State University of Campinas

Institute of Mathematics, Statistics and Scientific Computing

Python Implementation and Analysis of Quantum Information Theory Algorithms.

Marina de Souza Ripper RA:183930

2022.

Conteúdo

1 Introduction

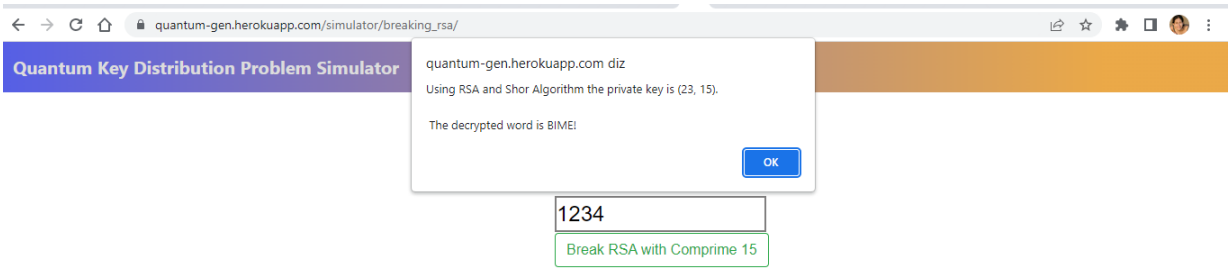
The present work was developed based on the Introduction to Quantum Information Sciences course, presented on videos 6.1 to 6.10 in the course Introduction to Quantum In- formation Sciences.

2 Practical Studies

The RSA (Rivest–Shamir–Adleman) cryptosystem is an algorithm which enables one group to encrypt and decrypt data while another to only decrypting. Two distinct pieces of information are required to obtain the full range of the RSA function, a public and a private key.

Based on the code developed on Applying Shor's Algorithm, I corrected some minor errors for it to run and adapted it for any entry, also making it user friendly, for this you may access the addition I made to the panel I created in dissertation 5 on:

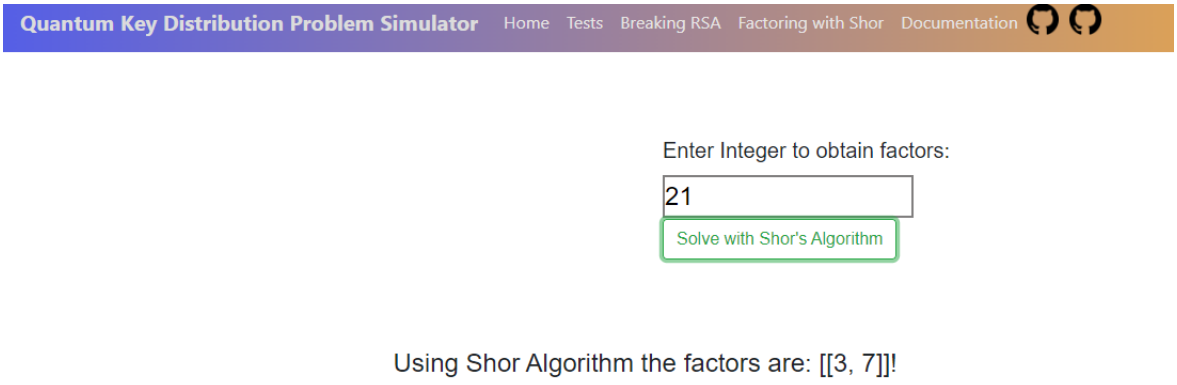
- Breaking RSA With Shor



Using RSA and Shor Algorithm the private key is (23, 15). The decrypted word is BIME!

I have also added a facturing of integers using shor algorithm implemented with thw IBM python lib qiskit, shown on the link below, as are tests shown in the subsequent image:

- Factoring integers with Shor



The factoring by shor is still costly, even though doing so in polynomial time, still, in no doubt less than classical computers.

All Algorithms composing the panel are in:

- Git Frontend
- Git Backend

3 Dissertation

Quantum computing comes to performing operations on qubits representing binary strings and eventually evaluating Boolean functions, the logical gates used to evaluate the Boolean functions such as the NOT and AND gates raise problems such as the reversibility of such operations, where the AND gate is not an easily reversible operation.

Reversibility, essentially defines the energy cost into computing, and arises the question of how to construct computation in a way you can reverse it.

Now quantum computing essentially is about quantum interference, when we have many qubits we start quantum interference with the hadamard transform, opening the quantum interference, then we introduce phase factor by using quantum function evaluation,finally we close quantum interference with haddamard or the fourier transform.

Instead of using the phase gate, the use of phase kickback where a second register with quantum function evaluation substitutes the first. Quantum computing comes in transforming function evaluation into phase factors. Additionally if we allocate $(0\rangle - (1\rangle$ it duplicates on the outcome of the second register and adds $(-1)^{f(x)}|x\rangle$ introducing interesting quantum interference.

With these concepts in mind we come to the quantum algorithms such as:

- Deutsch-Jozsa Algorithm: attempt to define if a function is constant where it returns all 0's or all 1's for any input, or a balanced function that returns 0's for exactly half of all inputs and 1's for the other half. In the quantum solution the addition of a $(0\rangle$ - $(1\rangle$ registry for the second registry introduces the $(-1)^{f(x)}|x\rangle$ which is always positive for constant function and so the output of the quantum operations is always 0 and

always negative for balanced functions and so the output of the quantum operations is always 1.

- Bernstein-Vazirani Algorithm: is an extension of the Deutsch-Jozsa algorithm, in this, the function is guaranteed to return the bit wise product of the input with some string,s. We can find s, given an input x, f(x)=sx(mod 2).

  The classical solution gives us: $f_s(x) = s \cdot x \mod 2$ given an input x. The hidden bit string s can be revealed by querying the oracle with a sequence of inputs so each query reveals a different bit of s, Essentially, this means we would need to call the function $f_s(x)$,n times.The idea of the quantum algorithm is to initialize the inputs qubits to $|0\rangle^{(\otimes_n)}$ state, and output qubit to $|$ -$\rangle$, apply Haddamard gates to the input register, apply the operation $|x\rangle \xrightarrow{f_s} (1)^{x \cdot s}|x\rangle$,then apply Haddamard gates to the input register and measure.

- Simon's Algorithm. Simon's problem is if f is (1:1) or two-to-one (2:1), where the first refers exactly one unique output for every input, and the second refers to exactly two inputs to every unique output.

  For the quantum algorithm we initialize 2 qubit input registers as the zero state, then we apply a Haddamard transform to the first register, we follow to apply the query function $\frac{1}{\sqrt{2^n}} \sum_{x0,1^n} |x\rangle|f(x)\rangle$ we then measure the second register and find a value of f, where f can be generated by input x or $y = x \otimes_b$. We then apply Haddamard on the first register, which will give an output if: $b \cdot z_i = 0 \mod 2$, repeating times we will have n z's from which we take b.

- Shor's Algorithm: Shor's algorithm relates to finding the factoring of large integers, which is difficult for classical computing, the idea is firstly to find the period x, where x is the smallest integer, besides 0, where $a^x \mod N = 1$, the quantum solution to find x, is to use the quantum phase estimation on the unitary operator U, where each time we apply U, we multiply the state of our register by a(mod N), and after x applications we have state $|1\rangle$ again.

  When quantum phase estimation is applied on U using the state $|1\rangle$, the phase n/x Where n is a random integer between 0 and x-1.With continued fractions algorithm on n/x, where continued fraction refers to decomposing an integer in iterations of sums of it's integer part and the reciprocal of another number, we then find x.

Referências

miscIQIS, author = Artur Ekert, title = Introduction to Quantum Information Sciences, howpublished = https://www.youtube.com/playlist?list=PLkespgaZN4gmu0nWNmfMflVRqw0VPkCGH, month = July, day = 20, year = 2021