ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **10** (2021), no.2, 729–742
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.10.2.5

DECRYPTING RSA ENCRYPTION USING QUANTUM ALGORITHM

Nitin Asthana and S. Shridevi[1]

ABSTRACT. Quantum computers evolve too much in this era that now we can think of decrypting the most used encrypted algorithm of the internet that is RSA Algorithm. A system for quick factoring of large numbers and related topics for deciphering messages or documents based on the quantity algorithm of Shor. In the field of quantum computing there is a very important algorithm which is known as Shor's algorithm. This algorithm helps to find the prime factor of any number. The factorization of any large integer is not possible in classical computer in finite time, while Peter Shor suggested a polynomial time factorization algorithm that could do the prime factorization. This paper discuss about the modification in Shor's algorithm to increase the probability of success to get the two factors of RSA algorithm.

1. INTRODUCTION

The goal of encryption is to hide data in such a way that no one who has the data can read it unless they have a proper recipient. And the encryptions of pretty much all private information on internet rely immensely on one numerical phenomenon that is factorization. As far as we know, it really hard to

[1]*corresponding author*

2020 *Mathematics Subject Classification.* 68Q12, 81P68.

Key words and phrases. Quantum computing, Encryption, Shor quantum algorithm, Fourier transformation.

Submitted: 27.12.2020; *Accepted:* 11.01.2021; *Published:* 09.02.2021.

tell a big number and find its factors, using a normal non-quantum computer unlike multiplication which is very fast. Finding the prime numbers to multiply together to give arbitrary big known prime number appears to be slow [1]. At least the best approach we currently have (Number Field Sieve) that runs on a normal computer even a very powerful one is very slow [2]. Quantum processing investigates hypothetical comprehension over scientific properties of quantum calculation framework, for example, superposition, and entanglement for playing out the concurrent activities on information. Due to the superposition of qubits it can store either 1 0r 0 at the same time [3]. The RSA algorithm is used for cipher development and its protocol operates on the whole factorization principle which splits the composite number into a prime number object. RSA encryption uses locks as a large number or unlocking the data requires knowing the factors of that number. Usually, RSA keys can be 1024 0r 2048 bits long, but experts believe that in the near future 1024 bit keys can be broken However, any encryption algorithm is vulnerable to attack [4].

Such schemes of encryption were never unbreakable. Alternatively, their protection is focused on the immense amount of time. Modern encryption methods are especially designed so that decoding them would take so long they are practically unbreakable [5]. But this concept is modified by the quantum computer. Such devices are much more powerful than traditional computers and should be able to easily crack certain codes. Quantum computer incorporates the data processing properties of quantum physics. Operating at temperatures greater than intergalactic space with Nano-scale materials, quantum computing has the ability to overcome some of the toughest challenges in the world. Quantum system uses Quantum Algorithms. In quantum computing, a quantum algorithm is an algorithm which runs on a realistic model of quantum computation, the most commonly used model being the quantum circuit model of computation [6]. A quantum algorithm can simultaneously calculate a bunch of possible answer for a single input by using a quantum superposition, but you only get one of the answers at the end randomly different probabilities for each one.

The principle or Superposition states that “while we do not know what the state of electron is, it is actually in all possible states” simultaneously, as long as we don’t look to check [7]. When electron start interacting with each other, then they start behaving in a cloud manner, here coupling means electron cannot represent the information individually, but if we read them together in some

particular manner then we will get the information [8]. We use a bit of information in classical machines, but quantum computers define a bit different way. In quantum systems, the smallest unit is called the Qubits. We use qubits to store quantity computing information. The state of a qubit can be described by

$$|\Psi\rangle = \alpha|0> + \beta|1> .$$

The variables alpha and beta in the above equation are numbers that are complex. They are actually special states, considered as the conditions of the mathematical basis. The key behind fast and reliable quantum computation is the setup a quantum superposition that calculates all possible answer at once while being cleverly arranged so that all the wrong answer directly interfere each other. $|a> + |b> + |c> -> f(x)$ Where a, b, c are representing the base state of each integer, and how they are going to convert into the superposition.

The paper is organized as given below: Section 2 reviews the state of art of the recent works on quantum computing and its algorithms for cryptographic systems. Section 3 discuss about the existing Shor's algorithm and highlights its grey area. Section 4 shows how Shor's algorithm can be modified and it is demonstrated step wise to prove the efficiency of the modified work. Finally, the work is concluded in section 5 by enumerating the key points.

2. Related works

This section summarizes the state of art of the quantum algorithm related works published during the recent years. In [8], the authors has created a web based platform with multiple tools that has features like block sphere and probability distrubution map. Basically the author has gathered all type of quantum algorithm in his database and presented about the entanglement but with no discussion about the fast factorization of big numbers. In [9] the authors discuss about the process that are necessary to devise a fast quantum algorithm that have to find the order perfectly and algorithm must be able to make factors of any given number, ragarding to solve hard problem it must search any of the unsorted database.

Reference [10] is a survey of information security, and there the authors discuss about the negative and positive effect of quantum computation on the information technology. The author also alerts the community that the implementation of the quantum supermancy will be a great threat to cybersecurity as cryptography can be broken easily. In [11] the authors tried to solve the grover algorithm with hybrid quantum architecture but he observed that it is still far from current implementation.

Reference [12] work tells us, how a quantum fourier transform contrast the amplitude with approximate base state of qubits. The author also use DLL implmentation of c-language to create a visual basic environment by which anyone can easily send the wavelength to the resonance of diffrent frequencies. The authors in [1] have worked on Shor's factorization calculation. The considered calculations don't require a practical quantum PC for their efficient usage. Rather, Calculations misuse extra degree of arbitrariness driven by concepts and principles derived from quantum mechanical structures such as qubits, interference or state overlapping. He analysed the shor's algorithm and tell that it has so less probability of success, by using multiple dimension register he able to create a new algorithm which has ¾ more probability of success from shor's algorithm. But he hasn't mentioned that how he converting the amplitude to zero which is helping the new algorithm to resonate the base state faster.

In [2] the author has explored, how a quantum algorithm can be described as quantum circuits. He shows that there is a problem in shor's algorithm for classical computers in finding the period of the frequency that a superposition will make. To overcome it, he used vector method to reduce the extra steps for efficient and fast result. But he didn't state that how to implement those numerical value in qubit superposition. And forming a vector state of qubits needs extra interference method to implement it. Paper [3] states that the classical computer algorithms were not able to find the factors of number bigger than 1000 decimal digit. He tries to compare shor's algorithm and general number field sieve. And he stated that if in future a quantum computer able to implement specific type of quantum algorithm then the RSA encryption will be in great threat, He came to know that the quick factorization can be achieved if we implement shor's algorithm using quantum computation.

In [4] the authors discusses the quantum-motivated developmental calculations, and this methodology doesn't require a utilitarian quantum PC. The objective of this paper was to introduce essentials of the calculations and a concise survey of the most significant investigates endeavours in this field from the previous decade. The author has compared the three quantum algorithm Quadratic Sieve Algorithm, Trail Division Algorithm and Shor's algorithm and concluded that sieve algorithm is very slow as it creates entanglement between the electrons; a trail division method is only depending on small numbers. Paper [5] presents a reasonable optimization that many people tried to modify the shor's algorithm but failed drasctially. He focused on the challenges that quantum computer faced that are Modulor Exponentiation and Maintaining Qubit Recycling. He simulates the Quantum fourier transform into instant key element and tries to work on it. The review of the related works throws light on various ways of designing quantum computing algorithm which gives us the technique for proposing our modified quantum algorithm.

3. Existing Quantum Algorithm

Factorization hardness over the classical computer has led to the development of more powerful algorithms based on quantum mechanics principles, known as Quantum Algorithms [9]. There is a known algorithm that would allow quantum computers to challenge today's commonly used cryptographic systems, which is the algorithm of Shor. Shor's algorithm enables large numbers to be factored in finite time on a quantum computer. What does it mean to say that a quantum computer solves a problem more quickly than a classical computer? As is commonplace in computational hypothesis, we will for the most part consider asymptotic scaling of multifaceted nature estimates, for example, runtime or space utilization with issue size, as opposed to singular issues of a fixed size. In both the traditional and quantum settings, we measure runtime by the quantity of rudimentary activities utilized by a calculation. On account of quantum calculation, this can be estimated utilizing the quantum circuit model, where a quantum circuit is a grouping of rudimentary quantum tasks called quantum entryways, each applied to few qubits (quantum bits). To look at the presentation of calculations, we use software engineering style documentation O(f(n)) which ought to be translated as 'asymptotically upper-limited by f(n)'.

Shor's algorithm's capacity would be able to break widely used cryptographic codes such as the public key system of RSA. Experimental realization of Shor's algorithm has been a very important goal in the research of quantum computing and quantum information processing. Significance of Shor's algorithms is that it helps to find fast factorization of any given number. Finding a factor of n-bit integer requires $exp(n \wedge 1/3(logn) \wedge 2/3)$ operations using best classical algorithm. Shor's algorithm can achieve this equivalent task utilizing $O(n^2(\log n(\log\log n))$ activities, for example a quantum PC can consider a number exponentially quicker than the best known traditional calculation. The calculation is reliant on

- Modular Arithmetic
- Quantum parallelism
- Quantum fourier Transform

The Shor's algorithm consists of two parts:-

1. Conversion of the problem of factoring to the problem of finding the period.
2. Finding the period using the quantum fourier transform, and is responsible for quantum speedup.

3.1. **Classical Part.** The following steps show how to transform a guess into a better guess.

1. Calculate gcd(a,N). This can be achieved using the algorithm Euclidean.
2. If gcd(a, N) != 1, then there is a nontrivial factor of N, so we are done.
3. Instead, use the below subroutine to find the " p " duration of the following function: $f(x) = ap \mod N$, i.e., the smallest integer "r" for which $f(x+p) = f(x)$.
4. If 'r' is odd, go back to step 1.
5. If $a \wedge p/2 = -1 mod N$ go back to step 1.
6. The factors of 'N' are gcd $(a \wedge p/2 \pm 1, N)$.
7. We are done, $g = g \wedge p/2 \pm 1$.

Shor's algorithm starts with a guess might share a factor with your target number but which probably doesn't. And then the algorithm turns that into a much better guess that probably does share a factor with your number. At this time we make a guess and we try to find p so that g^p is one more than a multiple of N. $g \wedge p/2 \pm 1$ probably shares factor with N. In quantum system we can send in superposition of number and the computation done simultaneously. First resulting into a superposition of all possible powers are guess could be raised to

and then a superposition of how much bigger each of those power are, than a multiple of N. We can't simply measure the superposition to find the solution, on the off chance that we did we get a solitary component of the superposition (x, r) as output. We need to plan something smart for find all the known p solution to ruinously interfere and counteract leaving us with just a single conceivable answer.

3.2. **Problem with Classical Algorithms.** The problems are:

1. In this equation $(g \wedge p/2+1)(g \wedge p/2-1) = m.N$ if the first factor is a factor of number N, and the second factor is equal to the factor of m then these results are not useful in classical computer.

2. And, Power p might be an odd number which is impossible for classical computer to calculate that at a particular time. This is the main problem that how we are going to find that guesses power p that it will give us a remainder of 1 and must be a factor of N.

4. Enhanced Shor's algorithm

To modify the Shor's Algorithm we must create the group of prime order in such a special case that where the guess is much less than the actual number. The basic calculation used in the algorithm is the function evaluation: $f(x) = A^x \bmod N$ So we have to do below extra step to increase the speed of finding the period. Instead of making any measurements add the contents of the input register to an auxiliary register which initially has been set to 0. Rerun the algorithm. In this step and all further steps add the contents of the input register to the auxiliary register. Repeat steps 2-3 as many times as it is anticipated are necessary to obtain a meaningful result. Measure the auxiliary register. There are essentially three ways in which errors can invalidate the results obtained from Shor's algorithm evaluation using a quantum computer, namely:

- Initiation errors that begin the machine in a well-defined initial state.
- Errors that occur in the calculation or in the output of the Fourier transformation of the equation due to inaccuracies in the phases or amplitudes used in the specific computation gates.
- Errors in the final response of the calculation.

We are focusing on these problems to increase the factorization speed of the given number, and finding the period of the base state fast. To achieve that we are going to follow these steps:

Step 1: Firstly we have to initialize the registers to a $G \wedge -1/2 \sum(p = 0, G - 1) >$ superposition where, given N (the integer to be factored), $G = 2q, N2 \leq G < 2N2$,

$$|\Psi_1\rangle = 2^{-(m+n)/2}|0\rangle_0|0\rangle_1 \cdots |0\rangle_{n-1}|0\rangle_0|0\rangle_1 \cdots |0\rangle_{m-1}.$$

Step 2: Create a function $f(p) = ap \mod N$ where a is a number randomly selected < N and apply it to the state from phase 1 to get $G \wedge -1/2(p) > f(p) > N$.

Step 3: Now, the main part Apply the Quantum Fourier Transform to get the base state of all the superposition.

$$\begin{aligned}\tilde{f}_j &= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \omega^{-ij} f_i \\ &= \frac{1}{\sqrt{N}} \left(\sum_{i=0}^{N/2-1} \omega^{-2ij} e_i + \sum_{i=0}^{N/2-1} \omega^{-(2i+1)j} o_i \right) \\ &= \frac{1}{\sqrt{N}} \left(\sum_{i=0}^{N/2-1} \omega_{N/2}^{-ij} e_i + \omega_N^{-j} \sum_{i=0}^{N/2-1} \omega_{N/2}^{-ij} o_i \right),\end{aligned}$$

where $\omega_{N/2} = e^{\frac{2\pi i}{T}}, \omega_N = e^{\frac{2\pi i}{N}}$.

Step 4: To obtain a value y and perform continuous fraction expansion on y / G, perform a quantum measurement to produce some c / r ' that satisfies r'<N and $/G - c/r' < 1/2G$ resulting in r ' being the correct period r with high probability.

Step 5: Check whether r ' meets the question. If not, seek other candidates with ideals similar to y or r ' multiples.

Shor's original computing algorithm as in figure 1 for discrete logarithms uses two index registers initialized to compare $(p-1)$ states. In case the group order is a prime q, this translates into a superposition of q states. Instead of following traditional step we are modifying this step of Shor's algorithm.

Step 6: If the problem remains unsatisfied, don't return to the first step as Shor's Algorithm, take a New Auxiliary register and create a superposition of the remaining number of base state until the period is over. Superposition of

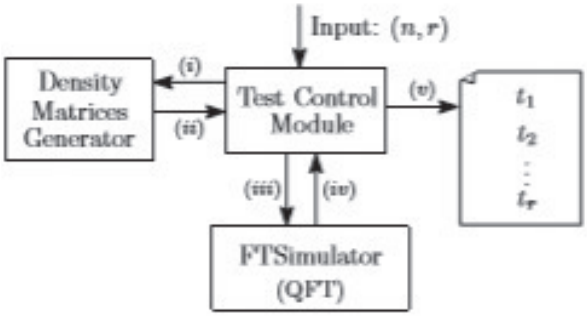

FIGURE 1.

numbers repeat periodically with a period of p. If we can find the frequency, we can find p and break the encryption.

Since finding a factor of n given the request for some component in $g \wedge p/2$ be done proficiently even on an old classical computer, regardless it stays to be demonstrated that we can discover the request for the component productively. The request discovering calculation depends essentially on a unitary operator, the "quantum Fourier transform" (QFT), which acts like a discrete Fourier change

$$|\Psi\rangle = \sum_{j=0}^{2^{\infty}-1} c_j|j\rangle \quad \xrightarrow{\text{QFT}} \quad \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} c_j e^{2xijk/2^n}|k\rangle.$$

Quantum Fourier Transform is defined as a linear operator in an orthonormal base group of $0, \ldots, 1N$ - that acts as follows on the base state of the N-dimensional space:

$$QFT_N : |a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{\epsilon=0}^{N-1} e^{2xi\underline{x}/N}|c\rangle.$$

The following formula is the transformation that acts on an arbitrary base state:

$$QFT_X : \sum_{a=0}^{N-1} x_\alpha|a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{\epsilon=0}^{N-1} \sum_{a=0}^{N-1} x_a e^{2\pi i(\alpha/N)}|c\rangle = \frac{1}{\sqrt{N}} \sum_{\epsilon=0}^{N-1} y_\ell|c\rangle.$$

Input a single number into quantum Fourier transform it will give you a superposition of all other but not any old superposition, the superposition where the other numbers are waited by different amount of time. And those wait roughly looks like sin wave frequency of single number that we put in. If we put higher

number you get a sin wave style superposition of all other numbers but with a higher frequency.

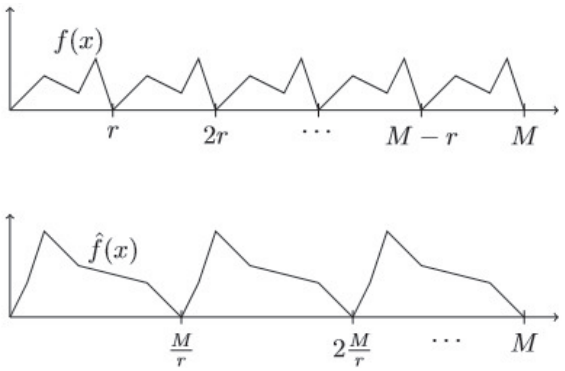

FIGURE 2.

The advantage of quantum fourier transform is when you put superposition of number, then we get superposition of superposition of the numbers and the sin waves all together as in figure 2. Then we will get a particular period by passing the superposition of number.t , this is the p in our guess, we will get period one by p by using Quantum fourier transform.

Step 7: Now we have created a new register which will create a different base state of all the non-prime number which will resonate the highest frequency that two will be the factor of our number N.

$$|\bar{\Phi}\rangle_z = QFT^{-1}\left(|\Phi_1\rangle_n\right) = \sqrt{\frac{r}{q}}\sum_{t=0}^{q/r-1}\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}\exp\left\{\frac{-2\pi i}{q}\left(a_0+rt\right)c\right\}|c\rangle$$

$$= \sqrt{\frac{r}{q^2}}\sum_{c=0}^{q-1}\exp\left\{-\frac{2\pi i}{q}a_0 c\right\}\underbrace{\sum_{t=0}^{\varphi/r-1}\exp\left\{-\frac{2\pi i}{q}\,\mathrm{trc}\right\}}_{a_e}|c\rangle.$$

To see how the calculation results are obtained by measurement, it is useful to consider the simplest example of factoring 3x5 = 15. We represent 4 qubits each of the input and output registers, i.e. n = m = 4. The condition of the input register with an overall normalization factor of 1⁄4 just before the calculation of step 1, the superposition of 0 and the first 15 integers.

Evaluating:

$$f(x) = A^x \bmod N,$$

with A=7 , we have to find non-common factor with N, by this we will get four different values namely 1,7,4,13. to show who this works the the decimal notation of state of machine is as follows:

$$\begin{aligned}\Psi_3 = \frac{1}{4}((|0\rangle + |4\rangle + |8\rangle + |12\rangle)|1\rangle) \\ + (|1\rangle + |5\rangle + |9\rangle + |13\rangle)|7\rangle) \\ + (|2\rangle + |6\rangle + |10\rangle + |14\rangle)|4\rangle) \\ + (|3\rangle + |7\rangle + |11\rangle + |15\rangle)|13\rangle)).\end{aligned}$$

We've completed the Fourier transformation:

$$\begin{aligned}\Psi_4 = \tfrac{1}{4}((0) + |4\rangle + |8\rangle + |12\rangle)|1\rangle\big) \\ +(|0\rangle + i|4\rangle - |8\rangle - i|12\rangle)|7\rangle) \\ +(|0\rangle - |4\rangle + |8\rangle - |12\rangle)|4\rangle) \\ +(|0\rangle - i|4\rangle - |8\rangle + i|12\rangle)|13\rangle)).\end{aligned}$$

One reason to write this out is to remember that the Fourier transformation actually increases the chances of measuring a specific value in the output register. If we measure after the transformation, we can only get the values 0,4, 8 and 12, each with a probability of 1⁄4 If we measured before the transformation, we could obtain any value from 0 to 15 for each value with a probability of 1/16. Results showing the final state of the input register after running the algorithm can be compared to that register's initial value. Looked at this way, we see that the first line is the algorithm that does not change the register of inputs at all. Remember also that the input register can be considered to consist of two parts; the right side that is unaffected by the algorithm and the left side in which each qubit has a probability of either changing or not changing 1⁄2.By counting the number of qubits that may change, we can calculate p. This seems simple enough, but taking input for register before the Fourier transform will increase the probability of getting a successful output. Most of the earlier work on quantum computing tacitly concluded that calculations could be done with absolute certainty, all one was searching for was a tag for a specific quantum mechanical condition. The Probability of the Distribution can be written as:

$$\frac{1}{2^{4\ell}} \cdot \left| \sum_b \exp\left[\frac{2\pi i}{2^\ell} \left(\left(b - 2^{\ell-1}\right)(dj + k) - \left\lfloor \frac{e + bd}{q} \right\rfloor \{jq\}_{2^t} \right) \right] \right|^2 ,$$

$$\frac{1}{2^4}\cdot \mid \sum_b \exp\left[\frac{2\pi i}{2^t}\left(bf_j - \mid \frac{e+bd}{q}\right\rfloor \{jq\}_{2^t}\right)\right] \mid^2 = \frac{1}{2^4}\cdot\left|\sum_b e^{if_i}\right|^2 \geq \frac{T_e^2}{2\cdot 2^{ft}}.$$

Since $|\theta b| \leq 2\pi/8 = \pi/4$ for all the Te values of b. Summing over all e, we get

$$\sum_{e=0}^{q-1}\frac{T_e^2}{2\cdot 2^{4\ell}} \geq \frac{2^{4\ell}}{q}\cdot\frac{1}{2\cdot 2^{4\ell}} = \frac{1}{2q}.$$

Because adding or subtracting multiples of 2' has no effect; it is equivalent to changing the angle of the step by a multiple of 2π So we can say that now the probability of success is more than the Shor's Algorithm. By using Hadamard Gates we created the qubit recycling and built this circuit from our algorithm as in figure 3:

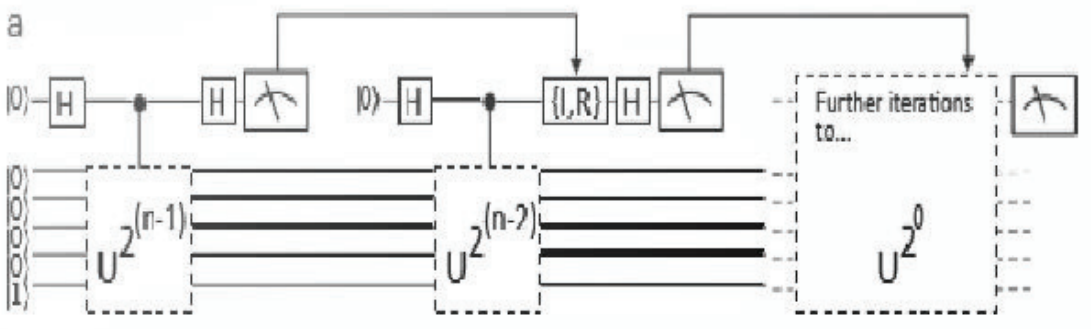

FIGURE 3.

5. CONCLUSION

The paper has demonstrated that the algorithm of Shor allows us to factor numbers much faster than traditional algorithms. It runs in

$$O((\log N)^2(\log\log N)(\log\log\log N))),$$

where N is the variable. With Shor's algorithm, factoring becomes a hard problem because on each run of the algorithm, we have a small probability of failure. We can be more and more certain of factor n by applying the algorithm multiple times. In a crucial way, the algorithm uses the QFT, which is essentially a quantum version of the FFT. In addition to factoring numbers, Shor's ideas also allowed us in polynomial time to calculate discrete logarithms. Shor's algorithm

is a true quantity algorithm that enables us to check the quantity machine capabilities. We are trying to find p because it will allow us to turn our guess into a good guess for a number that shares a factor with N, which allows us to break the encryption. And we have quantum superposition of numbers that repeat periodically with a period of p. But after modifying the Shor's algorithm we are able to find fast and reliable output from the Quantum Fourier Transform. RSA encryption is based into the integer factorization. In addition, the actual nature of the quantum computer would breach the RSA algorithm's security parameter and thus threaten the overall cryptosystem.

REFERENCES

[1] C. ZHOU, W. BAO, X. FU: *Design and Implementation of Quantum Factorization Algorithm, Intelligent Information Technology and Security Informatics*, International Symposium on Jinggangshan, China, 2010, 748-752.

[2] P. BARRERA, A. CALABRO, D. PORTO, L. FORTUNA: *A new method for implementing gate operations in a quantum factoring algorithm*, Universita' degli Studi di Milano Publ., Crema., **32** (2000), 777-780.

[3] S. M. HAMDI, S. T. ZUHORI: *A Compare between Shor's Quantum Factoring Algorithm and General Number Field Sieve*, International Conferenceon Electrical Engineering and Information & Communication Technology (ICEEICT), 2014, 1-6.

[4] K. K. SONI, A. RASOOL: *Cryptographic Attack Possibilities over RSA Algorithm through Classical and Quantum Computation*, International Conference on Smart Systems and Inventive Technology (ICSSIT), **4** (2018), 11-15.

[5] N. CHOUHAN, H. K. SAINI, S. C. JAIN: *A novel technique to modify the SHOR'S algorithm — Scaling the encryption scheme*, Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, 2017, 1-4.

[6] M. MOSCA: *Quantum Algorithms*. arXiv:0808.0369 [quant-ph]. 2008, 3:7-11

[7] R. KAMATCHI PRIYA, R. LATHA: *Literature Survey on Quantum Computing*, International Journal of Pure and Applied Mathematics, **118**(5) (2018), 901-904.

[8] Z. TAO, Y. PAN, A. CHEEN L. WANG: *ShorVis: A comprehensive case study of quantum computing visualization*, International Conference on Virtual Reality and Visualization L(ICVRV), (2017), 360-365.

[9] C. VIDYA RAJ, H. D. PHANNENDRA, M. S. SHIVAKUMAR: *Quantum algorithms and hard problems*, Proc. 5th IEEE Int. Conf. on Cognitive Informatics ICCI'06, (2018), 783-787.

[10] M. NJORBUENWU, B. SWAR, P. ZAVARSKY: *A Survey on the Impacts of Quantum Computers on Information Security*, 2nd International Conference on Data Intelligence and Security ICDIS, (2019), 212-218.

[11] R. T. POSSIGNOLO, C. B. MARGI: *A quantum-classical hybrid architecture for security algorithms acceleration*, IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, (2012), 1032-1037.

[12] Z. LI, J. WANG, H. CHEN: *The Study of Simulation Technique of Quantum Compute and Quantum Fourier Transform*, IEEE, **2** (2019), 1111-1115.

RESEARCH DIVISION OF ADVANCED DATA SCIENCE
VELLORE INSTITUTE OF TECHNOLOGY
CHENNAI
Email address: Nitin.asthana2019@vitstudent.ac.in

RESEARCH DIVISION OF ADVANCED DATA SCIENCE
VELLORE INSTITUTE OF TECHNOLOGY
CHENNAI
Email address: shridevi.s@vit.ac.in