shows that their efficiency largely depends on the specific properties of the number being factored. Fermat's algorithm is highly efficient for numbers with two prime factors that are close in size, as it works by finding factors with a small difference in magnitude [5]. Pollard's rho algorithm, in turn, is better suited for numbers with small factors, as its methods can quickly detect such factors. Lenstra's algorithm is particularly efficient when factoring numbers with large prime factors, especially when one of the factors is a large prime number. This makes it effective for certain types of numbers often used in cryptography. Dixon's algorithm is a generalized version of Fermat's method that uses a factor base and smooth numbers. It is effective for factoring numbers with large but less significant factors, thanks to

its ability to handle numbers with many medium-sized prime factors. The Continued Fraction Method has proven to be the most efficient for numbers with certain structures and small prime factors, allowing it to solve factorizations where other algorithms may be less effective. The results of the study show that choosing a factorization algorithm depends on the specific characteristics of the input numbers, enabling the optimization of the factorization process under different input parameters.

Conclusions. The choice of a factorization algorithm depends on the characteristics of the number, and each algorithm has its advantages in specific situations, making them useful in different contexts of number factorization. Factoring large numbers is the foundation of security for cryptosystems like RSA, where the task of breaking down a number N into prime factors P and Q is critically important. Finding these factors allows the private key to be revealed and the system to be compromised. It is important to note that if there is a specific relationship between P and Q, the factorization problem may become easier, as specialized algorithms are effective for such patterns.

List of references

1. Pollard JM. Theorems on factorization and primality testing. *Mathematical Proceedings of the Cambridge Philosophical Society*. 1974. Volume 76(3). Page 521 – 528. DOI: https://doi.org/10.1017/S0305004100049252.

2. Milanov E. The RSA Algorithm. URL: https://sites.math.washington. edu/~morrow/336_09/ papers/Yevgeny.pdf (date of access: 05.10.2024).

3. Kleinjung T. A Heterogeneous Computing Environment to Solve the 768-bit RSA Challenge. 2012. Volume 15. Page 53 – 68. DOI: https://doi.org/10.1007/s10586-010-0149-0.

4. Pevnev V., Yudin O. Method of testing large numbers for primality. *Advanced Information Systems*. 2024. Volume 8(2). Page 99–106. DOI: https://doi.org/10.20998/2522-9052.2024.2.11.

5. Aminudin A., Cahyono E. A Practical Analysis of the Fermat Factorization and Pollard Rho Method for Factoring Integers. 2021. Volume 12(1). Page 33 – 40. DOI: https://doi.org/10.24843/LKJITI.2021.v12.i01.p04.

Information about the authors

Oles Yudin, a PhD student from the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», o.yudin@csn.khai.edu

Vladimir Pevnev, Dr. Sc., professor from the Department of Computer Systems, Networks and Cybersecurity, National Aerospace University «Kharkiv Aviation Institute», v.pevnev@csn.khai.edu