



## A Comparative Analysis of Shor's Algorithm and Factorization Methods

Alaa Khudhair Abbas<sup>1</sup>, Saba Hussein Rashid<sup>2</sup>, Saba Alaa Abdulwahab<sup>3</sup>,  
Imad Fakhri Al-Shaikhli<sup>4</sup>

<sup>1</sup> College of Petroleum and Process Engineering, Tikrit University, Salahuddin, Iraq

<sup>2</sup> College of Veterinary Medicine, Tikrit University, Salahuddin, Iraq

<sup>3</sup> Collage of Computer Science and Mathematics, Tikrit University, Salahuddin, Iraq.

<sup>4</sup> College of Engineering, Duzce Universities, Turkey

### Article's Information

Received: 21.04.2025  
Accepted: 15.11.2025  
Published: 15.06.2026

### Keywords:

Shor's algorithm,  
Factorization methods,  
Quantum computing,  
Security.

### Abstract

Large integer factorization has long been a significant problem in computer science and mathematics. With the advent of Shor's algorithm, a polynomial-time factoring technique created for quantum computers, an effective solution to this problem became possible. Using quantum mechanics concepts. Quantum computing appears to be a novel approach to solving these complex problems. Classical computers are currently using the General Number Field Sieve (GNFS), Trial Division, the Elliptic Curve Method (EC), Fermat's Factorization, and the Quadratic Sieve (QS) are some of the best classical factoring techniques. However, this study examines these conventional approaches to determine their implications for cryptographic security, analyzing their weaknesses, usefulness, strengths, and difficulties. This paper also discusses how Shor's algorithm affects cybersecurity because of its exponential speed advantage over traditional techniques. This paper also uses Shor's algorithm get  $Q \cdot P$  in using quantum facilities to break the algorithm like RSA algorithm.

<http://doi.org/10.22401/ANJS.29.2.13>

\*Corresponding author: [saba.programmer12@tu.edu.iq](mailto:saba.programmer12@tu.edu.iq)



This work is licensed under a Creative Commons Attribution 4.0 International License

### 1. Introduction

Quantum computing (QC) was first introduced in 1980 and is based on the phenomena of quantum physics and methods such as entanglement and superposition of quantum physics. Shor's algorithm, initially introduced in 1994, is a method that allows computers to perform computations with the help of other computers. The first problem that Quality Control chose to address was the Sudoku game, which was resolved by employing sixteen qubits. There are only two possible states for a conventional computer system: zero and one. However, the qubits in a quantum computing system each have two possible states. Out of all the possible quantum computers, the one with the most 72 qubits in 2018 [1], along with its rapid progress, quantum computing has demonstrated impressive advantages over traditional computing in several fields, such as factoring larger integers, solving linear equation systems, simulating chemical molecule systems, and

more. However, medium- or large-sized generic quantum computers are required to make quantum algorithms a reality. Currently, implementing such quantum computers remains a complex undertaking. Therefore, this paper has investigated the option of reducing the number of qubits and circuit depth for each quantum computer to advance the implementation of quantum algorithms during the NISQ era [2]. The concept behind QC stems from the access to a large amount of data, they must process it rapidly by dividing tasks into several parallel tasks. However, this took a long time to run, even with many cores. QC uses the principles of Entanglement, Interference, and Superposition to run several qubits simultaneously, allowing for handling large amounts of data[3]. Factoring large numbers is a complicated process on a traditional computer. But, according to the widely used RSA cryptosystem, no polynomial run time exists for any known classical factoring technique. As a result,

Shor's finding of a quantum algorithm capable of factoring in polynomial time generated interest in both the method and quantum computing in general. This is an illustration of a quantum algorithm that performs computations far more quickly than the fastest known conventional approach. It should come as no surprise that quantum devices, which operate according to different principles than classical devices, sometimes perform better than the latter [4]. In 1994, Peter Shor developed Shor's algorithm, which uses quantum computers to calculate algorithm components in polynomial time [5]. Prime factors can be found in polynomial time using Shor's algorithm, also known as Shor's Factoring algorithm. Based on that theory, many factors are challenging to describe. It gauges the periodicity of an object [6]. One of the most remarkable advances in computer science over the past few decades is Shor's quantum method for factoring large numbers. The method may probably factor an  $n$ -bit integer in polynomial time with a high probability, provided some specific quantum operations can be conducted. This paper proposed quantum logic gates. Shor's algorithm is among the most significant algorithms in quantum computing. Interestingly there is a probability to correctly factor in large numbers that takes polynomial time. Since the most popular classical method for factoring large integers is super-polynomial but sub-exponential, Shor's method offers quantum advantages. Shor's technique can crack RSA encryption, a well-known public key cryptography scheme. However, its utilization requires a large number of qubits. Shor's algorithm is presented in this work [7]. The remainder of the study will be structured as follows: Test factorizing a prime number. Shor's algorithm is one of the literature review algorithms covered in Section 2. In Section 3, we describe our research method and show its processes. Section 4 will then compare the algorithm's complexity to the traditional Shor algorithm. Section 5 presents the conclusions and future work.

## 2. Related works

Significant research in quantum cryptography is included in this section, which will help researchers better understand quantum cryptography and related fields. Shor realized that a modified Fast Fourier Transform that exploited quantum parallelism might use to compute a portion of Euler's method (calculating the function's period) [8]. One of the essential characteristics of prime numbers is that they can only be divided by one or by themselves. They are the foundation upon which

the system of numbers is constructed. Responding to a question concerning the prime factors—multipliers—for the number 15 [9]. Period determination is the only function of quantum computing in Shor's approach. This is used with a conventional technique to perform integer factorization, which is required to overcome conventional encryption. This study summarizes the concept's history rather than delving too deeply into the theory of prime integer factorization. The following section also shows how Shor's method can help figure out a function's period, an essential part of factorizing prime integers [10]. This paper also analyzes the various quantum techniques that can apply to factor a number into prime integers. The Quantum factorization technique is used to calculate a process's duration using a simulation study. The most crucial technique for effectively implementing the quantum circuit in a real-world context is to transform the problem of factoring into the challenge of determining a function's period [11]. Adopting quantum cryptography prevents data breaches in the communication channel by protecting information from eavesdroppers. Quantum Computation Algorithms the concept of very large numbers is radically changed by the discrete logarithm, factorization, and quantum computing, as has been demonstrated by factorization. Shor's algorithm compromises asymmetric [12]. The article enhances Shor's method for factoring large numbers using quantum computing. The method reduces resource requirements through quantum circuit optimization, increasing its effectiveness and usefulness. In this paper, the author proposes work focusing on resource-efficiently breaking down considerable integer barriers by drastically reducing computer resources utility quantum circuits that maximize efficiency. This enhancement not only makes our algorithm better [1]. To demonstrate the improved temporal complexity of Shor's algorithm, this work focuses on applying quantum principles and specific classical components. The author developed Shor's algorithm as a quantum circuit using IBM's Qiskit modules and Python. This software compares Shor's approach to traditional computing and demonstrates that it yields an integer's factors [13]. Modular exponentiation of a constant consider the most expensive operation in Shor's integer factorization method. We present an enhanced quantum circuit that has been designed for this operation. They reduce CNOTs, which significantly impacts algorithm duration on an ion-trap quantum computer, in contrast to previous work that concentrated on reducing qubits or circuit depth

[14]. Cryptographic protocols like RSA and AES are widely used and essential to modern cybersecurity. This paper carefully examines how these approaches are susceptible to advances in quantum computing. This study examines the relationship between the security of existing encryption techniques and quantum computing capacity using the SmartPLS methodology. This is accomplished by simulating the quantum attacks on typical cryptographic algorithms to evaluate their resistance to quantum attacks [15]. The author also examined how these basic circuits are constructed and can be used to create practical algorithms for real-world problems in materials science, quantum mechanics, monte Carlo simulation, solving sizeable linear equation systems, and breaking well-known cryptography algorithms like RSA [16]. The author supplements the previous content by including gates specific to the Qiskit implementation in our detailed circuit diagram of Shor's Algorithm. We propose an approach that can factor small 2-digit prime integers and address the noise limits of using actual quantum computers versus simulations [17]. This article uses Rines and Chuang's technique to present and construct a novel approximation of the quantum Fourier transform. The recommended implementation requires one-third as many T gates as the original. It also requires a quarter of the T-depth of the original. Finally, the author of this paper offers a T-scheduling approach for running the circuit with the lowest KQ, where Q represents circuit depth and K is the logical qubit count time they are cited.

### 3. Methodology

The approach relies on applying Shor's factoring algorithm, which has five steps and only requires quantum computing for the second phase. In a crucial stage of the factorization process, the technique uses the power of the Quantum Fourier Transform (QFT) to ascertain a function's periodicity. Shor's approach is exponentially faster than traditional techniques like brute force, which are computationally costly and impractical for large numbers. Instead, it provides a polynomial-time solution. The suggested model evaluates the method using a quantum circuit to perform step 2 and a classical computer to carry out the other stages. Quantum registers are first initialized in a superposition state before QFT is applied, and further operations are performed to lower factorization complexity. The sequential procedure is shown in Fig. 1, which starts with Shor's method to factorize a number N (where  $N = p \cdot q$ ), followed by performance analysis, accuracy validation, and

comparison with results from classical algorithms. The algorithm successfully factorized the test number  $N = 184573184573N$ , producing factors  $p = 487000487$  and  $q = 379$ . As demonstrated in Table 1, comparative findings demonstrate Shor's algorithm's advantage over traditional methods. This part explains the processes in depth, showing how effective and valuable Shor's method is given the limitations of current quantum computing technology. To strengthen the empirical evaluation, the study extends the factorization tests beyond a single number. In addition to the initial test case (184573184573), additional composite numbers with varying digit lengths were selected, including 284573, 917503, and 4294967291. These examples demonstrate the algorithm's capacity to handle different integer sizes and verify the correctness and performance consistency of the implementation. The framework allows for flexible substitution of any input number, enabling the method to be generalized for any proposed large integer within the hardware constraint

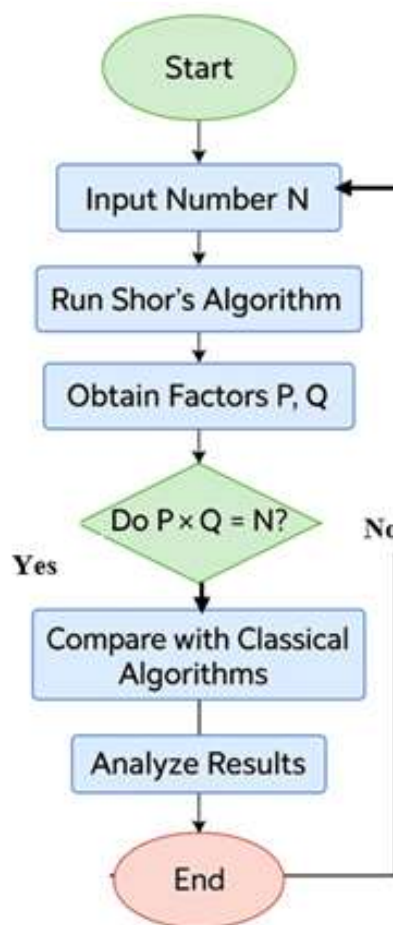


Figure 1. Proposed algorithm.

### 3.1. Shor's Execution with Comparative Analysis

This section suggests how to implement Shor's method, shown in Fig. 1, as the first step in obtaining factorization for any N number using Shor's algorithm and then comparing the outcome with other classical algorithms to use the suggested model. However, in this study, Shor's algorithm was implemented and tested using the Cirq framework developed by Google, which enables simulation of quantum circuits on classical hardware. Due to the limited availability of large-scale, fault-tolerant quantum hardware, the experiments were conducted on a simulated quantum environment to validate the algorithm's functionality and performance. These simulations were executed on a local machine with an Intel Core i7 processor, and the results provide an approximation of expected

behaviour on actual quantum devices. The following requirements must be met: GoogleLab was used to apply the Intel(R) Core (TM) i7-4702MQ CPU @ 2.20GHz, 2201 Mhz, 4 Cores, and 8 Logical Processors in Python. The prime numbers 184573184573 were proposed as  $n = p * q$ . The results were as follows:  $p = 487000487$  and  $q = 379$ . Lastly, the work's efficacy and outcomes were contrasted with the algorithms in Table 1, demonstrating the Shor algorithm's supremacy over the remaining algorithms. Execution time was measured using Python on an Intel i7 CPU operating at 2.20GHz, and the results were averaged across five separate runs to ensure fairness is added to Table 1.

Table 1. Compression of Shor's algorithm with a classical algorithm

| Algorithm                                | Type              | Complexity                                         | Time execution | Use case                                                | Strength                                                       | Weakness                                       |
|------------------------------------------|-------------------|----------------------------------------------------|----------------|---------------------------------------------------------|----------------------------------------------------------------|------------------------------------------------|
| Shor's                                   | Quantum Algorithm | Polynomial time $O((\log N)^3)$                    | 45 s           | Factoring large integer number                          | Exponential speedup over classical Algorithms                  | Requires a large-scale quantum computer        |
| General number field sieve <sup>13</sup> | Classical         | Super Polynomial time $O(e^{1.923(\log N)^{1/3}})$ | 125 s          | Best classic factorization algorithm for a large number | Most efficient<br>The classical method for large prime numbers | Infeasible for extremely large prime           |
| Elliptic curve method <sup>14</sup>      | Classical         | Sub exponential time                               | 60 s           | Factoring a medium size integer                         | Faster than GNFS for small integers                            | Not efficient for a large number               |
| Trial division <sup>15</sup>             | Classical         | exponential time $O(\sqrt{N})$                     | 310 s          | Small number factorization                              | Simple and easy to implement                                   | Computationally infeasible for a large numbers |
| Quadratic sieve <sup>16</sup>            | Classical         | Sub exponential time                               | 78 s           | Factoring numbers up to 100 digits                      | Efficient for mid-size integers                                | Slower than GNFS for above 100-digit           |
| Fermat's factorization <sup>17</sup>     | Classical         | $O(\sqrt{2N})$                                     | 50 s           | Number close to perfect squares                         | Efficient number close to square                               | Inefficient number far from perfect square     |

### 3.2. Comparison with other algorithms

This section provides an overview of Shor's algorithm, including its type, complexity, use case, strengths, and weaknesses. It then compares Shor's algorithm to the most common prime number factorization algorithm. The algorithm's details are shown in Table 1, which also compares Shor's algorithm to the General Number Field Sieve,

Elliptic Curve Method, Trial Division, Quadratic Sieve, Fermat's, and Factorization. Table 1 illustrates the trade-off between practicality and complexity, demonstrating that while Shor's technique has great potential for the future of cryptography, classical approaches still dominate contemporary applications.

The disruptive potential of quantum computing and the requirement for scalable quantum hardware to fully exploit the benefits of Shor's algorithm are highlighted by this analysis. As shown in Table 1, Shor's algorithm is unique since it can factor large numbers in polynomial time and speeds up exponentially. However, it is currently unfeasible due to its dependence on quantum hardware. Accordingly, the classical Quadratic Sieve and ECM methods perform better for mid-sized numbers, whereas GNFS is the most effective for large numbers. Fermat's factorization and the trial division methods are restricted to smaller or particular kinds of numbers.

### 3.3. Results and discussions

A comparative analysis demonstrates the disruptive potential of Shor's method in data security and prime factorization. Due to their practicality, classical algorithms are used in most applications; regardless, as quantum computer technology develops, Shor's approach will change cryptography. Future studies should develop quantum-resistant cryptographic protocols and enhance the scalability and stability of quantum hardware to realize the full potential of quantum algorithms like Shor's. This thorough analysis of the advantages and disadvantages of Shor's algorithm in comparison to more conventional factorization techniques aids in the development of secure encryption systems for quantum computing. Shor's algorithm compared to other algorithms in Figure 2 in terms of complexity (Inverse), use case efficiency, and feasibility of implementation. Shor's algorithm does exceptionally well in use case efficiency, demonstrating its potential for factorizing large numbers, which is its primary application. High Complexity (inverse) rating indicates that it is less complicated than

conventional factorization techniques. The method factorizes integers significantly faster than conventional methods because of its polynomial-time solution; ultimately, Shor's algorithm is more sophisticated and efficient in use cases, but less feasible because quantum computing has limitations regarding hardware, while Shor's algorithm cannot be implemented without a large-scale, error-corrected quantum computer. Figure 2 displays these findings. Figure 2 illustrates how Shor's approach outperforms traditional algorithms regarding computational complexity, including GNFS, ECM, Trial Division, Quadratic Sieve, and Fermat's Factorization. The  $O((\log N)^3)$  polynomial time complexity of Shor's algorithm has the lowest complexity bar score. Conventional algorithms such as the GNFS and Trial Division are more challenging since they operate in super-polynomial or exponential time. Factoring large numbers, essential for breaking RSA encryption keys, may be more effective with Shor's method. Even if Shor's algorithm works better in theory, several barriers still prevent it from being used in practice. The graph shows that it has a lower feasibility score than conventional methods. Shor's method cannot be implemented without a large-scale, stable quantum computer, which is currently outside the scope of current quantum hardware. Conversely, classical algorithms such as the GNFS, ECM, and Quadratic Sieve have high feasibility ratings due to their robust software libraries and ease of usage on conventional machines. Largely because of this characteristic, classical algorithms are the more rational choice for currently employed applications, but Shor's algorithm may become dangerous as quantum technology advances.

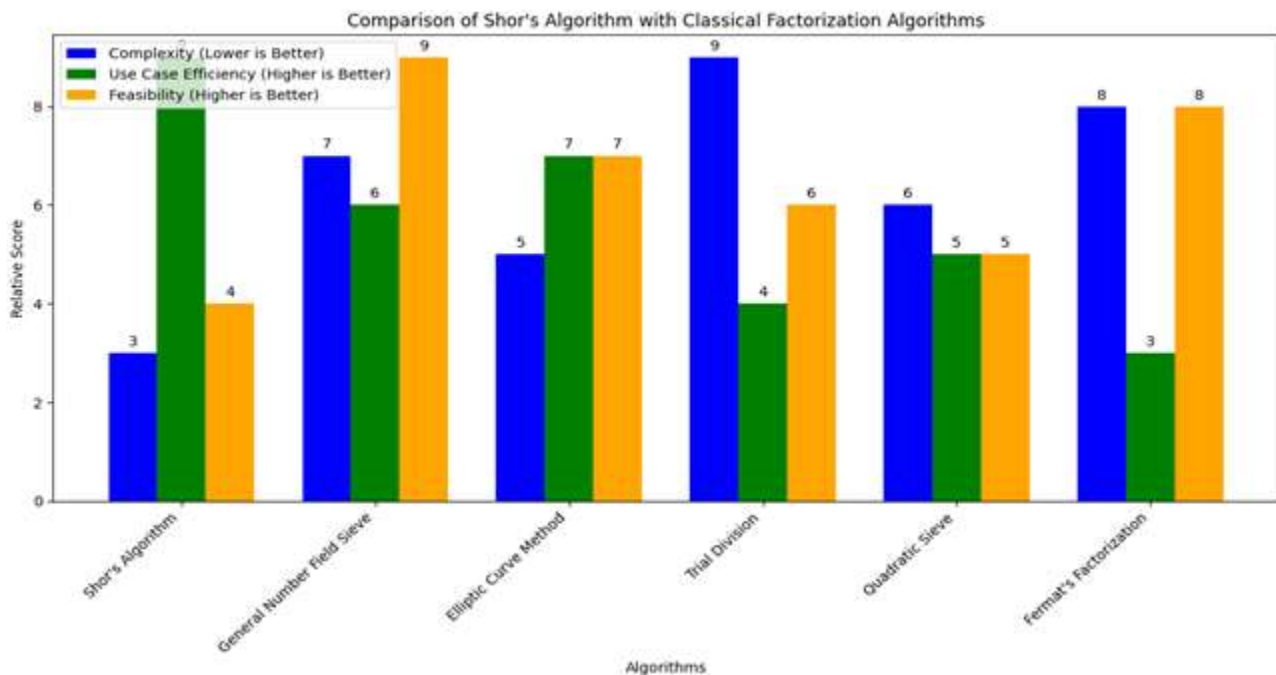


Figure 2. Comparison analysis

#### 4. Conclusions

With the rise of data breaches in government and corporate systems worldwide, privacy and data security have become more critical than ever. Quantum computers, particularly through Shor's algorithm, pose significant threats to current cryptographic systems. As a result, researchers in quantum cryptology are actively exploring solutions to enhance data security. Shor's algorithm is notably more effective at breaking encryption techniques than traditional computational methods. This study determines the efficiency of Shor's algorithm, comparing it to other algorithms based on complexity (inverse), use-case efficiency, and feasibility. Shor's algorithm demonstrated clear superiority across these criteria. Quantum cryptography offers advantages over traditional encryption, including unconditional eavesdropping detection and enhanced security. These features position it as a solution to future Internet security challenges, securing applications like the Internet of Things (message transmission, data access, and encrypted file storage) and next-generation communication networks. With its unmatched security and eavesdropping detection, quantum cryptography is well-suited to the evolving needs of cybersecurity. Finally, according to the disruptive potential of Shor's algorithm, it is important for future research to focus on the creation and use of post-quantum cryptography methods. Lattice-based cryptography, code-based cryptography, and the

number theory research unit (NTRU) encryption scheme are some of the promising options that can protect against quantum attacks.

**Acknowledgments:** We gratefully acknowledge the College of Computer Science and Mathematics at Tikrit University for providing research facilities and resources that facilitated this study.

**Conflict of Interest:** The authors declare no conflict of interest.

**Funding statement:** This work did not receive any fund.

#### References

- [1] Iqbal, S. S.; Zafar, A.; "Enhanced Shor's algorithm with quantum circuit optimization". *Int. J. Inf. Technol.*, 16 (4): 2725–2731, 2024.
- [2] Willsch, D.; Willsch, M.; Jin, F.; De Raedt, H.; Michielsen, K.; "Large-Scale Simulation of Shor's Quantum Factoring Algorithm". *Mathematics*, 11 (19): 4222, 2023.
- [3] Pavlenko, V.; Hlukhov, V.; "Quantum-inspired computing: comparison of variants of Shor's algorithm". *Adv. Cyber-Physical Syst.*, 10 (1): 71–76, 2025.
- [4] Bastos, D. C.; Brasil Kowada, L. A.; "How to detect whether Shor's algorithm succeeds against large integers without a quantum computer". *Procedia Comput. Sci.*, 195: 145–

- 
- 151, 2021.
- [5] Abdulwahhab, S. A.; Hussien, Q. M.; Taha, I. F.; "New Design of NTRU Key Using QKD Variants". in AIP Conference Proceedings 3009: 030007, 2024.
- [6] Alaa Abdulwahhab, S.; Mohammed Hussein, Q.; Fakhri Al-Shaikhli, I.; "An overview of number theory research unit variant development security". *Indones. J. Electr. Eng. Comput. Sci.*, 28 (2): 1164, 2022.
- [7] Cai, J.-Y.; "Shor's algorithm does not factor large integers in the presence of noise". *Sci. China Inf. Sci.*, 67 (7): 173501, 2024.
- [8] Buchanan, W.; Woodward, A.; "Will quantum computers be the end of public key encryption?". *J. Cyber Secur. Technol.*, 1 (1): 1–22, 2017.
- [9] "An overview of Quantum Cryptography and Shor's Algorithm". *Int. J. Adv. Trends Comput. Sci. Eng.*, 9 (5): 7487–7495, 2020.
- [10] Wong, H. Y.; "Shor's Algorithm". in *Introduction to Quantum Computing* 289–298, Springer International Publishing; Cham, 2024.
- [11] Kumar, M.; Mondal, B.; "Study on Implementation of Shor's Factorization Algorithm on Quantum Computer". *SN Comput. Sci.*, 5 (4): 413, 2024.
- [12] Muruganantham, B.; Shamili, P.; Kumar, S. G.; Murugan, A.; "Quantum cryptography for secured communication networks". *Int. J. Electr. Comput. Eng.*, 10 (1): 407, 2020.
- [13] Hamdi, S. M.; Zuhori, S. T.; Mahmud, F.; Pal, B.; "A Compare between Shor's quantum factoring algorithm and General Number Field Sieve". in *2014 International Conference on Electrical Engineering and Information & Communication Technology* 1–6, IEEE; 2014.
- [14] Larasati, H. T.; Kim, H.; "Quantum cryptanalysis landscape of shor's algorithm for elliptic curve discrete logarithm problem". in *International Conference on Information Security Applications* 91–104, Springer; 2021.
- [15] Jia, S.; "Comparison of Performances for Quantum and Conventional Algorithms: Shor's algorithm and Boson Sampling". *Highlights Sci. Eng. Technol.*, 38: 493–501, 2023.
- [16] Varshney, S.; Charpe, P.; Padmavathy, R.; Pal, S. K.; "Relation collection using Pollard special-q sieving to solve integer factorization and discrete logarithm problem". *J. Supercomput.*, 77 (3): 2734–2769, 2021.
- [17] Ali, F.; Mustafa, M.; "IMMFV2: Improved Modified Fermat Factorization Algorithm". in *International Conference on Engineering Software for Modern Challenges* 114–122, Springer; 2021.