QUANTUM CRYPTOGRAPHY - BREAKING RSA ENCRYPTION USING QUANTUM COMPUTING WITH SHOR'S ALGORITHM

Shaheed Nehal A[1], Mubasheer Farhan[2], Sunad Y S[3]
Dept. of Computer Science, JSSSTU, Mysuru, India

Abstract: ***Quantum computing is a quickly growing field of research thanks to recent hardware advances. The quantum mechanical properties of quantum computers allow them to solve certain families of problems faster than classical computers. One such problem is the unstructured searching problem and this can be solved in a much better efficiency on a quantum machine using the well known Grover's algorithm than the currently available best efficiency classical algorithm ie; linear search. Quantum p computing provides a quadratic speedup, O(N), for such a problem as compared to the linear efficiency provided by the classical algorithm, O(N), where N is the search space. Another very important application is the polynomial time quantum algorithm called Shor's Algorithm for factoring integers and computing discrete logarithms. Shors algorithms were the first quantum algorithms that achieved an exponential speedup over classical algorithms, applied to problems outside the field of quantum mechanics, and had obvious applications. In particular, Shors algorithms may be used to break the RSA cryptosystem based on the hardness of factoring integers that are the product of two similarly-sized primes, and cryptosystems based on the discrete logarithm problem (DLP), such as the Diffie-Hellman key agreement protocol and the Digital Signature Algorithm. The most expensive operation performed by Shors factoring algorithm is a modular exponentiation. Modern classical com-puters can perform modular exponentiations on numbers with thousands of bits in under a second. These two facts may at first glance appear to suggest that factoring a thousand bit number with Shors algorithm should only take seconds, but unfortunately (or perhaps fortunately), that is not the case. The modular expo-nentiation in Shors algorithm is performed over a superposition of exponents, meaning a quantum computer is required, and quantum hardware is expected to be many orders of magnitude noisier than classical hardware. This noise necessitates the use of error correction, which introduces overheads that ultimately make performing reliable arithmetic on a quantum computer many orders of magnitude more expensive than on classical computers.***
Although Shors algorithms run in polynomial time, the con-stant factors hidden by the asymptotic notation are substantial. These constant factors must be overcome, by heavy optimization at all levels, in order to make the algorithms practical. Current quantum computers are far from being capable of executing Shors algorithms for cryptographically relevant problem sizes. In this paper, an approach and experiment to implement Shors quantum factoring algorithm are proposed. The implementation is done using Python and a quantum computer simulation using Pyquil.

Index Terms: ***quantum, entanglement, cryptography, encryp-tion, decryption***

I. INTRODUCTION

The basic unit of information in quantum computing is 'qubit' as opposed to the classical bits. The classical bits can have a value of either 1 or 0 at a time. So given N bits, we can perform operations on only one bit state out of the 2N possible states at a time. The qubits in the quantum computing realm on the other hand can have a value of 0 or 1 or both at the same time. This is because of the special property of qubits called the superposition. Thus, given N qubits, we can perform operations on all 2N possible states at the same time. This is what makes quantum computers more powerful as compared to classical computers. We will make use of such quantum computing properties to demonstrate how quantum computers could potentially break RSA encryption and also show how they can outperform classical by comparing the two.

The problem statement is as follows: Given an integer N, find its prime factors. On a quantum computer, to factor an integer N, Shor's algorithm runs in polynomial time (the time taken is polynomial in logN, the size of the integer given as input). Specifically, it takes quantum gates of order O((logN)2 (loglogN) (logloglogN)) using fast multiplication, thus demonstrating that the integer-factorization problem can be efficiently solved on a quantum computer. The efficiency of Shor's algorithm is due to the efficiency of the quantum Fourier transform, and modular exponentiation by repeated squarings. If a quantum computer with a sufficient number of qubitscould operate without succumbing to quantum noise and other quantum-decoherencephenomena, then Shor's algorithm could be used to break public-key cryptography schemes, such as the widely used RSA scheme. RSA is based on the assump-tion that factoring large integers is computationally intractable. As far as is known, this assumption is valid for classical (non-quantum) computers; no classical algorithm is known that can factor integers in polynomial time. However, Shor's algorithm shows that factoring integers is efficient on an ideal quantum computer, so it may be feasible to defeat RSA by constructing a large quantum computer. It was also a powerful motivator for the design and construction of quantum computers, and for the study of new quantum-computer algorithms. It has also facilitated research on new cryptosystems that are secure from quantum computers, collectively called post-quantum cryptography.

In 2001, Shor's algorithm was demonstrated by a group at IBM, who factored into, using an NMR implementation of a quantum

computer with qubits[5].After IBM's implementa-tion, two independent groups implemented Shor's algorithm using photonic qubits, emphasizing that multi-qubit entangle-ment was observed when running the Shor's algorithm cir-cuits[6][7].In 2012, the factorizationwas performed with solid-state qubits[8].Also, in 2012, the factorization was achieved, setting the record for the largest integer factored with Shor's algorithm[9].

A. Aim/Statement of the Problem

The main aim is to demonstrate quantum computational efficiency by breaking RSA encryption using the Quantum Algorithm of Shors.

B. Objectives

Main objectives of our project work would include the following:

Setting up of an environment to be able to run RSA a custom web based text messaging application.

Implementation of RSA encryption/decryption.

Demonstration of the Shors Algorithm on a classical computer.

Demonstration of the Shors Algorithm on a quantum computer.

Comparison of the 2 approaches.

Solution Methods

Shor's algorithm is a quantum algorithm for finding the prime factors of an integer N(it should not be a prime/even/integer power of a prime number). For example, you want to hack into a crypto system and you have apriori knowledge of one fact concerning N(the RSA public key): that N has exactly 1 prime factorization. So, in order to find N's factors in order to crack the RSA encryption, you need to compute $f(x) = (x^r)$ mod N. Here, N is a very large number and it is hence not possible to just randomly guess its factors. We have one hint to our consideration, that is, N is one prime factorizable and r is periodic, i.e. $f(x) = f(x + r)$.Shor's algorithm tackles this problem by making use of a method called the Quantum Discrete Fourier transform(QFT) to find the period 'r'. Now, a Discrete Fourier Transform transforms a set of numbers into a set of sines and cosines. A QFT instead generates a list of the "probability amplitude" for the given list of qubit states. Since a quantum computer can 'exist' in many states simultaneously, it enables it to evaluate the periodic function f(x) at all points simultaneously.

A QFT is computed by a quantum circuit which uses:

Hadamard transform- a square matrix consisting of +1 and -1, and the rows are orthogonal to each other. The Hadamard transformation is at most times a sort of 'preprocessing step' in most of the quantum algorithms; it

maps n qubits(0 or 1) to a superposition of 2^n orthogonal states.

Quantum Gates: Time-invertible quantum circuits operat-ing on a set of qubits(units of quantum computation)

Now, for a particular 'possible' value of the period 'r', the quantum computer can exist in different states and in some way contribute to the value of 'r'. In the end, these states cancel out each other. However, only for the correct value of 'r' do the states add up along the same direction.

Shor's algorithm is probabilistic in nature and its per-formance improves with repetitions. Shor's algorithm uses quantum computing(faster) to find the prime factors of a RSA public key in order to hack into a cryptosystem.

II. LITERATURE REVIEW

In 1994, an American applied mathematician Peter Shor, working at Bell Labs in Murray Hill, New Jersey, formulated Shors algorithm[1], a quantum algorithm for integer factoriza-tion. Because Shors algorithm shows that a quantum computer, or quantum supercomputer algorithm, with a sufficient number of qubits, operating without succumbing to noise or other quantum interference phenomena, could theoretically be used to break public-key cryptography schemes such as the widely used RSA[6] scheme, its formulation in 1994 was a powerful motivator for the design and construction of quantum comput-ers, and for the study of new quantum computer algorithms. The algorithm is significant because it implies that public key cryptography might be easily broken, given a sufficiently large quantum computer. RSA, for example, uses a public key N which is the product of two large prime numbers. One way to crack RSA encryption is by factoring N, but with classical algorithms, factoring becomes increasingly time consuming as N grows large; more specifically, no classical algorithm is known that can factor in time O((log N)k) for any k. By contrast, Shors algorithm[8] can crack RSA in polynomial time. It has also been extended to attack many other public key cryptosystems. Like all quantum computer algorithms, Shors algorithm is probabilistic: it gives the correct answer with high probability, and the probability of failure can be decreased by repeating the algorithm. In 2001[4], IBM demonstrated the factorization of the number 15 (3 x 5) using a 7 qubit quantum computer. This was based on NMR(Nuclear Magnetic Resonance) implementation which is one of the ways to build a quantum machine. It makes use of the spin states of the nuclei within the molecules as Qubits - the quantum version of bits. Since then, many other groups have implemented Shors Algorithm using various other techniques like photonic qubits, where the quantum machine is based on the polarization of photons. In 2012, factorization of 21 was achieved[5]. The current largest number factored on a quan-tum device is 4088459(20172027), which was achieved using IBMs 5-qubit processor[2]. Shors Algorithm helps in finding prime factors of a given odd composite positive integer n. The problem of factoring integer n can be reduced by choosing a random integer in relatively prime to n and finding smallest positive integer P such that mP = 1 mod n[7] Shor developed the algorithm in polynomial time to solve the factorization problem. It consists of five steps. Among these steps the step 2 requires the use of quantum computer. Remaining steps of the algorithm can be performed in classical computer.

Shors algorithm is not a pure quantum algorithm, but a hybrid of classical and quantum processing. The quantum part of the algorithm determines the period of a random number selected in step 1 of the algorithm. Some of the superficial and high level questions which we are trying to

answer include questions like the number of qubits required for some n bit integer to be factorized. One paper, Beauregard, reports requiring only 2n + 3 qubits[9] 8195 qubits for a 4096-bit semiprime. Another algorithm, by Pavlidis and Gizopoulos, requires a lot more qubits, 9n + 3[10], but fewer gates must be executed. Breaking RSA on a small scale with Shors Algorithm also requires additional knowledge of some other algorithms like the Rabin Millers Primality Testing algorithm and the Euclidean algorithm to find the gcd of 2 numbers. Both these algorithms are classical and are done on a classical computer pre and post the quantum processing. These algorithms are essential to fulfil the research objectives. There are a few limitations with quantum computing and Shors algorithm. Shors algorithm is, like other quantum algorithms, probabilistic in nature. The best use of Shors Algorithm can be made when there is a powerful enough Quantum Computer with a large number of qubits. The largest quantum computer publicly available to run programs on, is the IBM-Q, which is a 53Qubit machine. Google has a 72 Qubit Quantum computer, but it isnt available for the open public.

III. SYSTEM DESIGN

The paper includes the design of 2 algorithms.
RSA Algorithm.
Shors Algorithm.

RSA Algorithm

RSA algorithmis an asymmetric cryptography algorithm which means, there should be two keys involve while communicating, i.e., public key and private key. There are simple steps to solve problems on the RSA Algorithm.
Generate 2 large random prime numbers of the specified bits.
Compute n = p*q.
Computer the Eulers Totient = (p-1) * (q-1)
Compute the public key 'e'. Choose'e', such that 'e' should be co-prime. Co-prime means it should not multiply by factors of Eulers Totientand also not divide by Eulers Totient.
Compute the private key (d). The condition is given as, gcd(phi, e) = phi(x) + ey = 1.
Encryption and decryption. Encryption is given as, $C = m^e$ mod n. Decryption is given as, $m = c^d$ mod n.
Here are some points on RSA and how quantum can play a vital role in breaking RSA:
RSA is a well known encryption/decryption algorithm. It is used in various applications such as emails, chat
applications, VPNs etc. RSA relies on the fact that it is very hard to factorize a number, and harder to factorize into its prime factors.
The only thing we know is the Public Key, and using just this we cannot decrypt the cipher.
Finding the prime factors of the modulus used in RSA is a difficult mathematical problem. Even though there are algorithms available, it takes infinite time to find the prime factors.
Once prime factors are known, we can build the private key using the known public key and the totient value. But since prime factors are very hard to find, we cannot break RSA unless we have some superpower with us.
If the prime factors p and q of the modulus N are somehow found out efficiently, then we can surely break RSA.
Quantum computing and Shors Algorithm help us in doing just this.
We can do the pre and post Shors part on a classical computer, but the Shors Algorithm should run on a quantum machine.
The concept of superposition and QFT (Quantum Fourier Transform) play a vital role here.
Shor's Algorithm
The following steps are involved in Shors Algorithm:
Step 1: Pick a random number a such that 1 < a < N and gcd(a, N) = 1.
Step 2: Compute the period r of (a mod N).
Step 3: Check the following conditions:
is 'r' even?
Is $(a^r=2) + 1 \text{mod} N! = 0$.
Step 4: If both conditions are true, then we can derive that
$p = \gcd((a^r=2) + 1; N)$
&
$q = \gcd((a^r=2) \quad 1; N)$
Step 5: else go to step 1 and start over with a different 'a'.
Here, the crux of this algorithm lies in step 2 of period finding. Finding the period of (a mod N) on a classical computer would take a very long time, longer than the use time of the keys used in RSA. But we know that quantum computers have the ability to work on 2^n states simultaneously, Shors Algorithm makes use of this very fact to brilliantly find the period. Thus Shors Algorithm is a hybrid algorithm which involves both classical and quantum parts. All classical parts can surely be done on a classical computer and the period finding can be done on a quantum computer.
Shor's algorithm uses a method called the Quantum Discrete Fourier transform(QFT) to find the period 'r'. Now, a Discrete Fourier Transform transforms a set of numbers into a set of sines and cosines. A QFT instead generates a list of the "probability amplitude" for the given list of qubit states.
A Quantum computer fits this task perfectly for the following reason: A quantum computer can 'exist' in many states simultaneously, which enables it to evaluate the periodic function f(x) at all points simultaneously.

A QFT is computed by a quantum circuit which uses:

- Hadamard Transform - A square matrix consisting of +1 and -1, and the rows are orthogonal to each other. The Hadamard transformation is at most times a sort of 'preprocessing step' in most of the quantum algorithms; it maps n qubits(0 or 1) to a superposition of 2^n orthogonal states.
- Quantum Gate - Time-invertible quantum circuits oper-ating on a set of qubits(units of quantum computation).

Now, for a particular 'possible' value of the period 'r', the quantum computer can exist in different states and in some way contribute to the value of 'r'. In the end, these states cancel out each other. However, only for the correct value of 'r' do the states add up along the same direction. Shor's

algorithm is probabilistic in nature and its performance improves with repetitions.
All that Shors Algorithm does is that it tries to estimate the phases of p and q, and gives the result if the phases of both p and q turns out to be 0 at the point N. The below given quantum circuit is used to perform phase estimations.

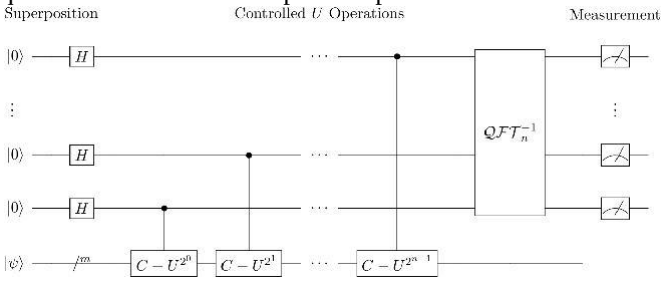

Fig. 1. Quantum Circuit performing phase estimations.

The step 2 of Shors Algorithm, which is Period finding, can be done only on a quantum machine. We have algorithms which can find the period even on a classical machine, but this would take years of computation time to get the period. We will discuss both the algorithms below.
Period finding on a Classical machine: Suppose the number to be factorized is N=21, and the random number x we chose is 2, then we would have to go through each power until we find that the values repeat. In classical computers we have to go through all powers to find the period of the function f = X^amodN. This is done by the following steps:
2 mod 21 = 2 4 mod 21 = 4 8 mod 21 = 8 16 mod 21 = 16 32 mod 21 = 11 64 mod 21 = 1
128 mod 21 = 2 (values repeat here). 256 mod 21 = 4
Thus, the period is 'r'=6.
If we take a bit larger number, say 221 and the random number=5, then the number of steps in this case would be very huge, as given below:
5 mod 221 = 5
25 mod 221 = 25
125 mod 221 = 125 625 mod 221 = 183 3125 mod 221 = 31
15625 mod 221 = 155 78125 mod 221 = 112 .....
152587890625 mod 221 = 1
Thus we can infer that classical computers are no good for breaking RSA. So we need some superpower that can do all these computations in a single shot and give us the results. This is where quantum computing comes to play.
Period finding on a Quantum machine: The algorithm for period finding with a Quantum machine is given by the following steps:
Step 1: Choose a random integer 'x such that 1 < x < N and gcd(x,N)=1.
Step 2: Find 'q such that $q = 2^l$ and $N^2 <= q < 2N^2$.
Step 3: Initialize 2 quantum registers r1 and r2 of length l each. Initialize both the quantum registers to j0 > state.
Step 4: Initialize r1 with a superposition of all qubits (H).
Step 5: Initialize r2 with a superposition of X^amodN(U_a).
Step 6: Apply QFT on r1 (entangled with r2).
Step 7: Measure r1 - gives some superposed state j > of some base state jk > such that k is the power of x mod N. Let the result be 'c'.
Step 8: Apply continued fractions of c/q until we get some r such that c/q=d/r(even approximation will do).
Step 9: Check neighborhood of r to confirm the period, if trivial - repeat.
In this period finding algorithm,it can be seen that there is the usage of a special function which is QFT. We will compare DFT with QFT first, so that it is easy to understand QFTs.

The discrete Fourier transform acts on a vector $(x_0; :::; x_{N-1})$ and maps it to the vector $(y_0; :::; y_{N-1})$ according to the formula

$$x_k \rightarrow y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk} \quad \text{where, } \omega_N^{jk} = e^{2\pi ijk/N}.$$

QFT does the following transformation when applied on jx >:
$$|x\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle \omega^{xa} \text{ where } \omega = e^{2\pi i/q}.$$
We will try to break RSA with the same modulus N=21 but this time, we will demonstrate Shors algorithm by using Quantum machine
Pre-Quantum Steps - On a Classical Machine
Step 1: Choose a random integer 'x such that 1 < x < N and gcd(x,N)=1.
x = 11 > 1 < 11 < 21 and gcd(11,21)=1.
Step 2: Find q such that $q = 2^l$ and $N^2 <= q < 2N^2$ q = 512 = 2^9
l = 9
441 <= 512 < 882
Quantum Steps - On a Quantum Machine
Step 3: Initialize 2 quantum registers 'r1 and 'r2 of length 'l each. Initialize both the quantum registers to j0 > state. Initializing r1 and r2 with 9 qubits each.
Step 4: Initialize r1 with a superposition of all qubits (H).

We get $|j_0\rangle = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle |0\rangle$

$|j_0\rangle$ is now superposed with state of j0 >; j1 >; :::; j511 >

Step 5: Initialize r2 with a superposition of X^amodN(U_a)

We get $|j_1\rangle = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle |11^a (\text{mod} 21)\rangle$

Step 6: Apply QFT on r1

On applying QFT to $|j_0\rangle$, we get this transformation

$$|j_0\rangle \rightarrow \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle \omega^{0a}$$

But r1 is entangled with r2, i.e with

$$|j_1\rangle \rightarrow \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle |11^a (\text{mod} 21)\rangle$$

So applying on QFT to this entanglement, we get the following superposition:

$$|j_{new}\rangle \rightarrow \frac{1}{\sqrt{512}} \sum_{a=0}^{511} \sum_{c=0}^{511} e^{2\pi iac/512} |c\rangle |11^a (\text{mod} 21)\rangle$$

Step 7: Measure r1 gives some superposed state $|j_s\rangle$ of some base state jk > such that k is a power of x mod

N. Let result be 'c'
Assuming the superposed qubits collapse to jc >= 427
Step 8: Apply continued fractions of c/q until we get some 'r such that c/q=d/r(or approximately equal).
$d_0 = a_0$; $d_1 = a_0a_1 + 1$; $d_n = a_nd_{n\ 1} + d_{n\ 2}$
$r_0 = 1$; $r_1 = 1$; $r_n = a_nr_{n\ 1} + r_{n\ 2}$ On solving, we get,
c=q = 427=512 < > d=r = 11=6
Step 9: Check neighborhood of 'r to confirm the period, if trivial - repeat
– Checking multiples of r=6.
– For mr=6*1, check if x^xmodN = x^{x+mr}modN, if so return 'mr' as period.
x^xmodN = $11^1$1mod21 = 2
x^{x+mr}modN = 11^{11+6}mod21 = 2
Since 2=2, returning mr=6*1=6 as the period
– Therefore, PERIOD = 6.
Post-Quantum Steps - On a Classical Machine
Step 10: Check if 'r is even, if not - go to Step 1 and repeat.
r=6(even)
Step 11: Check if $x^{r=2}$ + 1(modN)! = 0, if not go to step1 and repeat.
11^3 + 1(mod21) = 1331 + 1(mod21) = 1332(mod21) = 9!=0
Step 12: Prime factors are:
p = gcd($x^{r=2}$ + 1; N) = gcd(11^3 + 1; 21) = gcd(1332; 21) = 3
q = gcd($x^{r=2}$ 1; N) = gcd(11^3 1; 21) = gcd(1330; 21) = 7
Now that we have successfully found the values of p and q, it is very easy to decipher the RSA encrypted messages.

IV. CONCLUSION

We successfully built Shors algorithm and tested it on both the classical as well as the quantum machines. As it is evident from the System Design part of this paper, using a classical machine could takes years to find the prime factors of a number of length sizing up to 2048 bits and beyond. Nevertheless, with quantum computers, we perform all these operations on all the possible states at the same time. This helps us achieve tremendous speed up in terms of finding the prime factors even of numbers as large as 2048 bits and beyond. The only downside with quantum computing is that we dont have quantum machines of such a high capacity. When time comes, and when we have quantum machines with huge number of qubits, we will be able to easily break RSA and the entire security system of chatting applications, VPNs, emails, etc would fail.

Is RSA dead then?
As of now, RSA uses 2048 bit keys majorly. But the largest quantum machine we have is of around 100 qubits.
To be able to break RSA which uses such long keys, we have a long road ahead. The latest rigetti forest (Aspen-7) provides 28 qubit quantum machine.(This is for us to work on our project).
Thus, we can still rest assured that our RSA encryption is safe, but there is a tremendous amount of research and investment being done on quantum computing and with this, there is a nearing possibility that we might reach a day when it is possible to break RSA.

REFERENCES

[1] A fast quantum mechanical algorithm for database search: Lov K. Grover (3C-404A, Bell Labs, 600 Mountain Avenue, Murray Hill, NJ, 07974).
[2] Quantum Searching: A survey of Grovers Algorithm and its Descendants: Logan Mayfield.
[3] Implementing Grovers Algorithm on the IBM Quantum Computers: Aamir Mandviwalla*, Keita Ohshiro and Bo Ji
[4] 4-qubit Grover's algorithm implemented for the ibmqx5 architecture: VERA BLOMKVIST KARLSSON, PHILIP STRMBERG: [STOCK-HOLM, SWEDEN 2018].
[5] Vandersypen, Lieven M. K.; Steffen, Matthias; Breyta, Gregory; Yannoni, Costantino S.; Sherwood, Mark H. &Chuang, Isaac L.(2001),"Experimental realization of Shor's quantum factoring algo-rithm using nuclear magnetic resonance"
[6] Lu, Chao-Yang; Browne, Daniel E.; Yang, Tao & Pan, Jian-Wei (2007),"Demonstration of a Compiled Version of Shor's Quantum Fac-toring Algorithm Using Photonic Qubits".
[7] Lanyon, B. P.; Weinhold, T. J.; Langford, N. K.; Barbieri, M.; James, D. F. V.; Gilchrist, A. & White, A. G. (2007),"Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entangle-ment".
[8] Lucero, Erik; Barends, Rami; Chen, Yu; Kelly, Julian; Mariantoni, Matteo; Megrant, Anthony; O'Malley, Peter; Sank, Daniel; Vainsencher, Amit; Wenner, James; White, Ted; Yin, Yi; Cleland, Andrew N.; Martinis, John M. (2012). "Computing prime factors with a Josephson phase qubit quantum processor".
[9] Martn-Lpez, Enrique; Martn-Lpez, Enrique; Laing, Anthony; Lawson, Thomas; Alvarez, Roberto; Zhou, Xiao-Qi; O'Brien, Jeremy L. (12 October 2012). "Experimental realization of Shor's quantum factoring algorithm using qubit recycling".
[10] Bloch, Felix (Oct 1946). "Nuclear induction".Phys. Rev.70(7-8): 460-474. Bibcode:1946PhRv...70..460B.doi:10.1103/physrev.70.460.
[11] Nielsen, Michael A.;Chuang, Isaac L.(2004).Quantum Computation and Quantum Information. Cambridge University Press.ISBN978-0-521-63503-5.
[12] Quantiki : http://www.quantiki.org/wiki/Bloch sphere
[13] Poincar, Henri(1892).Thorie mathmatique de la lumire II. G. CarrFeynman, Richard; Vernon, Frank; Hellwarth, Robert (January 1957)."Geometrical Representation of the Schrdinger Equation for Solving Maser Problems".Journal of Applied Physics.28(1): 4952.Bibcode:1957JAP....28...49F.doi:10.1063/1.1722572.
[14] Milonni, Peter W.; Eberly, Joseph (1988).Lasers.

New York: Wiley. p.340.ISBN978-0471627319.
[15] Appleby, D.M. (2007). "Symmetric informationally complete measurements of arbitrary rank".Optics and Spectroscopy.103(3): 416428.arXiv:quant-ph/0611260. doi:10.1134/S0030400X07090111.
[16] Aharonov, Dorit (2003-01-09). "A Simple Proof that Toffoli and Hadamard are Quantum Universal".arXiv:quant-ph/0301040
[17] "Monroe Conference"(PDF).online.kitp.ucsb.edu.
[18] "Demonstration of a small programmable quantum computer with atomic qubits"(PDF). Retrieved2019-02-10.
[19] Jones, Jonathan A. (2003). "Robust Ising gates for practical quantum computation".Physical Review A.67.arXiv:quant-ph/0209049.doi:10.1103/PhysRevA.67.012317.
[20] Deutsch, David (September 8, 1989), "Quantum computational networks", Proc. R. Soc. Lond. A,425(1989): 73 90,Bibcode:1989RSPSA.425...73D,doi:10.1098/rspa.1989.0099
[21] Bausch, Johannes; Piddock, Stephen (2017), "The Com-plexity of Translationally-Invariant Low-Dimensional Spin Lattices in 3D",Journal of Mathematical Physics,58(11): 111901,arXiv:1702.08830,doi:10.1063/1.5011338
[22] Raz, Ran(2002). "On the complexity of matrix product".Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing: 144.doi:10.1145/509907.509932. ISBN1581134959.
[23] Defining adjoined operators in Microsof Q#.
[24] Q# Online manual: Measurement.
[25] Juan Yin, Yuan Cao, Yu-Huai Li, et.c."Satellite-based entanglement distribution over 1200 kilometers".Quantum Optics.
[26] Billings, Lee."China Shatters "Spooky Action at a Distance" Record, Preps for Quantum Internet", Scientific American.
[27] Popkin, Gabriel (15 June 2017)."China's quantum satellite achieves 'spooky action' at record distance".Science - AAAS.
[28] QCL 0.6.4 source code, the file "lib/examples.qcl".
[29] Quantum Programming in QCL (PDF).
[30] Phys. Rev. A5234573467 (1995), doi:10.1103/PhysRevA.52.3457; e-printarXiv:quant-ph/9503016
[31] R. P. Feynman, "Quantum mechanical computers",Optics News, Febru-ary 1985,11, p. 11; reprinted inFoundations of Physics16(6) 507531.

