Advances in Global Innovation & Technology

Recebido
Received
Recibido
Jun. 2024

Aceito
Accepted
Aceptado
Dez. 2024

Publicado
Published
Publicado
Jan./Mar. 2025
Ene./Mar. 2025

https://**git.fateczl.edu.br**

e_ISSN
2965-3339

DOI
10.29327/2384439.3.2-3

São Paulo
v. 3 | n. 2
v. 3 | i. 2
e32015
Janeiro-Março
January-March
Enero-Marzo
2025

Sharing sensitive supply chain information: A study on vulnerabilities in RSA cryptography systems using *Shor's Algorithm*

Compartilhamento de informações sensíveis da cadeia de suprimentos: Um estudo sobre vulnerabilidades nos sistemas de criptografia RSA a partir do uso do algoritmo de shor

Compartir información confidencial de la cadena de suministro: Un estudio sobre las vulnerabilidades en los sistemas de cifrado RSA por el uso del algoritmo shor

Rodrigo de Souza [1]
rodrigo.souza119@fatec.sp.gov.br

Antonio Cesar Galhardi [2]
antonio.galhardi@cpspos.sp.gov.br

1 – Fatec of Franco da Rocha
2 - Postgraduate, Extension and Research Unit - Centro Estadual de Educação Tecnológica Paula Souza

Abstract:

This work proposes a comparative analysis of algorithms, considering Shor's Algorithm as a basis. The ChatGPT artificial intelligence tool is used in the work, asking it to create codes in the Qiskit language and quantum computing in August 2023 and February 2024. The results showed that there was considerable learning from the ChatGPT tool concerning better strategies for implementing the Algorithm in such a way that, considering the last suggestion from February 2024, there is a considerable risk to RSA encryption if this type of code is implemented without supervision in remote access environments to quantum computers. It is noteworthy that the risk to RSA encryption has direct implications for contemporary logistics, especially in the sharing of sensitive information in the supply chain, increasing the need to develop information security solutions that make it possible to prevent or mitigate the consequences of such advancement.

Keywords: *cryptography, machine learning, Shor's Algorithm, cybersecurity.*

Resumo:

Este trabalho propõe uma análise comparativa de algoritmos, considerando-se como base o Algoritmo de Shor. No trabalho emprega-se a ferramenta de inteligência artificial ChatGPT, solicitando-a que elabore códigos na linguagem e computação quântica Qiskit em dois momentos: agosto de 2023 e fevereiro de 2024. Os resultados mostraram que houve um aprendizado considerável da ferramenta ChatGPT em relação às melhores estratégias de implementação do algoritmo, de tal forma que, considerando a última sugestão, de fevereiro de 2024, tem-se um risco considerável à criptografia RSA, caso este tipo de código seja implementado sem supervisão nos ambientes de acesso remoto a computadores quânticos. Destaca-se que o risco à criptografia RSA tem implicações diretas na logística contemporânea,

DOI: https://doi.org/10.29327/2384439.3.2-3

especialmente no compartilhamento de informações sensíveis na cadeia de suprimentos, aumentando a necessidade de desenvolvimento de soluções em segurança da informação que possibilitem prevenir ou atenuar as consequências de tal avanço.

Palavras-chave: criptografia, aprendizado de máquina, algoritmo de Shor, cibersegurança.

Resumen:

Este trabajo propone un análisis comparativo de algoritmos, basado en el algoritmo de Shor. En la obra se utiliza la herramienta de inteligencia artificial ChatGPT, pidiéndole que escriba códigos en el lenguaje Qiskit y la computación cuántica en dos momentos: agosto de 2023 y febrero de 2024. Los resultados mostraron que hubo un aprendizaje considerable de la herramienta ChatGPT en relación con las mejores estrategias para implementar el algoritmo, de tal manera que, considerando la última sugerencia, a partir de febrero de 2024, existe un riesgo considerable para el cifrado RSA, si este tipo de código se implementa sin supervisión en entornos de acceso remoto a computadoras cuánticas. Cabe destacar que el riesgo para el cifrado RSA tiene implicaciones directas en la logística contemporánea, especialmente en el intercambio de información sensible en la cadena de suministro, aumentando la necesidad de desarrollar soluciones de seguridad de la información que permitan prevenir o mitigar las consecuencias de dicho avance.

Palabras clave: *criptografía, machine learning, algoritmo Shor, ciberseguridad.*

DOI: https://doi.org/10.29327/2384439.3.2-3

1. INTRODUCTION

Cryptography is the technique of encrypting (coding) and decrypting (decoding) a message using a key. Its use dates back to periods before the advent of contemporary computing, but it was with the use of electromechanical computers during the Second World War, especially the German machine called Enigma, that cryptography reached a new level. Around the 1960s, with the emergence of semiconductors and integrated circuit architecture, computing began to have greater processing capacity, storage and agility.

The encryption (*Rivest-Shamir-Adleman)* emerged in 1978 as an asymmetric technology, that is, based on two keys, one of which is public and consists of the product of two private keys, which grant access to data. The benefit of the RSA model is the difficulty of factoring large prime numbers, which is the technique used to generate the public key. Breaking this type of encryption would require factoring a number into its prime constituents. This operation demands an extremely high level of processing due to the size of the number used in the public key (Uzeda et al., 2022).

RSA technology was free from threats until 1994 when American mathematician Peter Shor proposed an algorithm model that would require its implementation on quantum computers, and that would easily break RSA encryption since it is based on the power of quantum computing for factorization (Shor, 1994).

The increasing interconnectivity of contemporary supply chains, driven by web technologies and the need for operational efficiency, has brought to light a series of challenges concerning information security. Sharing sensitive information between different entities, such as suppliers, manufacturers, distributors and retailers, has been a common practice, as the aim is to optimize the flow of goods. However, this practice carries with it a substantial risk, especially in a scenario in which cybersecurity is a growing concern. Vulnerabilities in IT systems, lack of robust encryption and inefficient authentication methods can expose information to cyberattacks, resulting in data leaks, supply chain disruptions and loss of customer trust. This risk is amplified when considering the possibility of attacks originating from quantum computers, which could break traditional encryptions, such as RSA, much faster than classical computers. In this context, organizations must adopt preventive measures to ensure the integrity of sensitive information in their supply chains. Additionally, emerging technologies such as blockchain and artificial intelligence can help strengthen the security and resilience of supply chains by ensuring the integrity and confidentiality of shared information.

Until the mid-2000s, *Shor's Algorithm* was seen as much more as an academic curiosity than a practical instrument. However, given the advances in the development of quantum computers, their applicability went from a hypothetical situation to a possibility (VIEIRA; ALBUQUERQUE, 2020). Furthermore, with the emergence of generative artificial intelligence systems, the most famous and perhaps one of the most advanced of which is ChatGPT, launched in late 2022, the possibility of translating *Shor's Algorithm* into a quantum language such as *Qiskit*, which can be implemented in IBM's remote access quantum environment

DOI: https://doi.org/10.29327/2384439.3.2-3

called IBM Quantum Experience, opens up previously little-explored possibilities of inappropriate use of these tools in an attempt to break cybersecurity systems based on RSA encryption. In this study, the basic research question is: What is the level of assertiveness of the *Qiskit code proposed by ChatGPT for Shor*'s Algorithm? To answer this question, this article was initially based on state-of-the-art bibliographical research on quantum computing and generative AI. Then, empirical research was carried out. ChatGPT was asked to propose a *Qiskit code for Shor*'s Algorithm, which was subjected to a comparative analysis between the proposed code and the original Algorithm to verify the level of assertiveness. In a third moment, the same request was made to ChatGPT after 6 months to verify the platform's level of learning about the Algorithm's construction.

2. Theoretical Basis

Current computing is based on semiconductor technology, which is used in the construction of transistors, devices capable of conducting and not conducting electricity. In other words, the name semiconductor comes from the fact that the material used has conductive and insulating properties. This semiconductor property is essential for computers' functioning since the computer system is based on bits, which are the values of 0 (zero) and 1 (one), used in binary code.

Semiconductors are formed from a crystalline structure of silicon, an insulating material. Its crystalline structure is then doped (with the addition of small amounts) with elements that will make it a semiconductor. The doping process can be carried out in two distinct ways (Weste; Harris, 2009):

a) Negative doping: Phosphorus atoms with an extra electron are inserted;
b) Positive doping: Indium atoms are inserted with one less electron.

The absence of an electron or the presence of an extra electron will make the crystalline structure a semiconductor, enabling the binary coding process. Note that the operation of the transistor presents only one state at a time, or only one bit at a time can be read. In general, in negative doping, with the flow of current, there is bit 1, and in the absence of current, there is bit 0. In positive doping, the opposite occurs (Weste; Harris, 2009).

In quantum computing, processing is not done by transistors. Atomic sub-particles, such as electrons and photons, individually manipulate the elements used. Since the agitation of molecules is a thermodynamic phenomenon, a relevant aspect is the need to keep such computers at absolute zero, 0°K (-273ºC), since at higher temperatures, there will be agitation of subatomic particles. At absolute zero, only selected particles undergo agitation (Portugal; Marquezino, 2019).

The term "quantum" refers to using the precepts of quantum mechanics to determine and explain computational processes. Regarding these concepts, quantum entanglement is especially important since it is how information is exchanged in a quantum computer. Quantum entanglement is the parity of spin (angular momentum) of two subatomic particles that, once together, maintain this parity even when later separated (Toneli, 2022).

DOI: https://doi.org/10.29327/2384439.3.2-3

Heisenberg's uncertainty principle, according to which, in a simplified way, it is not possible to define the position and velocity of a subatomic particle with precision. The choice of one quantity implies the probabilistic determination of the second. From the perspective of the quantum computer, this applies to the possibility that a quantum bit (or *qubit*) can assume the value 0, the value 1 or both simultaneously. It is estimated that a computer with around 100 *qubits* currently has the same processing capacity as the sum of all computers worldwide (Parada, 2019).

2.1 Shor's Algorithm

Shor's Algorithm was developed by Peter *Shor* in 1994 and allows the factorization of a number N in polynomial time using a quantum computer. Its steps are (Shor, 1994):

a) Quantum State Preparation: A quantum state must be prepared in a superposition of all possible values for the Algorithm's input. *Qubits are used* in two registers, one for the input and one for the output. The input register is initialized in a superposed state with all possible input values, while the output register is initialized in a state with value 1.
b) Quantum Fourier Transform: A quantum Fourier transform must be applied to the input register. This transform converts the superposition of all possible values into a set of amplitudes corresponding to the number's prime factors to be factored.
c) Measurement and post-processing: To find the prime factors, the output register must be measured, and classical post-processing must be performed. The measurement of the output register collapses the superposition of states, resulting in a single value that represents one of the prime factors of the number to be factored. Once one factor is found, a classical factorization algorithm is used to find the other factor.

The pseudocode of the Algorithm, based on the work of Shor (1994), is shown below:

Start
1. Choose a random integer between 1 and N-1.
2. Calculate the greatest common divisor of a and N. If the gcd is not equal to 1, then a and N have a factor in common, and the factorization of N is found.
3. Create a quantum circuit that operates on the basis {|0>, |1>, ..., |N-1>} with a unitary gate U(x) that calculates a^x mod N.
4. Prepare a quantum register in a superposed state using sufficient *qubits* to store N.
5. Apply the U(x) gate to the superposed state. This creates a superposition of states representing all possible powers of a *mod* N.
6. Apply the quantum Fourier transform to the superposed state.
7. Measure the quantum recorder. The result of the measurement is a number r that is an integer multiple of the period of the function a^x mod N.

DOI: https://doi.org/10.29327/2384439.3.2-3

8. Calculate the greatest common divisor of N and a^(r/2) + 1. If the gcd is not 1, then the factorization of N is found. Otherwise, go back to step 1 and choose a new value for a.

End

The above pseudocode can be understood as follows: The Algorithm starts by choosing a random integer "a", which is used to calculate the possible powers of "a" *mod* N. Then, the greatest common divisor between "a" and N is calculated. If the gcd is different from 1, then the factorization of N is found. The next step is to create a quantum circuit that operates on the basis {|0>, |1>, ..., |N-1>} and include a unitary gate U(x) that calculates a^x mod N for each x in the basis. Next, a quantum register is prepared in a superposed state, and the U(x) gate is applied to the superposed state. This creates a superposition of states representing all possible powers of a mod N. The quantum Fourier transform is applied to the superposed state, and then the quantum register is measured. The result of the measurement is a number r that is an integer multiple of the period of the function a^x mod N. The period is the smallest positive integer r for which a^r mod N = 1. Finally, the greatest common divisor of N and a^(r/2) + 1 is calculated. If the gcd is different from 1 and different from N, then the factorization of N is found. Otherwise, the Algorithm must be repeated with a new random value of "a" (Shor, 1994). It is worth noting that the purpose of the " mod " function is to return the remainder of a division. The remainder of the division is a key element for the encryption and decryption processes.

2.2 RSA Encryption

The RSA encryption system is based on the following assumption: given prime factors p and q, with large values in each, such values form a private key of a public key n = p . q (CASTRO, 2019). This means that both encoding and decoding depend on p and q. This cryptographic approach obeys the Fundamental Theorem of Arithmetic in N, according to which considering a positive integer n = 1, it can be written as follows n=p_1 ^(e_1) ⋯ p_k^ek, where 1< p1...< pk and p = prime. In summary, any natural number can be decomposed into prime constituents.

The application of the RSA system requires converting letters into numbers and defining a number for the spaces between words. Once this is done, an "encoding exponent" is chosen, and the number corresponding to the message is raised by this exponent. Then, the result is divided by an "encryption module." The decoding process follows inverse logic (Teixeira, 2020).

2.3 ChatGPT

ChatGPT is a *chatbot*, a robot for chat conversations, based on generative Artificial Intelligence. Generative AI is usually based on an Artificial Neural Network and acts through free research on a given subject, performing a synthesis in the response, which is also stored as learning. This artificial intelligence approach is based on natural language processing, and this is

DOI: https://doi.org/10.29327/2384439.3.2-3

relevant given the need to bring the AI interface closer to the end user (Ramesh et al., 2017).

3. Theme Development

The research compares the proposed *Shor Algorithm in the Qiskit* quantum language, developed by ChatGPT, and the original Algorithm to verify whether there is a correspondence between both. This comparison occurred in two moments: the first in August 2023 and the second in February 2024. The use of two moments is due to the objective of investigating the level of learning and improvement of the platform concerning the proposed Algorithm. On the ChatGPT platform, the provocative text was "Make the *Shor Algorithm in the Qiskit* language". The result of the first code generated (August 2023) can be seen in Figure 1.

A high level of fidelity is observed when comparing the ChatGPT proposal line by line with Peter Shor's original Algorithm in pseudocode. It should be noted that the first three lines of the code in Figure 1 refer to the imported libraries. The proposed code also used a number for factorization (15) to exemplify its operability. The line-by-line analysis is performed as follows:

1. Shor: Choose a random integer a between 1 and N-1;
   ChatGPT: N = 15;
2. Shor: Calculate the greatest common divisor of a and N. If the gcd is not equal to 1, then a and N have a factor in common, and the factorization of N is found;
   ChatGPT: Not done;
3. Shor: Create a quantum circuit that operates on the basis {|0>, |1>, ..., |N-1>} with a unitary gate U(x) that calculates a^x mod N;
   ChatGPT: qc = QuantumCircuit(4, 4);
4. Shor: Prepare a quantum register in a superposed state using a sufficient number of qubits to store N;
   ChatGPT: qc.h(range(3))
   qc.x(3);
5. Shor: Apply the U(x) gate to the superposed state. This creates a superposition of states representing all possible powers of a mod N;
   ChatGPT: shor = Shor(N, a=2,
   quantum_instance = Aer.get_backend (' qasm_simulator '))
   qc. append (shor.circuit, qc. qubits [:3]);
6. Shor: Apply the quantum Fourier transform to the superposed state;
   ChatGPT: qc.append (QFT(3, inverse=True), qc.qubits [:3]);
7. Shor: Measure the quantum register. The measurement result is a number r that is an integer multiple of the period of the function a^x mod N;
   ChatGPT: qc.measure (range(3), range(3));
8. Shor: Calculate the greatest common divisor of N and a^(r/2) + 1. If the gcd is not 1, then the factorization of N is found. Otherwise, go back to step 1 and choose a new value for a;

DOI: https://doi.org/10.29327/2384439.3.2-3

ChatGPT: backend = Aer.get_backend (' qasm_simulator ')
job = execute(qc, backend, shots=1024)

Figure 1 – *Shor*'s Algorithm in *Qiskit* proposed by *ChatGPT* in August 2023

```
from qiskit import QuantumCircuit, Aer, execute
from qiskit.algorithms import Shor
from qiskit.circuit.library import QFT

# Define o número a ser fatorado
N = 15

# Cria um circuito quântico para o algoritmo de Shor
qc = QuantumCircuit(4, 4)

# Aplica as portas necessárias no primeiro registrador
qc.h(range(3))
qc.x(3)

# Aplica o algoritmo de Shor
shor = Shor(N, a=2, quantum_instance=Aer.get_backend('qasm_simulator'))
qc.append(shor.circuit, qc.qubits[:3])

# Aplica a transformada de Fourier inversa no primeiro registrador
qc.append(QFT(3, inverse=True), qc.qubits[:3])
```

Source: The authors (2024)

The other points of Shor's Algorithm were met except item 2. Although not proposed by ChatGPT, item 2 is not essential for the result since it is a priori verification of the factorization. As for the others, from a functional perspective, they were fully met.

The second generated code (February 2024) is shown in Figure 2.

DOI: https://doi.org/10.29327/2384439.3.2-3

Figure 2 – *Shor*'s algorithm in *Qiskit* proposed by *ChatGPT* in February 2024

```
from math import gcd
from random import randint
from qiskit import Aer, QuantumCircuit, transpile, assemble, execute
from qiskit.visualization import plot_histogram

def modular_exponentiation(a, exponent, N):
    """Perform modular exponentiation a^exponent mod N."""
    return pow(a, exponent, N)

def quantum_modular_exponentiation(qc, a, exponent, N, n):
    """Implement modular exponentiation a^exponent mod N using a quantum circuit."""
    for i in range(n):
        qc.x(i) if (exponent & (1 << i)) else None
    for i in range(n):
        quantum_multiple_by_a_mod_↓:, a, 2 ** i, N, n)
    return qc
def quantum_multiple_by_a_mod_N(qc, a, exponent, N, n):
    """Implement the controlled modular multiplication by a."""
    for i in range(n):
        qc.swap(i, n + i)
    for i in range(n):
        qc.cx(i, n + i) if (exponent & (1 << i)) else None
    for i in range(n - 1, -1, -1):
        qc.cx(i, n + i) if (exponent & (1 << i)) else None
    return qc

def find_period(N, a, n):
    """Find the period r of the function f(x) = a^x mod N."""
    qc = QuantumCircuit(n * 2, n)


    quantum_modular_exponentiation(qc, a, 1, N, n)

    for q in range(n):
        qc.measure(q, q)

    backend = Aer.get_backend('qasm_simulator')
    result = execute(qc, backend, shots=1).result()
    counts = result.get_counts(qc)

    measured_period = int(list(counts.keys())[0], 2)
    measured_period = measured_period if measured_period else 2 ** n

    return measured_period
def shor(N, n_trials=20):
    """Shor's Algorithm to find prime factors of N."""
    if N % 2 == 0:
        return 2

    for _ in range(n_trials):
        a = randint(2, N - 1)
        if gcd(a, N) != 1:
            return gcd(a, N)
        else:
            measured_period = find_period(N, a, n)
            if measured_period % 2 != 0:
                continue

            if modular_exponentiation(a, measured_period // 2, N) != N - 1:
                p = gcd(modular_exponentiation(a, measured_period // 2, N) + 1, N)
                q = gcd(modular_exponentiation(a, measured_period // 2, N) - 1, N)
                return p, q

# Example usage
N = 15
n = 4  # Number of qubits needed to represent N
factors = shor(N)
print("Factors of", N, "are:", factors)
```

Source: the authors, 2024.

DOI: https://doi.org/10.29327/2384439.3.2-3

After ChatGPT proposed *the* new code, it was analyzed about the original Algorithm. The new code (February 2024) proposed by ChatGPT, in addition to the items already met in the previous Algorithm (August 2023), also met item 2 (greatest common divisor), which had not been met in the first code generated. The next step was the comparative analysis of the algorithms to verify which of the two codes makes a more rational and optimized use of the *Qiskit tool*. The criteria for this analysis were (Johnston; Harrigan; Segovia, 2019):

a) Control and Flexibility: This is the ability to control every aspect of the Algorithm and implementation, as well as its flexibility to adjust and optimize different parts of the code according to the specific needs of the problem.

b) Implementation of the Quantum Fourier Transform (QFT): Checks how the QFT is implemented in the code. A manual implementation is considered to offer greater control over each detail of the QFT, allowing specific adjustments as needed. On the other hand, opting to use a predefined class simplifies the code; however, it can limit the accuracy of the calculations.

c) Customization and Optimization: This is the ability that the model offers to customize the Algorithm as a whole to meet the specific needs of the problem in question. Such customization includes adjusting parameters, optimizing parts of the code and adapting the Algorithm for different usage scenarios.

4. Results and Discussion

Regarding code control and its flexibility, it can be seen that in the February 2024 code, *Shor's Algorithm* is implemented manually, which offers greater control over each step of the process, including reference values for variables and possible approximations. In this way, it is possible to optimize each step of the Algorithm according to the specific characteristics of the problem, for example, by adjusting the choice of 'a' dynamically, maximizing the probability of finding the period quickly.

Shor's Algorithm is encapsulated in a standardized class provided by the *Qiskit library*. While this strategy offers convenience, it can limit control over the process, as some specific optimizations or tweaks may not be possible without directly modifying the source code of the “Shor ” class.

Regarding the implementation of the Quantum Fourier Transform (QFT), it is noted that in the February 2024 code, the QFT is implemented manually using a function defined in the code itself. Such a solution allows for greater control over the implementation of the QFT and the ability to adjust it for different register sizes or precision requirements.

In the August 2023 code, QFT is implemented using a standardized class provided by the *Qiskit* library. Again, this option provides the ease and agility of working with a predefined solution. However, this option may limit QFT's customization capacity, especially in situations where specific implementation is required.

Regarding code customization and optimization, it is noted that the features offered by the manual implementation of *Shor's Algorithm* and QFT in the

DOI: https://doi.org/10.29327/2384439.3.2-3

February 2024 code allow for greater customization and optimization for the specific problem at hand. For example, it is possible to adjust the parameters of *Shor's Algorithm*, such as the number of attempts to find prime factors or the accuracy of QFT, to improve the performance or efficiency of the Algorithm.

In the August 2023 code, customization options may be more limited due to the encapsulation of *Shor's Algorithm* and QFT in classes provided by the *Qiskit library*. Using these standardized resources may make it difficult or impossible to make specific adjustments to improve the Algorithm's performance or efficiency.

5. Final Considerations

Shor's Algorithm was used in the codes proposed by ChatGPT. However, the platform seems to have learned something in the 6 months between requests. This learning seems to have improved the code's ability to handle situations requiring factoring. In other words, the first proposal (August 2023) features massive use of standardized resources already available to users of the *Qiskit language*. However, in the February 2023 proposal, most of these resources were implemented “manually.”

The option for "manual" implementation of the resources is positive when one considers that the structure of RSA algorithms is not entirely standard. Therefore, for each RSA algorithm, there is a specific need to factor prime numbers, given the complexity level of the encryption technique. When using a code with many standardized elements, as proposed in August 2023, one may encounter problems that cannot be decomposed with such tools. On the other hand, opting for a customized solution, as observed in the February 2024 code, means the possibility of not breaking the encryption is lower since the code is adjustable to the specifications of the problem in question.

We see that the ChatGPT tool has apparently "learned" that *Shor's Algorithm* is best implemented when we leave aside the standard tools of the *Qiskit language* and use customized and adjustable codes. From an artificial intelligence point of view, this learning is very positive since the numerous iterations have significantly improved the Algorithm's construction. On the other hand, considering cybersecurity aspects, there is a greater risk since implementing this optimized code on a platform for remote access to quantum computers could open potential loopholes in cryptography systems, compromising sensitive operations such as blockchain and traditional financial transactions. Therefore, there is a latent need to improve the current cryptography systems.

Finally, the increasing dependence of logistics systems on the exchange of sensitive information demands the implementation of preventive measures to guarantee the integrity of data and information. Possibilities to be explored include the adoption of robust encryption, strict security policies and investment in emerging technologies, such as blockchain and artificial intelligence, to strengthen supply chains' security and resilience.

DOI: https://doi.org/10.29327/2384439.3.2-3

REFERENCES

CASTRO, C. C. **Criptografia RSA**. Trabalho de Conclusão de Curso - Universidade Federal de Santa Catarina, Departamento de Matemática (MAT), Centro de Blumenau, Curso de Licenciatura em Matemática, Blumenau, 2019.

JOHNSTON, E.R.; HARIGAN, N.; SEGOVIA, M.G. **Programming quantum computers**. O'Reilly Media, Inc., Sebastopol, 2029.

PARADA, I.U. **Aplicação de processamento quântico para aceleração da resolução de autômatos probabilísticos**. Trabalho de conclusão de curso (Bacharelado em Engenharia Mecatrônica) — Universidade de Brasília, Brasília, 2019.

PORTUGAL, R.; MARQUEZINO, F. **Introdução à Programação de Computadores Quânticos**. Sociedade Brasileira de Computação, 2019.

RAMESH, K.; RAVISHANKARAN, S.; JOSHI, A.; CHANDRASEKARAN, K. **A survey of design techniques for conversational agents**. International Conference on Information, Communication and Computing Technology, Springer, Singapore, 2017.

SHOR, P.W. **Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer**. 35th Annual Symposiumon Foundations of Computer Science, Santa Fe, NM, Nov. 20–22, 1994, IEEE Computer Society Press, pp. 124–134, 1994.

TONELI, D.A. **Desenvolvimento de um tutorial para o ensino de computação quântica**. Trabalho de Conclusão de Curso – Unifesp, 2022.

TEIXEIRA, M.A.F. **Números inteiros e criptografia RSA**. Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro, 2020.

UZEDA, E. E. F.; DALLASEN, R. V.; SANTOS, M. C.; ROLLWAGEN, A. F.; RIBEIRO, D. D.; FIGUEIREDO, J. A. O. **Utilização do algoritmo de Shor para quebra de criptografia RSA em computadores quânticos**. Salão do Conhecimento, Unijuí, Ijuí, 2022.

VIEIRA, L.A.; ALBUQUERQUE, C.D. **Um estudo passo a passo do Algoritmo de Shor**. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 7, n. 1, 2020.

WESTE, N.H.E., HARRIS, D. M. **CMOS VLSI Design A Circuits and Systems Perspective**. Addison – Wesley, New York, 2009.