Turk. J. Math. Comput. Sci.
11(2)(2019) 78–83

http://dergipark.gov.tr/tjmcs
http://tjmcs.matder.org.tr

MATDER

Fundamental Structure of Shor's Quantum Algorithm for Factoring Integers

Turgut Hanoymak[1,*], Akram Chehrazi[2]

[1]*Department of Mathematics, Van Yuzuncu Yil University, Van, Turkey.*
[2]*Department of Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran.*

Received: 26-05-2019 • Accepted: 18-06-2019

Abstract. One of the most well known mathematically hard problems in number theory is the integer factorization problem, roughly stated that decomposition of a composite number into its prime factors. In modern cryptography, RSA encryption algorithm whose security is based on integer factorization problem is highly practical, widespread and up to date no classical algorithm having polynomial running time for the factorization of large numbers is known. In 1994, Peter Shor proposed an efficient algorithm on quantum computer. In this paper, we mention about the fundamentals of Shor's quantum algorithm illustrating a concrete example.

2010 AMS Classification: 68Q12, 81P68, 81P94, 94A60.

Keywords: Factorization, measurement, superposition principle, qubit.

1. Introduction

One of the most monumental discovery in quantum computation is probably Shor's algorithm [15] which ensures incomparable speedup over the fastest classical algorithms for factoring integers indicating that decomposition of a composite number into its prime factors. Since factorization of huge numbers is computationally intractable, the security of commonly known and widely used asymmetric key encryption algorithm, RSA, proposed by [14] after the invention of public key cryptography [4] depends on this fact and up to now, there have not been found any algorithms executing on classical computer which makes Shor's quantum algorithm have great importance and one of the milestones both in cryptography and computer science.

Shor's algorithm runs in polynomial time-$O((logN)^2(\log\log N)(log\,log\,logN))$- to factor an integer N on a quantum computer which is a huge step compared to the classical counterpart where the most efficient known algorithm called number field sieve [12] runs in $O(e^{1.9(log\,N)^{1/3}(log\,log\,N)^{2/3}})$. After having been introduced Shor's algorithm, many researchers tried either to implement this algorithm on a quantum computer or to modify it for getting better results on integers having large digits or to find another kind of algorithms under the light of this awesome algorithm. In 2001, Vandersypen et al., [19] was able to succeed to factor 15 into its prime factors 3 and 5 on a quantum computer with 7 qubits. Martin-Lopez et al., [9] factorized 21 and the same year 143 was factorized by using adiabatic quantum computation [20]. In 2014, Nikesh S. Dattani and Nathaniel Bryans [3] were able to factor 56153 with only 4 qubits and they showed a 3-qubit factorization of 175 which would be the first quantum factorization of a triprime. Soham Pal et al., [10] factored 551 using a 3-qubit quantum adiabatic processor in 2016. Zhaokai Li et al., [8] proposed a scheme

*Corresponding Author
Email addresses: turguthanoymak@gmail.com (T. Hanoymak), achehrazi95@gmail.com (A. Chehrazi)

for adiabatic quantum computation by using the intrinsic Hamiltonian of a realistic spin system and factored 291311 in 2017. Shuxian Jiang et al., [6] presented how to factorize 15, 143, 59989, and 376289 using 4, 12, 59, and 94 logical qubits, respectively in 2018. Avinash Dash et al., [2] experimentally demonstrated the factorization of two bi-primes, 4088459 and 966887 using IBM's 5- and 16-qubit quantum processors, hence making those the largest numbers that has been factorized on a quantum device and they also factorized the number 175 using only two qubits in 2018.

In this paper, we explain the fundamentals of Shor's quantum algorithm with underlying mathematical aspects and finally we give the factorization of 15.

2. Preliminaries

The most important characteristics of quantum computing according to classical counterpart is the superposition principle meaning that a quantum system is in all of its possible states at the same time until the system is observed while a classical system has only one value at any given time. We emphsize that after the observation of the system, it collapses into a definite classical state with probability. Because of this, a quantum system is probabilistic. While in the classic systems, information is represented in terms of classical bits 0 and 1, quantum computing works on quantum bits which are called qubits.

The mathematical structure of quantum computing is considered in Hilbert space H which is a generalization of Euclidean space. Hilbert space is a vector space with an inner product $\langle u, w\rangle$ such that the *norm* defined by $\|u\| = \sqrt{\langle u, u\rangle}$ turns H into a complete metric space. Every qubit is represented by a vector in a complex space. Let v be the vector in a n-dimensional vector space, we denote that by $|v\rangle = \{v_1, v_2, ..., v_n\} \in \mathbb{C}^n$ and we call it *ket* of v. The *bra* of this vector is denoted by $\langle v|$ such that $\langle v| = \overline{v^T}$ where $\overline{v^T} = \{\overline{v_1}, ..., \overline{v_n}\}$ and $\overline{v}$ is the complex conjugate of v. The inner product of two vectors u and v is given by $\langle u|v\rangle = \overline{u^T}v = \overline{u_1}.v_1 + ... + \overline{u_n}.v_n$ which is a scalar. The norm of a vector $|v\rangle$ in a Hilbert space is defined as $\| \ |v\rangle \ \| = \sqrt{\langle v|v\rangle}$ which is the distance from the origin to the vector $|v\rangle$. A way of obtaining larger vector spaces is to use *tensor product* operation which has great importance in quantum computation and defined for two vectors $|v\rangle = \{v_1, ..., v_n\}$ and $|u\rangle = \{u_1, ..., u_m\}$ as follows:

$$|u\rangle \otimes |v\rangle = |uv\rangle = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \otimes \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} u_1.v_1 \\ \vdots \\ u_1.v_n \\ u_2.v_1 \\ \vdots \\ u_2.v_n \\ \vdots \\ u_m.v_n \end{bmatrix}.$$

In quantum computing, vector spaces describe quantum states and when they are tensored, linear combinations of all vectors in the two vector spaces are produced. While in classical systems the information is represented in terms of classical bits 0 and 1, in quantum computing, qubits are used which can be in a superposition between 0 and 1. We present the quantum state of a qubit as a 2-dimentional orthonormal basis vectors

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and the superposition $|\varphi\rangle$ of a qubit is represented as a linear combination of $|0\rangle$ and $|1\rangle$ such that

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α, β are the complex numbers and the amplitudes of measuring $|0\rangle$ and $|1\rangle$, respectively. Because of the normalization condition for quantum computation, we assume that $|\varphi\rangle$ is a unit vector. Then, we have

$$1 = \langle\varphi|\varphi\rangle = (\overline{a_0}\langle 0| + \overline{a_1}\langle 1|) \cdot (a_0|0\rangle + a_1|1\rangle) = |a_0|^2\langle 0|0\rangle + |a_1|^2\langle 1|1\rangle + \overline{a_1}a_0\langle 1|0\rangle + \overline{a_0}a_1\langle 0|1\rangle = |a_0|^2 + |a_1|^2.$$

It means that the probability of the qubit being in the state $|0\rangle$ is $|a_0|^2$, and the probability that the qubit will be measured as $|1\rangle$ is $|a_1|^2$.

For more information about quantum algorithms and quantum computation, the reader is referred to [17] and [7].

3. RSA Encryption Scheme and Factorization Problem

RSA is a public key encryption scheme whose security is based on the hardness of the factorization of large integers [14]. The key generation algorithm produces a large composite number $N = p \cdot q$ where p and q are large odd primes, a public key e and private key d such that $e \cdot d = 1 \mod \phi(N)$ is satisfied, where $\phi(N)$ is Euler phi function of N. The encryption of a message m from $\mathbb{Z}_N^*$ is an element of $\mathbb{Z}_N^*$, namely $c = m^e \mod N$ where $\mathbb{Z}_N^*$ is the multiplicative group of $\mathbb{Z}_N$ defined as $\mathbb{Z}_N^* = \{ a \in \mathbb{Z}_N \mid \gcd(a, n) = 1 \}$. One finds m using the secret key d by computing $m = c^d \mod N$.

We state **RSA Problem** as follows:

Let $N = p \cdot q$ where p and q are large odd prime numbers. Let e be an integer relatively prime to $\phi(N)$. The RSA problem states that for a given $y \in \mathbb{Z}_N^*$, compute the e-th root of y, namely x, such that

$$y = x^e \mod N.$$

The security of RSA encryption scheme is based on the intractability of the integer factorization problem which is stated as the decomposition of a composite number into its prime factors which is defined as follows:

Integer Factorization Problem: Given a positive integer N, find its prime factorization; that is, $N = p_1^{k_1} p_2^{k_2} p_3^{k_3} ... p_s^{k_s}$ where p_i's are distinct primes and $k_i > 0$. Since the security of RSA is based on the IFP, many factoring algorithms such as Pollard's rho [1], Pollard's $p - 1$ [11], elliptic curve factoring [16], quadratic sieve algorithm [5] have been proposed and after a while improved however, up to date no classical algorithm is known for this problem which runs in polynomial time.

If the factorization of N is known, then the RSA problem can be easily solved; an adversary could calculate $\phi(N) = \phi(p \cdot q) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1) = p \cdot q - p - q + 1 = N - (p+q) + 1$ and could easily find the secret key d from $e \cdot d = 1 \mod \phi(N)$, hence is able to decrypt any ciphertext. It means that RSA problem is polynomially reduced to factoring. It is conjectured that there is no effective way except factorization to find the e-th roots modulo N. We note that breaking a cryptographic scheme is not necessarily equivalent to solving the underlying mathematically hard problems. Rabin [13] proposed an encryption function that could be proved to be invertible only by someone who could factor N, i.e., if the scheme is broken and hence the plaintext is obtained, then N is factorized in a reasonable amount of time.

4. Shor's Quantum Algorithm

Shor's algorithm [15] is a quantum algorithm for factoring integers which has a polynomial running time on a quantum computer. Because of this, it has a significant breakthrough among factoring algorithms. In this section, we explain how to factor an odd number N by using Shor's algorithm. The algorithm consists of two parts. In classical part, we mention about a reduction method such that integer factorization problem reduces to the problem of finding order r modulo N whereas in quantum counterpart, we show how to find the order r in polynomial time. We want to find a non-trivial factor of N. Because of this, firstly, we select a random integer x such that $2 < x < N$. If the greatest common divisor of x and N, $gcd(x, N)$, is not 1 we succeed, otherwise, we try for another x until $gcd(x, N) = 1$. Then, we compute the order r of x modulo N, i.e., we find the smallest non-negative integer r satisfying $x^r \equiv 1 \mod N$.

$$x^r \equiv 1 \mod N \Leftrightarrow x^r - 1 \equiv 0 \mod N \Leftrightarrow \left(x^{\frac{r}{2}} + 1\right).\left(x^{\frac{r}{2}} - 1\right) \equiv 0 \mod N.$$

Here, r should be even otherwise we choose another random integer x. We assume that $\left(x^{\frac{r}{2}} + 1\right) \not\equiv 0 \mod N$, that means $gcd(x^{\frac{r}{2}} + 1, N) \neq N$. On the other hand, we also have $gcd(x^{\frac{r}{2}} + 1, N) \neq 1$ otherwise by the Euclidean algorithm, there exist $a, b \in \mathbb{Z}$ such that $a \cdot \left(x^{\frac{r}{2}} + 1\right) + b \cdot N = 1$. Hence, we have

$$a \cdot \left(x^{\frac{r}{2}} + 1\right) \cdot \left(x^{\frac{r}{2}} - 1\right) + b \cdot N \cdot \left(x^{\frac{r}{2}} - 1\right) = x^{\frac{r}{2}} - 1 \equiv 0 \mod N$$

yielding a contradiction with r of being the order. Thus, we decide that $gcd(x^{\frac{r}{2}} + 1, N)$ divides N which is a non-trivial factor [18]. Therefore, we conclude that if we find the order r of an non-negative integer less than N, we can able to find a non-trivial factor of N which means factoring problem polynomial time reduces to finding the order modulo N. This is the classical part of the algorithm.

Example 4.1. Let $N = 4183$ be the number of which we want to find a non-trivial factor. If we know the order of an integer which is relatively prime to 4183, we can easily factorize it by using the first classical part of Shor's algorithm. For this reason, we select $x = 7$ such that $gcd(7, 4183) = 1$, then we calculate the order r of 7 modulo 4183 such that

$$7^{2024} \equiv 1 \mod 4183.$$

Finally, we obtain

$$gcd(7^{\frac{2024}{2}} + 1, 4183) = gcd(7^{1012} + 1, 4183) = 89$$

and we have $N = 4183 = 47 \cdot 89$

In this part, we briefly summarize the steps especially the *quantum Fourier transform (QFT)* which is the main step for recovering the order r with some probability using quantum computing to emphasize the importance of quantum computers because of unbelievable speed up compared with the classical counterparts.

We note that finding the order r of x modulo N is same as finding the period of function $f(i) \equiv x^i \mod N$ such that

$$f(i + jr) = x^{i+jr} \mod N \equiv x^i.(x^r)^j \equiv x^i \mod N = f(i)$$

where $j = 1, 2, 3, ...$

We consider the initial state to be $|\bar{0}\rangle \otimes |\bar{0}\rangle$ such that $|\bar{0}\rangle = |0, 0, 0, ..., \rangle$, and the length of $|\bar{0}\rangle$ is large enough to represent a big integer $M > N$. By using *Hadamard operator* to write $|\bar{0}\rangle$ as a *super position* of $|i\rangle$, we obtain

$$|\varphi_0\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle \otimes |\bar{0}\rangle.$$

Then, we subtitute the function of $f(i)$ in the second register of the state as

$$|\varphi_1\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle \otimes |x^i \mod N\rangle.$$

After measuring the second register, the first register reduces into

$$|\varphi\rangle = \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |i_0 + jr\rangle$$

where $f(i_0) = f(i_{0+jr}) = x^{i_0} \mod N$ and $A = \frac{M}{r}$ such that A is the number of periods. For finding r, we apply the discrete Fourier transform as follows:

$$U|K\rangle = \frac{1}{\sqrt{M}} \sum_{L=0}^{M-1} e^{\frac{2\pi iKL}{M}} |L\rangle.$$

This step is performed by the quantum computer and after the discrete Fourier transform over $|\varphi >$, we have

$$|\varphi'\rangle = \frac{1}{\sqrt{A}} \frac{1}{\sqrt{M}} \sum_{j=0}^{A-1} \sum_{K=0}^{M-1} e^{\frac{2\pi iK(i_0+jr)}{M}} |K\rangle.$$

After measuring the first register, we have K with probability

$$p(K) = \frac{1}{MA} |\sum_{j=0}^{A-1} e^{\frac{2\pi iK(i_0+jr)}{M}}|^2$$

with the number of period in M, $A \simeq \frac{M}{r}$. We try to find an acceptable K with high probability and this K satisfies $|\frac{K}{M} - \frac{m}{r}| < \frac{1}{2M}$. We note that since r is the order of x in mod N it means that r must be in $[1, N]$. Therefore, If we select M as $M > N^2$, there is only one fraction that is obtained by simplying $\frac{K}{M}$ such that the denominator of the fraction is less than N. Finally, we simply $\frac{K}{M}$ to find the order r by using continuous fractions

$$\frac{K}{M} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + ... \frac{1}{a_k}}}}$$

where $a_0 = 0$, since $K < M$.

We omit the intermediary calculations and refer the reader to [18] for details.

5. A Concrete Example: How to Factor 15 by Using Shor's Algorithm

In this section, we show how to factor $N = 15$ using Shor's algorithm on a quantum computer. First, we select a random integer $x = 7$ which is relatively prime to 15. We try to find the order of 7 modulo 15 such that $7^r \equiv 1 \bmod 15$ holds.

We consider $|\overline{0}\rangle \otimes |\overline{0}\rangle$ as an initial state and write the first register in a superposition as follows:

$$|\varphi_0\rangle = \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle \otimes |\overline{0}\rangle = \frac{1}{\sqrt{M}} (|0\rangle + |1\rangle + ... + |M-1\rangle) \otimes |\overline{0}\rangle.$$

Applying the function $f(i) = 7^i \bmod 15$, we have

$$\begin{aligned}
|\varphi_1\rangle &= \frac{1}{\sqrt{M}} \sum_{i=0}^{M-1} |i\rangle \otimes |7^i \bmod 15\rangle = \frac{1}{\sqrt{M}} \left(|0\rangle \otimes |1\rangle + |1\rangle \otimes |7^1 \bmod 15\rangle + |2\rangle \otimes |7^2 \bmod 15\rangle + ...\right) \\
&= \frac{1}{\sqrt{M}} (|0\rangle \otimes |1\rangle + |1\rangle \otimes |7\rangle + |2\rangle \otimes |4\rangle + |3\rangle \otimes |13\rangle + |4\rangle \otimes |1\rangle + |5\rangle \otimes |7\rangle + |6\rangle \otimes |4\rangle + |7\rangle \otimes |13\rangle + ...) \\
&= \frac{1}{\sqrt{M}} \{(|0\rangle + |4\rangle + |8\rangle + ...)\,|1\rangle + (|1\rangle + |5\rangle + |9\rangle + ...)\,|7\rangle + (|2\rangle + |6\rangle + |10\rangle + ...)\,|4\rangle + (|3\rangle + |7\rangle|11\rangle + ...)|13\rangle\}.
\end{aligned}$$

We realize that the second register is in state $|1\rangle$, $|7\rangle$, $|4\rangle$ or $|13\rangle$. After measurement, we assume that the system collapses to the state $|4\rangle$. In this case, the first register reduces into $|\varphi\rangle = \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |i_0 + jr\rangle$ such that $A = \frac{M}{r}$ where $i_0 = 2$ and $r = 4$. In other words, $|\varphi\rangle = \sqrt{\frac{4}{M}} \{|2\rangle + |6\rangle + |10\rangle + |14\rangle + ...\}$. Now, we apply the *QFT* and get the result as follows:

$$|\varphi'\rangle = \sqrt{\frac{4}{M}} \frac{1}{\sqrt{M}} \sum_{j=0}^{A-1} \sum_{K=0}^{M-1} e^{\frac{2\pi i K(i_0+jr)}{M}} |K\rangle = \sum_{K=0}^{M-1} \alpha_K |K\rangle.$$

We notice that the sequence $\{2, 6, 10, 14, ...\}$ is of the form $2(2k + 1)$. Now, We randomly select $M = 2048$ and for $K = 512$, we calculate $\alpha_K = \frac{-1}{2}$. Hence, we have

$$|\varphi'\rangle = \frac{1}{2} \{|0\rangle - |512\rangle + |1024\rangle - |1536\rangle\}.$$

If we apply measurement to this state, we assume that the output is $|1536\rangle$. Therefore, $\frac{K}{M} = \frac{1536}{2048} = \frac{3}{4} \Rightarrow r = 4$. So,

$$\gcd(x^{\frac{r}{2}} + 1, N) = gcd(7^{\frac{4}{2}} + 1, 15) = gcd(50, 15) = 5$$

is a factor of 15.

6. Conclusion

In this paper, we briefly mention about quantum computing and the advantages of computations on quantum computer compared with the classical counterparts. Then, we give the basics of Shor's algorithm running time of polynomial for factoring integers which is known to be one of the hardest problem in mathematics. Finally, we show the steps for factoring 15 using Shor's algorithm on quantum computer as a concrete example.

References

[1] Bach, E., *Toward a theory of Pollard's rho method*, Information and Computation, **90**(1991), 139–155. 3

[2] Dash, A., Sarmah, D., Behera, B.K., Panigrahi, P.K., *Exact search algorithm to factorize large biprimes and a triprime on IBM quantum computer*, 2018. 1

[3] Dattani, N.S., Bryans, N., *Quantum factorization of 56153 with only 4 qubits*. arXiv:1411.6758 [quant-ph], 2014. 1

[4] Diffie, W., Hellman, M., *New Directions in Cryptography*, IEEE Transactions on Information Theory, **22(6)**(1976), 644–654. 1

[5] Gerver, J., *Factoring large numbers with a quadratic sieve*, Mathematics of Computation, **41**(1983), 287–294. 3

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

[7] Kute S., Desai C.G., *Quantum Cryptography: A Review*, Indian Jour. of Scien. and Techn., **10(3)**(2017). 2

[8] Li, Z., Dattani, N.S., Chen, X., Liu, X., Wang, H., Tanburn, R., Chen, H., Peng, X., Du, J., *High-fidelity adiabatic quantum computation using the intrinsic Hamiltonian of a spin system: Application to the experimental factorization of 291311*, (2017). 1

[9] Martin-Lopez, E., Laing, A., Lawson, T., Alvarez, R., Zhou, X., O'Brien, J.L., *Experimental realization of Shor's quantum factoring algorithm using qubit recycling*, Nature Photonics, 6 **11**(2012), 773–776. 1

[10] Pal, S., Moitra, S., Anjusha, V.S., Kumar, A., Mahesh, T.S., *Hybrid scheme for factorization: Factoring 551 using a 3-qubit NMR quantum adiabatic processor*, (2016). 1

[11] Pollard, J.M., *Theorems on factorization and primality testing*, Proceedings of the Cambridge Philosophical Society, **76**(1974), 521–528. 3

[12] Pomerance, C., *A tale of two sieves*, The Notices of the Amer. Math. Soc., **43**(1996), 1473–1485. 1

[13] Rabin, M., *Digitalized signatures and public-key functions as intractable as factorization*, MIT Laboratory for Computer Science, January 1979. 3

[14] Rivest, R., Shamir, A., Adleman, L., *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM 21, **2**(1978), 120–126. 1, 3

[15] Shor, P.W., *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc., 1994. 1, 4

[16] Silverman, R.D., Wagstaff JR., S.S., *A practical analysis of the elliptic curve factoring algorithm*, Mathematics of Computation, **61**(1993), 445–462. 3

[17] Strubell, E., An Introduction to Quantum Algorithms, COS498, Chawathe, 2011. 2

[18] Valle, C., Shor's Algorithm and Grover's Algorithm in Quantum Computing, Master's thesis, University of Kansas, 2011. 4, 4

[19] Vandersypen, L.M., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H. , Chuang, I.L., *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature, **414**(2001), 883–887. 1

[20] Xu, N., Zhu, J., Lu, D., Zhou, X., Peng, X., Du, J., *Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system*, Physical Review Letters, 108 **13**(2012). 1