

Computational Efficiency Enhancements in Number-Theoretic Cryptographic Algorithms

¹Pavithra R, ²Akshatha B

¹ Assistant Professor, ² Assistant Professor

Department of Mathematics, School of Applied Science

¹KLE Society's S Nijalingappa College, ²Sapthagiri NPS University

pavithrar20320@gmail.com, akshatha.v95974@gmail.com

Abstract- With the increasing development of digital communication, there is a necessity for reliable and fast security algorithms. Many commonly used security algorithms are built on complex mathematics, which ensures high reliability but results in reduced efficiency, especially when such algorithms are applied on smaller devices, for example, phones or smart sensors.

This research paper considers possible ways to increase the speed and efficiency of calculations related to cryptography. Fast modular exponentiation and Chinese remainder theorem are discussed as examples of techniques that would allow improving performance of these calculations. The study also compares various cryptographic techniques based on memory efficiency and their ability to process large volumes of data.

The results indicate that, although traditional encryption algorithms are reliable, they can be significantly improved by using mathematical optimization techniques.

The performance of modular exponentiation, Euclidean algorithm, extended Euclidean algorithm, and hash-based validation techniques was comparatively analyzed using execution time, memory usage, and computational accuracy.

The findings demonstrate that computational optimization techniques can significantly enhance the efficiency of number-theoretic cryptographic algorithms while maintaining a high level of security.

Keywords— Number Theory, Cryptography, RSA, Modular Exponentiation, Chinese Remainder Theorem, Computational Efficiency.

I. INTRODUCTION

In the contemporary digital world, cryptographic schemes are vital for securing communications, data protection, and user identification processes. The majority of cryptographic schemes rely on concepts from number theory, such as modular arithmetic, prime numbers, and congruence relations. [2]. Cryptographic schemes RSA and Diffie–Hellman operate based on integer factorization and discrete logarithm problems which make the algorithm computations difficult to execute.

However, while these algorithms are effective, they usually require heavy computations with the use of large numbers, which brings a lot of complications regarding time and resources usage in cloud computing, digital identities, and embedded devices. Thus, enhancing efficiency has become an important issue in cryptography research.

Research suggests that the connection between computational complexity and algorithm efficiency is vital for the feasibility of the cryptographic scheme [3]. Also, there are different types of cryptographic schemes which have different strengths in terms of security but computational costs vary as well [3].

Research Objectives

The objectives of this study are:

- To analyze computational bottlenecks in number-theoretic cryptographic algorithms.
- To compare the performance of different optimization techniques used in cryptographic computations.
- To evaluate computational efficiency based on execution time, memory usage, and computational accuracy.
- To identify methods that improve computational performance while maintaining cryptographic security.

II. LITERATURE REVIEW

Research on number-theoretic algorithms shows that they are essential for both cryptographic security and data validation systems. A comparative study of commonly used algorithms demonstrates that methods such as the Euclidean algorithm, modular exponentiation, and hash-based validation techniques exhibit different performance characteristics depending on the computational context [1].

Experimental findings indicate that the Euclidean algorithm is highly efficient due to its low computational overhead and minimal memory usage. In contrast, modular exponentiation is more computationally demanding but is crucial for cryptographic operations such as encryption and key generation [1]. The extended Euclidean algorithm provides reliable computation of modular inverses, which are essential in digital signature systems [1].

From a theoretical perspective, modular arithmetic and congruence theory form the foundation of cryptographic algorithms. Key mathematical principles such as Euler's theorem and Fermat's theorem enable efficient computation of large exponentiations, which are central to RSA and similar systems [2]. Additionally, the Chinese Remainder Theorem improves efficiency by decomposing large computations into smaller operations, thereby reducing execution time [2].

Modern research also emphasizes that optimizing computational complexity is critical for scalable cryptographic systems, especially in large-scale networks and secure communication infrastructures [3].

Research Gap

Although many studies have investigated computational optimization techniques for cryptographic algorithms, limited research has comparatively analyzed multiple number-theoretic algorithms using common performance metrics such as execution time, memory usage, and computational accuracy. The present study addresses this gap by providing a comparative assessment of selected number-theoretic algorithms to identify efficient optimization techniques for secure cryptographic applications.

III. METHODOLOGY

A. Dataset

A sample dataset consisting of cryptographic parameters, including prime numbers, modulus values, public keys, private keys, and varying message sizes, was used to evaluate the computational performance of the selected number-theoretic algorithms. The dataset was designed to provide a consistent basis for evaluating execution time, memory usage, and computational accuracy under identical computational conditions. The sample cryptographic dataset used for the experimental evaluation is presented in Table 1.

Test Case	Prime Number (p)	Prime Number (q)	Modulus (n=p×q)	Public Key (e)	Private Key (d)	Message Size
1	61	53	3233	17	2753	128
2	71	67	4757	17	3533	256
3	97	89	8633	17	497	512
4	101	113	11413	17	3953	1024
5	131	137	17947	17	10313	2048

Table 1: Sample Cryptographic Dataset

B. Algorithms Used

- Modular Exponentiation
- Euclidean Algorithm (GCD)
- Extended Euclidean Algorithm
- Hash-Based Number-Theoretic Validation

C. Performance Metrics

The algorithms were evaluated using:

- Execution Time
- Memory Usage
- Computational Accuracy
- Computational Complexity

D. Experimental Procedure

Each algorithm was executed using the same dataset under identical computational conditions. The performance was measured based on execution time, memory usage, computational accuracy, and computational complexity. The results were comparatively analyzed to identify the most efficient optimization techniques.

IV. RESULTS AND DISCUSSION

It is possible to state the obvious differences in computational efficiency for the examined algorithms. The Euclidean algorithm appears to be the fastest in its execution due to the simplicity of the algorithmic process and small amount of computations needed for the algorithm [1].

Modular exponentiation plays an important role in the cryptographic operation; however, it demonstrates higher complexity level because of its dependence on big exponent values. Nevertheless, there is an optimization technique that lowers the level of the computational cost of modular exponentiation, namely the squaring and multiplying technique [2].

The extended Euclidean algorithm provides an efficient method for computing modular inverses required in encryption and decryption processes [1].

Hybrid hash-based algorithms perform quite well in validation tasks of high scales. These algorithms show high efficiency and good scalability but require extra memory for performing [1].

Thus, from the above, it follows that the efficiency of cryptographic algorithms depends not only on their mathematical properties but also on their complexity and computational efficiency [3].

Table 2 presents the comparative analysis of the selected number-theoretic algorithms based on their applications, strengths, and limitations, providing a clear overview of their suitability for different cryptographic applications.

Algorithm	Best For	Strength	Limitation
Euclidean (GCD)	Speed & validation	Fastest, low memory	Not for encryption
Modular Exponentiation	Encryption	Strong security	Slower
Extended Euclidean	Modular inverse	Essential for RSA	Moderate complexity
Hash-Based Methods	Large-scale systems	Scalable	Memory + collisions

Table 2: Comparative Analysis of Number-Theoretic Algorithms

V. CONCLUSION

In this research, emphasis is put on how efficiency plays a significant role in number theory cryptographic algorithms. The paper uses both theoretical and practical findings to indicate how improvements in efficiency can be realized without putting the security at risk.

From the results of the paper, it is evident that simpler algorithms give better performance, but complex algorithms are better in terms of cryptography. Optimization approaches, such as efficient exponentiation and modular decomposition, have an important role in performance improvement [2].

Future research should focus on developing cryptographic algorithms that achieve an optimal balance between efficiency, scalability, and security [3].

Overall, this study demonstrates that computational optimization techniques can significantly enhance the efficiency of number-theoretic cryptographic algorithms without compromising their security. The findings contribute to the understanding of efficient cryptographic implementations and provide valuable insights for future research in secure communication systems.

VI. REFERENCES

- [1] D. Dhuware and J. Kushwah, "A Comparative Evaluation of Number-Theoretic Algorithms for Assessing Their Relative Efficacy in Digital Identification," IJETIR, 2025.
- [2] B. Atiq, N. Fakhri, and Z. Wahdat, "Number Theoretic Foundations of Cryptography: From Congruence Theory to RSA," Journal of Advanced Computer Knowledge and Algorithms, 2026.
- [3] "Cryptographic Algorithms and Computational Complexity: A Mathematical Approach to Securing IT Networks," ResearchGate, 2025. Available: https://www.researchgate.net/publication/393799942_Cryptographic_Algorithms_and_Computational_Complexity_A_Mathematical_Approach_to_Securing_IT_Networks