RSA Prime Factorization on IBM Qiskit

Matthew Evans Audric Rengkung[1] and Arya Wicaksana[2*]

[1]Research Scholar, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia. matthew.rengkung@student.umn.ac.id, Orcid: https://orcid.org/0009-0009-6819-2612

[2*]Associate Professor, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia. arya.wicaksana@umn.ac.id, Orcid: https://orcid.org/0000-0002-0888-036X

Received: March 10, 2023; Accepted: April 13, 2023; Published: May 30, 2023

Abstract

The advancement of quantum computing in recent years poses severe threats to the RSA public-key cryptosystem. The RSA cryptosystem fundamentally relies its security on the computational hardness of number theory problems: prime factorization (integer factoring). Shor's quantum factoring algorithm could theoretically answer the computational problem in polynomial time. This paper contributes to the experiment and demonstration of Shor's quantum factoring algorithm for RSA prime factorization using IBM Qiskit. The performance of the quantum program is evaluated based on user time and the success probability. The results show that a more significant public modulus N in the RSA public key improves factorization's computational hardness, requiring more quantum bits to solve. A further enhancement on implementing Shor's oracle function is essential in increasing success probability and reducing the number of shots required.

Keywords: IBM Qiskit, Prime Factorization, Quantum Factoring, RSA, Shor's Algorithm.

1 Introduction

Integer factoring is a critical security factor of today's commonly used public-key cryptography due to its computational hardness (Jiang et al., 2018). The applications of integer factoring in security are RSA encryption and digital signatures, where RSA has been the existing standard for encryption over the internet (Salem et al., 2020). The problem stated that an attacker could efficiently compute the private key by factoring the public modulus N of the RSA public key into its prime factors p and q. However, no known algorithm could efficiently factor any integer with hundreds of decimal digits. Factoring the 1024-bit key with the state-of-the-art approach requires around 500,000 core years of computation (Boudot et al., 2022). The establishment of quantum computing pushes forward the progress on solving integer factoring and cracking the RSA cryptosystem (Nordrum, 2016).

Peter W. Shor showed that a quantum computer could carry out the factoring with much less computational effort for sufficiently large N (Gerjuoy, 2005). The exact size of the quantum computer needed to break RSA depends on several factors, such as the number of bits in the modulus and the error rate of the quantum computer. The rapid growth of quantum computing technologies proves that cracking the RSA is imminent (Anthony & Wicaksana, 2019; Wicaksana et al., 2020; Wicaksono & Wicaksana, 2019). State-of-the-art quantum technologies such as IBM Qiskit that provides quantum

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 2 (May), pp. 203-210.
DOI: *10.58346/JISIS.2023.I2.013*

*Corresponding author: Associate Professor, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia.

simulation could be used for integer factorization to emphasize the vulnerabilities of RSA to quantum computing and the need for more secure public-key cryptography and digital signatures in the future.

Related works on the topic began as early as 2001, experimenting with realizing Shor's algorithm with nuclear magnetic resonance (Vandersypen et al., 2001). The work implemented Shor's algorithm to factor N = 15. The most recent work uses IBM quantum processors, extending complexity by factoring N = 21 (Skosana & Tame, 2021). The work uses quantum processors and successfully creates entanglement with only five quantum bits (qubits). An experiment using IBM Qiskit (quantum simulator) is proposed in this paper as a more accessible and affordable approach compared to the two related works.

Implications for the success of Shor's quantum factoring algorithm realization on full-scale quantum computers are harmful to cybersecurity in general (Mavroeidis et al., 2018; Veliche, 2018), which is heavily dependent on cryptosystems such as RSA, as has been demonstrated in previous studies where theoretically and practically the quantum factoring algorithm is successful in factoring primes efficiently. Apart from the availability of full-scale quantum computers and access to them (Brooks, 2023; MIT Technology Review Insights, 2023), several quantum computing simulators are already available for the public to use for free. Although its current limitation is the number of quantum bits and simulation capacity, quantum computing poses a real threat and imminent danger to cryptography (Vaishnavi & Pillai, 2021).

This paper aims to program and simulate Shor's quantum factoring algorithm in breaking down the RSA cryptosystem using IBM Qiskit. The scope of this paper is to crack the RSA cryptosystem with prime size tailored to the availability of the number of quantum bits and simulation capacity currently provided by the IBM Qiskit. This paper contributes to demonstrating and experimenting with Shor's quantum factoring algorithm for the RSA prime factorization problem on IBM Qiskit. The study extends the experimental analysis in (Skosana & Tame, 2021) to break the RSA cryptosystem in a limited testing environment. Simulation time (user time) and the number of shots required for each test case are gathered for evaluation, including the success probability.

The rest of this paper is organized as follows. Section 2 describes the preliminaries of the paper. Section 3 explains the research methods. Section 4 presents the experimental results and discussion. Finally, Section 5 concludes this paper with suggestions on future work.

2 Preliminaries

RSA Cryptosystem

The RSA cryptosystem security primarily relies on the difficulty of factoring large integers. RSA is one of the most widely used public-key encryption schemes for communication over the Internet (Lindner, R., 2011). The RSA comprises three significant parts: key generation (private and public), encryption, and decryption. The stages of making RSA key pair and the encryption and decryption functions are as follows (Rivest et al., 1978).

1. Choose two distinct prime numbers, p and q.
2. Calculate n = pq, which is the modulus for the public and private keys.
3. Calculate the Euler totient function of n: φ(n) = (p-1) (q-1).
4. Choose an integer e such that 1 < e < φ(n) and gcd (e, φ(n)) = 1. This is the public exponent.
5. Calculate d as the modular multiplicative inverse of e modulo φ(n): de ≡ 1 (mod φ(n)). This is the private exponent.

6. The RSA public key is (n, e) and private key is (n, d).

The encryption function is carried out with the mathematical operation *plaintext*^{e} mod n. The decryption function is carried out with the mathematical operation *ciphertext*^{d} mod n (Rivest et al., 1978). If an attacker could efficiently factorize n back into its prime factors: p and q, they would be able to compute φ(n) and proceed to deduce the private exponent, d. This would compromise the security of the encryption scheme.

Number theory provides the mathematical foundation for RSA's security. The security of RSA rests on the assumption that factoring large numbers into their prime factors is computationally challenging, especially when the numbers involved are large prime numbers. No classical and efficient algorithm has been discovered to factorize large numbers in polynomial time, making RSA a secure encryption scheme based on number theory principles. Quantum computers with Shor's algorithm could factor large integers significantly faster than classical computers. As a result, if a large enough quantum computer is built, it could break the RSA cryptosystem by factoring the modulus used in the public key. This would allow an attacker to decrypt the encrypted messages and recover the plaintext (Gidney & Eker˚a, 2021).

Shor's Quantum Factoring Algorithm

Shor's quantum algorithm solves the Prime Factorization Problem that previously could not be achieved in Polynomial Time with high probability (Skosana & Tame, 2021). Shor's algorithm is an efficient way of factoring large numbers that underlie many commonly used cryptographic systems (Lomonaco Jr., 2000). The algorithm is summarized as the following (Beauregard, 2003; Lomonaco Jr., 2000).

1. Choose a large integer to be factored, n, that is the product of two prime numbers, p and q.
2. Choose a random number a between 1 and n-1, such that a and n are coprime (i.e., they have no common factors).
3. Define a quantum function f(x) = (a^x) mod n, which maps a non-negative integer x to a value between 0 and n-1.
4. Use a quantum computer to find the period r of the function f(x), which is the smallest positive integer such that a^r is congruent to 1 modulo n. This can be done using the quantum Fourier transform (QFT).
5. If r is odd or if $a^{r/2}$ is congruent to -1 modulo n, go back to step 2 and choose a different value of a.
6. If r is even and $a^{r/2}$ is not congruent to -1 modulo n, then the factors of n can be found by computing gcd ($a^{r/2}$ + 1, n) and gcd ($a^{r/2}$ - 1, n).

Part of the process that uses the principles of quantum mechanics on the Shor algorithm is in the fourth step of the Shor algorithm, which has been described. Here is the formulation of the quantum Fourier transform used in step 4 (Beauregard, 2003; Lomonaco Jr., 2000):

1. Let x be a non-negative integer with k bits, and let U be a unitary operator defined by U|x⟩ = |f(x)⟩, where f(x) = (a^x) mod n. Then the quantum Fourier transform of |x⟩ is defined by:
2. QFT|x⟩ = 1/sqrt(2^k) * sum (y=0 to 2^{k-1}) exp(2*pi*ixy/2^k) |y⟩, where |y⟩ is the state corresponding to the integer y, and i is the imaginary unit. Hadamard and controlled phase gates could be used to create a circuit on a quantum computer to implement the QFT.

This section is part of the search for the period r of the periodic function a^x mod n. This order-finding section consists of a modular multiplication function which will describe the results of the periodic function a^x mod n into a superposition state and an inverse Quantum Fourier Transform (QFT) function to obtain the value of r or period of the periodic function after being spelled out in the superposition state (Beauregard, 2003; Lomonaco Jr., 2000).

The implications of Shor's algorithm for RSA prime factorization are the breaks of RSA security, including the direct impact on related data confidentiality. Since RSA encryption relies on the assumption that factoring large composite numbers into their prime factors is computationally challenging, Shor's algorithm could efficiently factorize large numbers on a quantum computer, which means it can quickly find the prime factors of the RSA modulus, rendering RSA encryption vulnerable to attacks. This poses a significant threat to the security of RSA-encrypted data. Furthermore, If an attacker could factorize the RSA modulus used for encryption, they could calculate the private exponent and decrypt any intercepted ciphertext. This compromises the confidentiality of encrypted data, including sensitive information such as financial transactions, personal communications, and classified documents (Mavroeidis et al., 2018; Vaishnavi & Pillai, 2021; Veliche, 2018).

IBM Qiskit

IBM Qiskit is an open-source software development kit (SDK) for quantum computing. It is developed and maintained by IBM and is designed to help users write programs and algorithms for quantum computers and simulate quantum circuits on classical computers. Qiskit provides various tools for creating and manipulating quantum programs and running them on IBM Q Experience's own prototype quantum devices or from a simulator on a local computer. IBM Q Experience provides a website with features that could assist in quantum research using Qiskit. Qiskit supports a variety of backends, including IBM's cloud-based quantum computers, local simulators, and other third-party quantum hardware. It also provides tools for monitoring and analyzing quantum computations, including job status tracking, quantum state visualization, and quantum error diagnostics. A list of quantum computers and quantum simulators on the website could be used publicly in quantum research. Every time a quantum circuit is run on a quantum simulator, a job is created and sent to the queue owned by each quantum simulator. This web page allows users to manage and monitor all job lists implemented. Qiskit provides a quantum circuit implementation based on the Shor algorithm, which is open source, and it provides several functions that could be used for manufacturing Shor quantum circuits (IBM, 2022a).

3 Methods

The design of the quantum program for Shor’s quantum factoring algorithm uses the general implementation of IBM Qiskit’s Shor algorithm with 4x + 2 quantum bits (qubits), where x is the bit size of the number to be factored. The quantum program developed on Qiskit in this paper aims to demonstrate Shor’s algorithm for the prime factorization of RSA. Its security is based on the assumption that factoring large numbers is computationally challenging. If an attacker applies Shor's algorithm to a quantum computer with enough qubits, they could quickly find the RSA modulus's prime factors. Once the prime factors are known, the attacker could compute the private key from the public key, which breaks the encryption. If an attacker could factorize the RSA modulus using Shor's algorithm, they could decrypt any intercepted ciphertext and access the original message. This compromises the confidentiality of encrypted data and poses a significant security threat.

Due to the limited availability of 26 qubits on Qiskit when this paper was written, it is not feasible to experiment with the program on the actual RSA cryptosystem. In quantum computing, one qubit could take the value of two bits; two qubits could take the values of four bits, and so on. Thus, a smaller model of the RSA cryptosystem is required, tailored to the specification of the quantum program and simulator capacity. The RSA algorithm’s implementation is based on the necessary RSA steps by decrypting each letter in the string. Figure 1 presents the application flowchart describing the whole process designed for the application.

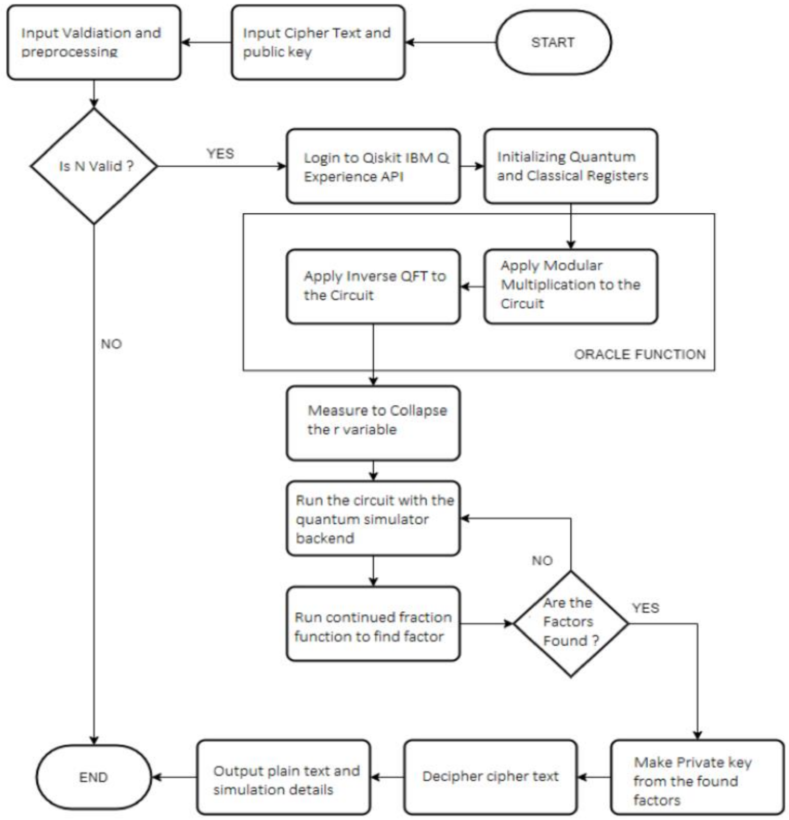
START
Input Cipher Text and public key
Input Valdiation and preprocessing
Is N Valid ?
YES
NO
Login to Qiskit IBM Q Experience API
Initializing Quantum and Classical Registers
Apply Modular Multiplication to the Circuit
Apply Inverse QFT to the Circuit
ORACLE FUNCTION
Measure to Collapse the r variable
Run the circuit with the quantum simulator backend
Run continued fraction function to find factor
Are the Factors Found ?
NO
YES
Make Private key from the found factors
Decipher cipher text
Output plain text and simulation details
END

Figure 1: Flowchart of the Quantum Program

As illustrated in Figure 1, the quantum program searches for the factors by measuring the collapsing qubits from each shot, and it continues to repeat this process until the factors are found. The program uses the factors to recompute the private key and crack the ciphertext of this paper's specially designed RSA cryptosystem. The quantum program is implemented on the web connected to the IBM Qiskit. The web implementation of the quantum program consists of six steps, including Shor’s oracle function:

1. Validate public key input and ciphertext.
2. Determine if the number N is a perfect power.
3. Initialize quantum and classical registers.
4. Perform modular exponentiation and inverse Quantum Fourier Transform (QFT).
5. Period finding by storing the period r (search results) in classical registers.
6. Confirm the factors when found.

Six test cases are made to evaluate the performance of the quantum program. The test is run using the backend ibmq_qasm_simulator. Test cases and variables are set in Table 1.

Table 1: Simulation Scenarios and Test-Cases

No.	Primes	N	Bits	Qubits	Public Key	Ciphertext
1	3x5	15	4	18	(15,3)	GAMIL
2	3x7	21	5	22	(21,5)	POBTQI
3	5x7	35	6	26	(35,5)	IOaX
4	3x13	39	6	26	(39,5)	JETRCV\C
5	3x17	51	6	26	(51,3)	[RA [EOie
6	3x19	57	6	26	(57,5)	SMcHaAEgcHuJM^

4 Results and Analysis

The test results are shown in Table 2 consisting number of shots and simulation time for each test case. The number of shots and the average shots' duration required to vary according to the number of qubits used due to the probabilistic characteristic of Shor's quantum factoring algorithm. The summary of the testing is given in Table 3.

Table 2: Test Results

Test-case No.	Shot No.	Simulation Time (s)	Test-case No.	Shot No.	Simulation Time (s)
1	1	27.3	6	1	652.8
2	1	75.5		2	717.7
	2	52.4		3	739.9
3	1	867.4		4	713.4
	2	767.2		5	775.9
	3	598.7		6	869.7
4	1	626.7		7	508.3
	2	567.6		8	571.9
	3	591.5		9	841.3
5	1	547		10	662.8
	2	541.3		…	
	3	544.7		46	748.3
	4	547.2			
	5	542.6			
	6	616.8			
	7	554			
	8	552			
	9	653.4			
	10	543.9			

Table 3: Test Summary

No.	N	Qubits	User Time (s)	Shots	Average Shots' Duration (s)	Plaintext
1	15	18	31	1	27.3	MAGIC
2	21	22	194	2	63.95	DOKTER
3	35	26	2,395	3	744.43	DOCS
4	39	26	1,935	3	595.27	DEKRIPSI
5	51	26	6,417	10	564.29	CRACKING
6	57	26	34,086	-	-	-

The varying number of shots for each test case shows the unreliability of the quantum program regarding the success probability. Enhancement is required to improve the oracle function and Shor's quantum circuit in increasing the success probability. The decryption results of each test have produced plaintext with 100% accuracy for the five test cases except for test case number six. The simulation is halted after 46 failed shots in determining the factors. The availability of more qubits would eventually allow the program to complete its task.

This paper emphasizes the effect of quantum computing on public-key cryptosystems, specifically the RSA. The research and development of quantum computers would expand the number of qubits available for quantum simulation. In the following years, fully working and scalable quantum computers could render the current RSA cryptosystem useless. It is shown in this paper that the only barrier to cracking the RSA is the availability of an adequate amount of qubits with a higher success probability.

5 Conclusions

This paper demonstrates Shor's quantum factoring algorithm for RSA prime factorization on the IBM quantum simulator: Qiskit. The experiment proves that the quantum program could solve the prime factorization problem with N = 51 in under two hours, which is more efficient than any existing classical algorithms. Notably, this is achieved using a quantum simulator with only 26 qubits. Although, the current RSA cryptosystem could use a more significant number of N to give extra computational hardness against the quantum attack. Nevertheless, The development of quantum technologies in the future would ultimately render the RSA cryptosystem useless, including other public-key cryptosystems and digital signatures that rely on similar mathematical foundations for security.

The limitation of this paper is the availability of qubits and access to a more powerful quantum simulator. Besides technological limitations, optimizing the quantum program and simulator is essential to achieve more efficient and effective results. Suggestions for future works include the optimization of Shor's quantum circuit to reduce the number of qubits required for simulation and enhancing the implementation of the oracle function to reduce the number of shots required significantly. This paper finally contributes to the demonstration and experiment of Shor's quantum factoring algorithm for the integer factoring problem in the RSA cryptosystem using a state-of-the-art quantum simulator from IBM.

6 Acknowledgements

Universitas Multimedia Nusantara fully supported this work. The authors would like to thank the reviewers and the associate editors for their valuable input on the paper.

References

[1] Anthony, A., & Wicaksana, A. (2019). Implementation of Grover's Quantum Search Algorithm using Rigetti Forest. *International Journal of Engineering and Advanced Technology (IJEAT), 8*(653).

[2] Beauregard, S. (2003). Circuit for Shor's algorithm using 2n+3 qubits. *Quantum Information and Computation, 3*(2), 175–185.

[3] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., & Thomé, E. (2022). The State of the Art in Integer Factoring & Breaking Public-Key Cryptography. *IEEE Security and Privacy Magazine*, *20*(2), 80-86.

[4] Brooks, M. (2023). *What's next for quantum computing*. MIT Technology Review. https://www.technologyreview.com/2023/01/06/1066317/whats-next-for-quantum-computing/

[5] Gerjuoy, E. (2005). Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. *American journal of physics*, *73*(6), 521-540.

[6] Gidney, C., & Ekerå, M. (2021). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum, 5*, 433.

[7] IBM. (2022a). *Qiskit 0.39.2 documentation*. Qiskit. https://qiskit.org/documentation/

[8] IBM. (2022b). *Qiskit Runtime*. IBM. https://www.ibm.com/quantum/qiskit-runtime

[9] Jiang, S., Britt, K. A., McCaskey, A. J., Humble, T. S., & Kais, S. (2018). Quantum annealing for prime factorization. *Scientific reports, 8*(1), 1-9.

[10] Lomonaco Jr., S.J. (2000). *A Lecture on Shor's Quantum Factoring Algorithm Version 1.1*.

[11] Lindner, R., & Peikert, C. (2011). Better key sizes (and attacks) for lwe-based encryption. *In Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA'11), San Francisco, 6558 of Lecture Notes in Computer Science, 319–339. Springer Berlin, Heidelberg*.

[12] Mavroeidis, V., Vishi, K., Zych, M.D., & Jøsang, A. (2018). The Impact of Quantum Computing on Present Cryptography. *International Journal of Advanced Computer Science and Applications*, *9*(3).

[13] MIT Technology Review Insights. (2023). *Delivering a quantum future*. MIT Technology Review. https://www.technologyreview.com/2023/04/07/1069778/delivering-a-quantum-future/

[14] Nordrum, A. (2016). Quantum computer comes closer to cracking RSA encryption. *IEEE Spectrum*, *3*.

[15] Rivest, R.L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, *21*(2), 120–126.

[16] Salem, F.K.A., Arab, M. Al, & Yang, L.T. (2020). Extending the limits for big data RSA cracking: Towards cache-oblivious TU decomposition. *Journal of Parallel and Distributed Computing*, *138*, 65–77.

[17] Skosana, U., & Tame, M. (2021). Demonstration of Shor's factoring algorithm for N= 21 on IBM quantum processors. *Scientific reports*, *11*(1), 1-12.

[18] Vaishnavi, A., & Pillai, S. (2021). Cybersecurity in the Quantum Era-A Study of Perceived Risks in Conventional Cryptography and Discussion on Post Quantum Methods. *Journal of Physics: Conference Series*, *1964*(4), 1-12.

[19] Vandersypen, L.M.K., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H., & Chuang, I.L. (2001). Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, *414*, 883–887.

[20] Veliche, A. (2018). Shor's Algorithm and Its Impact On Present-Day Cryptography. *no. Math*, *4020*, 1-19.

[21] Wicaksana, A., Anthony, A., & Wicaksono, A.W. (2020). Web-app realization of Shor's quantum factoring algorithm and Grover's quantum search algorithm. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(3), 1319-1330.

[22] Wicaksono, A.W., & Wicaksana, A. (2019). Implementation of Shor's quantum factoring algorithm using projectQ framework. *International Journal of Engineering and Advanced Technology (IJEAT)*, *8*(653).

Author Biography

Matthew Evans Audric Rengkung received a BSc in Informatics from Universitas Multimedia Nusantara, Indonesia, in 2020. His research interest is quantum computing and cryptography, and he is currently working at a web3 startup as a DevOps Engineer.

Arya Wicaksana is an associate professor at the Department of Informatics at UMN. He received a Master's Degree in research in VLSI Engineering from Universiti Tunku Abdul Rahman (UTAR). He successfully demonstrated the UTAR first-time success ASIC design methodology on a multi-processor system-on-chip project using 0.18μm processing technology 2015. His main research interests are blockchain applications and computational intelligence. He recently worked on blockchain-based decentralized autonomous social media. He has been an invited reviewer in IEEE ACCESS, QEIOS, IJNMT, and IFERP and an invited author in Intech Open and other scientific publications.