Analysis of Shor’s Algorithm and Quantum Resistant Cryptographic Systems

Helen Huang

Abstract—**Shor’s algorithm paved the development of quantum computing theories and technologies and in turn the need for quantum resistant cryptography systems. A brief overview of time complexity characterization, Rivest-Shamir-Adelman cryptosystem, and quantum computing terminology is done to provide context to Shor’s algorithm and its corresponding resistant cryptosystems. To break the Rivest-Shamir-Adelman encryption system, Shor’s algorithm in conjunction with quantum computers exhibits a significantly lower complexity over classical algorithms. The application of quantum Fourier transform in Shor’s algorithm is the primary technique to lower the time complexity required to factor a large number. An analysis of the time complexity and quantum mechanics of Shor’s algorithm concludes with a polynomial complexity of $O(N)^3$. Current classical algorithms, such as the general number field sieve and quadratic sieve determine factors of large numbers with exponential time complexity. The paper focuses on two cryptographic techniques that are resistant to Shor’s algorithm. Quantum key distribution, specifically the BB84 protocol, focuses on the usage of quantum mechanics to prevent quantum computing attacks. The McEliece cryptosystem relies heavily on its problem class of NP-hardness to keep data secure.**

Index Terms— **Computational complexity, Cryptography, Fourier transforms, Quantum computing,**

I. INTRODUCTION

Exploration in the 1980s of quantum theory and mechanics forged the foundation for quantum computers and the inception of Shor’s algorithm. Understanding quantum computing relies heavily on multiple disciplines including quantum physics, computer science, and mathematics that delves deeply into number theory. Potential applications of quantum computing are diverse and comprised of fields from medical research to a tremendous emerging concern: the security of current cryptographic systems. An ongoing and highly popular form of cryptosystem is Rivest-Shamir-Adelman (RSA) which encrypts commonly used digital components, including electronic money as in credit card numbers, digital signatures, and secure network communications. RSA cryptosystems rely heavily on the factorization of large integers which, at the time, was considered an intractable problem. Introduction of Shor’s algorithm altered the landscape of cryptography as it shows the capacity to breach RSA security with the use of quantum computing. Evaluation of time complexity of Shor’s algorithm will expose the exponential reduction in complexity over classical algorithms like the general number field sieve and the quadratic sieve. Continuing apprehension regarding the expansion of quantum computing algorithms capable of collapsing RSA encryption compelled the return of old cryptographic systems like the McEliece cryptosystem and new cryptographic protocols as presented through quantum key distribution that utilize quantum computing to secure our data. Quantum key distribution may provide a route towards averting the security issues that will emanate as quantum computers scale and their prevalence increases. Although, the McEliece cryptosystem was developed in the late 1970s and employs traditional techniques of cryptography, it has been proven to be resistant to Fourier sampling techniques emphasized in Shor’s algorithm. An extensive overview of Shor’s algorithm provides insight to the urgency for cryptographic system evolution.

II. UNDERSTANDING TIME COMPLEXITY

Time complexity is a crucial concept, therefore an introduction to the relevant terminology is essential. The impact of Shor’s algorithm on the domain of RSA cryptosystems is rooted in time complexity concepts. To properly grasp Shor’s algorithm, we need to define Big-Oh complexity which can be defined as the function:

$$f(n) = O(g(n)) \tag{1}$$

if and only there exist positive constants c and n_0 such that $f(n) \leq c * g(n)$ for all $n, n \geq n_0$ [7]. Familiarity with polynomial time complexity, exponential time complexity and linear time complexity is vital to comprehension of the components involved in interpreting the optimal algorithm. Polynomial time complexity will have a Big-Oh notation of:

$$O(n^k) \tag{2}$$

for any $n, k \in \mathbb{N}$. A function with exponential time complexity will have a Big-Oh notation of:

$$O\left(2^{n^k}\right) \tag{3}$$

for any $n, k \in \mathbb{N}$. Linear time complexity may be defined as a function with Big-Oh notation of:

$$O(n) \tag{4}$$

for any $n \in \mathbb{N}$. Intractability may be formally defined as a problem that is so difficult that no polynomial time algorithm can possibly solve it [8]. The class of NP represents nondeterministic polynomial problems that are verifiable in polynomial time [8]. NP-hard problems consist of problems that can be transformed to an NP-complete problem with the property that it cannot be solved in polynomial time unless $P = NP$ [8].

III. UNDERSTANDING QUANTUM COMPUTING

Quantum computers use quantum mechanics to execute operations on data. Classical computers encode information in bits via the values 1 and 0 which represent on and off operators. Meanwhile, quantum computing encodes information as qubits, typically through the polarization of ions, and exerts the principles of quantum physics which involves superposition and entanglement. Qubits can be defined as a two-level quantum system consisting of the states $|0\rangle$ and $|1\rangle$ which are termed basis states [6]. A general state of a single quantum bit is a vector:

$$c_0|0\rangle + c_1|1\rangle \tag{5}$$

with unit length of $|c_0|^2 + |c_1|^2 = 1$ [6]. Observation of a quantum bit in state (5) will give 0 or 1 as an outcome with probabilities of $|c_0|^2$ and $|c_1|^2$, respectively [6]. Superposition allows for every qubit to be defined by the state 1, 0, or both at the same time. With superposition, each qubit can represent two states simultaneously, thus each qubit can contain two bits of information. Entanglement allows for the state of a qubit to depend on the state of another qubit therefore allowing multiple qubits to work synchronously. To demonstrate the potential of superposition and entanglement, imagine we were given 7 qubits, then at any time these qubits represent some combination of 1's and 0's out of the possible $2^7 = 128$ states. With entanglement, all 128 states can concurrently exist and be edited. Naturally, the concepts of superposition and entanglement provide qubits with the power to function as more intricate on and off operators as opposed to classical computers. Quantum registers are vectors holding the state of the entangled qubits [6]. Measuring a quantum state causes a quantum collapse which is the transition from superposition of quantum states to a measurable component state [6]. Hilbert space is defined as a vector space H with an inner product $\langle f, g\rangle$ such that the norm defined by $|f| = \sqrt{\langle f, f\rangle}$ turns H into a complete metric space [10]. Quantum computing exploits interference to destroy incorrect solutions, comparable with diffraction gratings in double-slit experiments. Processing information on a classical computer is performed sequentially, while quantum computers interpret data simultaneously. As such, implementation of quantum computing algorithms generally enhances the efficiency and speed of processing data, compared to classical algorithms.

IV. UNDERSTANDING CRYPTOGRAPHY

Cryptography terminology is fundamental to understanding RSA, McEliece cryptosystem, and quantum key distribution. To discuss cryptography, we will use the customary definitions of Alice, Bob, and Eve. Alice and Bob are two parties who want to communicate securely, and Eve is the unauthorized eavesdropper [12]. Security of cryptography depends on the secrecy of the key rather than the encryption process. Keys consist of public keys and private keys. Public keys are values that everyone, including Eve, has access to and are used in conjunction with private keys to encrypt a message. Private keys are values that only the recipient has access to and guide the decryption of a message. Construction of private keys are formed such that Eve cannot determine the private key easily.

V. RSA CRYPTOGRAPHY

Of the numerous cryptography systems that exist, RSA is a frequently used standard for the transmission of sensitive personal information. The system falls under the category of asymmetrical cryptosystems. During encryption, RSA embeds a form of structured randomized padding into the message. Embedded padding increases the time complexity required to break the RSA encrypted message. The primary design of RSA encryption depends upon the designation of determining factors for large numbers as belonging to NP. Encrypting messages using RSA involves the public key which is known to everyone and decrypting messages employs the private key. Generating keys in RSA relies on two basic mathematical definitions: prime number and greatest common divisor (gcd). A prime number is a number with factors consisting of only one and itself. The gcd of two non-zero integers is the largest positive integer that divides each of the integers. Keys for the RSA algorithm are generated by choosing two different very large random prime numbers, p and q with [9]:

$$N = pq \tag{6}$$

The Euler totient function can be defined as [9]:

$$\Phi(N) = (p-1)(q-1) \tag{7}$$

Choice for the public key exponent integer, e, where $1 < e < \Phi(N)$ is contingent on the necessity that [9]:

$$\gcd\big(e, \Phi(nN)\big) = 1 \tag{8}$$

For the private key exponent d, we choose a $k \in \mathbb{Z}$ such that [9]:

$$de = 1 + k\Phi(N). \tag{9}$$

We will not delve into the specifics of encrypting and decrypting a message as understanding Shor's algorithm primarily requires details on determining determine the value N as referenced in (6).

VI. QUADRATIC SIEVE AND GENERAL NUMBER FIELD SIEVE

During the popularization of RSA encryption, in the 1970s, factoring 20-digit numbers was considered an extremely difficult problem [5]. While Brillhart-Morrison's algorithm could factor 50-digit numbers, integers greater than 50-digits were still considered an intractable problem in the 1980s [5]. Algorithms of the time were inefficient and only worked for certain number instances. Pomerance created the quadratic sieve (QS) factoring algorithm in the 1990s, which could factor up to 129-digit numbers [5]. The QS remains as one of the most prominent methods for factoring numbers up to 129 digits. For larger numbers, Pollard's 1996 creation of the general number field sieve (GNFS) was able to factor a number up to 130 digits [5].

We will explore the time complexities of the QS and GNFS as use for comparison with the complexity of Shor's algorithm using quantum computing. The number of operations necessary to factor a binary number of N bits using the QS algorithm is [1]:

$$O\left(\exp\left(\left(\frac{64}{9}\right)^{\frac{1}{3}} N^{\frac{1}{3}} (\ln N)^{\frac{2}{3}}\right)\right) \tag{10}$$

From complexity (10) we see that it scales exponentially with the input size. The QS algorithm is a deterministic factoring algorithm with a conjectured time complexity of [3]:

$$O\left(\exp\left(\big(1 + o(1)\big)(\log n \log\log n)^{\frac{1}{2}}\right)\right) \tag{11}$$

By the definition of exponential time complexity, we can formulate that (11) has exponential complexity. Although, there are deviations in the specific implementation of the QS, the time complexity estimate remains approximately the same, as these variations generally only alter the $o(1)$ term. Given an integer n consisting of $\lfloor \log_2 n \rfloor + 1$ bits, the time complexity for GNFS is [5]:

$$O\left(\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\ln n)^{\frac{1}{3}} (\ln\ln n)^{\frac{2}{3}}\right)\right) \tag{12}$$

Again, by the definition of exponential time complexity, we can determine that (12) has exponential complexity. It should be noted that the time complexities for the QS and the GNFS are algorithms designed for classical computers.

VII. SHOR'S ALGORITHM

A significant development came during the advent of quantum computing in 1994 with Shor's algorithm. Shor's algorithm shows that with quantum computers, it is possible to factor significantly larger numbers than previous classical algorithms [6]. The primary elements of Shor's algorithm incorporate Quantum Fourier Transform (QFT), Euclidean algorithms, and Euler's theorem. A majority of Shor's algorithm is performed on classical machines, but we can exploit periodicity through QFT via the use of quantum computers. QFT is a linear transformation that maps one vector of complex numbers to another vector of complex numbers [9]. Periodicity of a function is essentially how often a function repeats itself. Essentially, QFT takes a function $f(x)$ and determines the periodicity. Functions may have a large value for periodicity if the function extends for a large sequence of terms before repeating. Therefore, classical algorithms that apply Fourier transform would be highly inefficient because each individual value in the sequence is evaluated to determine overall periodicity. Quantum computers resolve this problematic mechanism by creating a superposition on each value in the sequence.

A comprehensive view on each component of Shor's factoring algorithm is necessary to interpret the reduction of time complexity resulting from usage of this algorithm. We consider an integer n consisting of N bits, and determine if n is prime, an even number, or an integer power of a prime number [9]. If n belongs to any of these categories, we resort to classical algorithms. Now, we pick an integer q which is a power of 2 such that $n^2 \leq q < 2n^2$ and an integer x such that $x < N$ and $\gcd(x, N) = 1$ [9]. The performed steps, so far, are only computed through classical machines. A quantum register is created and divided into two registers with reg1 denoting register one and reg2 denoting register two. Register one must be large enough to hold q therefore we must ensure reg1 has enough qubits to represent integers as large as $q - 1$ [9]. Register two must be large enough to hold n therefore we must ensure reg2 has enough qubits to represent integers as large as $n - 1$ [9]. By doing so, we can represent the current state of the quantum computer as $left|\mathrm{reg1}, \mathrm{reg2}\rangle$ [9]. Loading reg1 with an equally weighted superposition of every integer from 0 to $q - 1$ which we define as the state $|a\rangle$ and loading reg2 with the 0 state updates the current state of the quantum computer to [9]:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a,0\rangle \tag{13}$$

Notice that $\frac{1}{\sqrt{q}}$ is a normalization constant resulting from the definition of 2-dimensional Hilbert space. Creating the quantum register, dividing it and loading the register will take at most linear time. From here, we let $a \in \mathbb{Z}$ such that $f(a) = x^a \bmod N$. Computation of the transformation for $f(a)$ of each number is stored in reg1 and we then store the result in reg2. By definition of entanglement, applying this transformation on reg1 takes only one step. The current total state of the quantum memory register is [9]:

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a,x^a \bmod n\rangle \tag{14}$$

Prior to this, we have been utilizing aspects from classical algorithms and encoding our registers with the values. It is at this point we see a reduction in time complexity as entanglement allows for the complexity to be lowered to polynomial time. At this stage, the transformation will have a complexity of $O(\mathrm{N}^3)$. A measurement of reg2 will collapse reg2 into a measurable component state we call k which we save in reg2. Both registers, reg1 and reg2, are entangled, therefore reg1 will collapse into an equal superposition of each value a between 0 and $q-1$ such that $x^a \bmod n = k$. All values of a will collapse to a'. To hold the original collapsed set of a values such that $x^a \bmod n = k$, we create a new vector A. An updated look at the total state of the quantum memory register at this point is [9]:

$$\frac{1}{\sqrt{\|A\|}}\sum_{a'=a'\in A}|a',k\rangle \tag{15}$$

Notice that $\|A\|$ is the number of elements in that set and $\frac{1}{\sqrt{\|A\|}}$ is the new normalization constant. Applying QFT on reg1 will transform the state $|a\rangle$. Recall that a' is such that it consists of all values that satisfy $x^a \bmod n = k$. With a probability amplitude graph, we will see that values of a' which have a high probability of being a multiple of $\frac{q}{r}$ will peak, where r is the desired period. Application of QFT to reg1 can be represented by the following state [9]:

$$|a\rangle = \frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}|c\rangle * e^{\frac{2\Pi iac}{q}} \tag{16}$$

Following QFT, the state of reg1 is now [9]:

$$\frac{1}{\sqrt{\|A\|}}\sum_{a'\in A}\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}|c,k\rangle * e^{\frac{2\Pi iac}{q}} \tag{17}$$

Again, due to entanglement, the QFT will be completed in a single step therefore, the QFT will have a complexity of $O(N^3)$. We let m be the measurement of the state of reg1 then the probability for state $|c, x^k \pmod n\rangle$ to occur is:

$$p(c) = \left|\frac{1}{\sqrt{q}}\sum_{c=0}^{q-1} e^{\frac{2\Pi iac}{q}}\right|^2 \tag{18}$$

As values of a' have peaked at multiples of $\frac{q}{r}$, the integer m, has a very high probability of being a multiple of $\frac{q}{r}$. Based on the values of m and q, we will be able to calculate r using various algorithms that are not presented here for the sake of brevity. Finally, we can compute:

$$d_1 = \gcd\left(n, x^{\frac{r}{2}} - 1\right) \tag{19}$$

and

$$d_2 = \gcd\left(n, x^{\frac{r}{2}} - 1\right) \tag{20}$$

using the Euclidean algorithm which usually gives us that either d_1 or d_2 is a non-trivial factor of N. There is a low probability that the choice of a will not lead to factors of n, resulting in the need to restart the algorithm. Generally, the probability that a random a will work is high enough that the algorithm typically only needs to be run a few times before it is able to find a non-trivial factor of n. From following Shor's algorithm, we identify that one of the largest reductions in time complexity came during the transformations, which had complexity of $O(N^3)$. Accordingly, we can say that Shor's algorithm has an approximate complexity of $O(N^3)$, where N is the number of bits of some integer. A more accurate estimate for Shor's algorithm shows that the complexity lies around $O(72N^3)$ [2]. Estimating the time complexity for algorithms that require quantum computing is not always accurate, because knowledge of quantum computing is still limited. For instance, the number of steps and processing speed at any given can vary based on factors such as the architecture of the quantum computer [2]. Current approaches to factoring prime numbers demands exponential time, meaning it may take years to factor a large number. Reducing an algorithm from exponential time to polynomial time is an astronomical decrease in complexity and widely considered a major feat in both quantum computing and cryptography.

Recall the two classical algorithms for integer factorization, QS and GNFS. QS has a complexity of (10) and GNFS has a complexity of (12), both of which are exponential time complexities. With Shor's algorithm, the approximate complexity is $O(N)^3$. By definition of polynomial time complexity, we evaluate that Shor's algorithm is of polynomial complexity. It becomes apparent that there is an exponential time reduction between classical and quantum algorithms. The premise of RSA primarily involves the difficulty of integer factorization, however finding the value of N, with reasonable time, in (6) becomes a plausible outcome with the use of Shor's algorithm. With QS or GNFS, the time required to process a larger number could be years. If quantum computers are able to fully apply Shor's algorithm for large numbers, the time reduction will be monumental.

Development of Shor's algorithm occurred before quantum computers were created. Initially it was unclear if Shor's algorithm would have proper applications within the realm of quantum computing. Early designs of quantum computers utilized ion traps with five atoms and approximately twelve qubits to factor the number fifteen. Recent quantum computing schemes need only five qubits to factor the number fifteen [5]. Scalability was determined to be achievable through additional atoms and lasers to construct a larger and faster quantum computer that carries the ability to factor significantly larger numbers [5]. Currently the record for the largest number factored through quantum computation is 291311 [16]. There is growing concern that RSA encryption will be rendered useless with further advancements in quantum computers as they will be able to apply Shor's algorithm to large numbers. Considerable strides have been made to quantum computers, however they remain unable to factor large enough numbers that would result in the destruction of RSA encryption. For now, cost remains a major barrier in scalability, thus RSA cryptography remains secure.

VIII. MCELIECE CRYPTOSYSTEM

Though, the McEliece cryptosystem (MECS) was officially published near the time of RSA, MECS gained less traction due to its use of large public key size. With the onset of quantum computing, interest in the MECS was renewed. Security of MECS relies on decoding linear code which is considered an NP-hard problem. Key generation and the resistance of MECS to Shor's algorithm is the primary discussion topic. To generate a key in the MECS, there will exist a generator matrix G for the code C, where G is a $k \times n$ matrix. Alice will select a random, binary, non-singular $k \times k$ matrix we call S and a $n \times n$ random permutation matrix we call P [13]. The public key will be computed as follows:

$$G' = S * G * P \tag{21}$$

Alice's private key will be:

$$(G, S, P) \tag{22}$$

In Shor's algorithm, QFT lowers the complexity of factoring large numbers. However, Fourier sampling is unable to crack the MECS [14]. Therefore, the MECS does not share the same vulnerabilities as other cryptosystems like RSA. The leading concern with the MECS is that it does secure messages against known quantum computer methods, like Shor's algorithm, but it is plausible that another quantum algorithm will be developed that does break the MECS. As the MECS is generally a less popular form of cryptography, it is possible that MECS has not undergone the rigorous testing and research compared to RSA. Research suggests the MECS may provide the simplest and at the least short-term insusceptibility to common quantum algorithms, because MECS is a well-understood algorithm that when applied properly has not been cracked by any currently known algorithms with a reasonable complexity.

IX. QUANTUM KEY DISTRIBUTION: BB84 PROTOCOL

Certain classical cryptography schemes, especially those that rely on large integer public keys, will become futile as quantum computers advance. Technological advancements in quantum computing are expected to replace these outdated cryptography systems with new and highly secure systems which are resistant to Shor's algorithm. One such cryptography scheme that avoids the abounding security issues that will arise from quantum computers and the implementation of Shor's algorithm is quantum key distribution.

Introduction to necessary terminology is essential to conceptualizing QKD. Raw keys consist of strings of received bits that are produced from sender information [12]. There are three primary concepts to understanding QKD, which consist of the no-cloning theorem, Heisenberg's uncertainty principle, and quantum mechanics. The descriptions of Alice, Bob, and Eve as defined earlier will be used in the discussion of QKD. The no-cloning theorem declares that given an unknown quantum state, it is not possible to create a copy of said quantum state [12]. Heisenberg's uncertainty principle asserts that it is not possible to simultaneously measure complementary variables with arbitrarily high precision [12]. As examined earlier, measurements on a quantum state will cause a collapse of said state, therefore it is not possible to measure a quantum state without disturbing the system. Recall, entanglement is a mechanism by which changes to a single state cause transitions to all other states. In QKD, the quantum state holds the data to transfer. If an eavesdropper attempts to measure an entangled photon then the resulting collapse will cause changes to the state of all entangled photons. From the no-cloning theorem, it follows that a single quantum state cannot be cloned [12].

While there is a plethora of distinctive QKD designs, our discussion of QKD will center on BB84 as it is one of the first and most popular QKD schemes. BB84 applies quantum mechanics to allow two parties to generate a shared secret. Quantum channels are typically optical fibers and classical public channels are usually phone lines or an Internet connection [12]. Horizontal-vertical basis of photons can be defined as whether photons are horizontal polarized or vertically polarized. The diagonal basis of the photon is the angle at which the photon is polarized. The possible angles of polarization of the photon can either be +45° polarized or −45° polarized. Fundamentally, QKD works by sending photons through these channels. To ensure security, Alice can randomly choose the horizontal-vertical basis and the diagonal basis of the photon. Bob then receives the polarization state from Alice and for each photon Bob will choose one of the two bases. From this, Bob will either have the correct base or he will have the incorrect base, which will allow Bob to receive the raw key. Alice and Bob will compare the selected bases and work together to correct the errors. The protocol for QKD will resemble the following figure [15]:

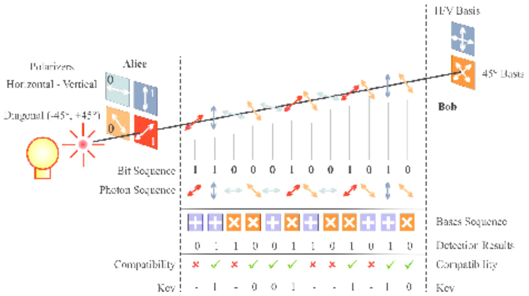

Fig. 1 Quantum Key Distribution protocol

The BB84 protocol is undoubtedly less vulnerable to attacks from quantum computer attacks than cryptosystems like RSA. Shor's algorithm is entirely unable to penetrate this type of security system, as the BB84 protocol does not rely on large numbers to protects its private key rather it uses randomized transfer of photons to protects its message. Attempts from Eve to intercept information will typically be detected as Alice and Bob will obtain a 25% error rate in their sifted key [12]. Despite, the improvements to known security issues, concerns remain about the overall susceptibility of the BB84 protocol. Provided a large set of quantum computers used in conjunction, there is a possibility that the BB84 protocol will be powerless to defend against such attacks. As we do not have access to such a set of quantum computers, we do not know the exact behaviors or performance of these attacks. Denial of service (DOS) attacks can be used to increase the error rates of transmission between Alice and Bob, which can prevent proper transfer of data. However, there are arguments that these DOS attacks do not present a genuine obstacle because Eve would require physical access to classical public channels and for quantum channels, large data networks would prevent most attacks [11]. From various sets of experiments and research, as quantum computing progresses, it appears that BB84 will likely be adopted as the cryptosystem standard.

X. CONCLUSION

Shor's algorithm provided a glimpse of the possibilities of quantum computers by reducing a once intractable problem to a polynomial time complexity. Increased research in quantum computing has increased speculation about the state of security of cryptographic systems. While, Shor's algorithm proves the ability to break encryptions based on public keys created from large integers, like the RSA encryption scheme, there are many unexplored cryptography methods that currently remain secure. Quantum computing is a relatively new, but fast approaching field, new and old cryptographic systems will likely improve with increased knowledge of the behavior of a quantum computer.

REFERENCES

[1] S. L. Braunstein, "Quantum computation: a tutorial," [Online]. Available: https://www.saylor.org/site/wp-content/uploads/2011/06/CS411-5.1-1.pdf/.

[2] D. Beckman, A. Chari, S. Devabhaktuni and J. Preskill, "Efficient networks for quantum factoring", *Physical Review A*, vol. 54, no. 2, pp. 1034-1063, 1996.

[3] C. Pomerance, "Smooth numbers and the quadratic sieve", Mathematical Sciences Research Institute, vol. 44, pp. 69-81, 2008.

[4] C. Pomerance, "A Tale of Two Sieves", Notices of the American Mathematical Society, vol. 43, no. 12, pp. 1473-1484, 1996.

[5] Chu, "The beginning of the end for encryption schemes?", MIT News, 2016. [Online]. Available: http://news.mit.edu/2016/quantum-computer-end-encryption-schemes-0303.

[6] M. Hirvensalo, Quantum computing, 2nd ed. Berlin: Springer-Verlag, 2010, pp. 1-47.

[7] E. Horowitz, S. Sahni and S. Rajasekeran, Computer Algorithms, 2nd ed. New Jersey: Silicon Press, 2008.

[8] M. Garey and D. Johnson, Computers and Intractability A Guide to the Theory of NP-Completeness. New York: W.H. Freeman and Co., 1979.

[9] M. Hayward, "Quantum Computing and Shor's Algorithm", University of Illinois, Urbana-Champaign, 2005.

[10] E. Weisstein, "Hilbert Space", MathWorld--A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/HilbertSpace.html. [Accessed: 03- May- 2018].

[11] A. Price, J. Rarity and C. Erven, "A quantum key distribution protocol for rapid denial of service detection", in QCrypt 2017, Cambridge, 2017.

[12] P. Pajic, "Quantum Cryptography", University of Vienna, Vienna, 2013.

[13] M. Repka and P. Zajac, "Overview of the Mceliece Cryptosystem and its Security", Tatra Mountains Mathematical Publications, vol. 60, no. 1, 2014.

[14] H. Dinh, C. Moore and A. Russell, "The McEliece Cryptosystem Resists Quantum Fourier Sampling Attacks", 2010.

[15] A. Iqbal, M. Aslam and H. Nayab, "Quantum Cryptography: A brief review of the recent developments and future perspectives", in The International Conference on Digital Information Processing, Electronics, and Wireless Communications, Dubai, 2016, pp. 43-46.

[16] Li, N. Dattani, X. Chen, X. Liu, H. Wang, R. Tanburn, H. Chen, X. Peng and J. Du, "High-fidelity adiabatic quantum computation using the intrinsic Hamiltonian of a spin system: Application to the experimental factorization of 291311", 2017.