

Quantum Cryptography for Secured Key Distribution

M.B Hammawa¹, Oluyemi E. Amujo¹, A.Y. Atumoshi¹, A. Abdulrahman²

¹Department of Computer Science, University of Abuja, Nigeria

²Department of Maths/Computer Science, IBB University, Lapai, Nigeria

Abstract: The strength of modern cryptography algorithms are based on the factors such as keys-space, transmission techniques and fundamental process of executing one-way/trapdoor functions which are said to be intractable. But modern cryptography is vulnerable to advancement in technology and computing power that can make possible billions guesses of keys within a short period of time. In other words, by the evolvement of high-speed computers such as quantum computers, the popular schemes used in modern cryptography are becoming insignificant. As a way-forward, the quantum mechanics and computer engineering together offer a new way for securing data known as quantum cryptography with major primitive such as quantum key distribution (QKD). QKD provides secure key distribution over a network by the laws of quantum physics and ensures that encryption keys are transmitted securely even in the presence of eavesdropper. This paper focuses on quantum cryptography and how it contributes value to a long-time strategy pertaining to completely secure key distribution. The scope of this paper covers the weaknesses of modern/classical cryptosystems, the fundamental concepts of quantum cryptography that contributes to its strength such as entanglement, superposition and quantum uncertainty. Finally, the paper demonstrates, through evaluation method, that all the family of one-way/trapdoor functions is tractable on quantum system.

Keywords: Quantum, cryptography, entanglement, superposition, encryption, performance, protocols, algorithms, simulator, one-way function, intractable algorithm, polynomial algorithm.

I. INTRODUCTION

Cryptography ensures that communication in the presence of adversary i.e. people can communicate securely and selectively. Today almost every system that needs secure communication uses cryptography as it protects the privacy and confidentiality qualities of information.

Research works by Sheila (2011) and Madeline (2015) showed that no system is perfectly secure for a long period of time and no encryption scheme is capable of ensuring 100% security of information against adversary. The reasons for these can be traced back to computer speed and length of key. Modern cryptography is vulnerable to both technological progress of computing power and evolution in mathematics to quickly reverse one-way functions such as that of factoring large integers. High-powered supercomputers can crack many of today's standard encryptions, and those encryption schemes that are not breakable now will become so in the near future as the speed and power of supercomputers continue their ever-accelerating uptick while the development of quantum computers also make it worse. Modern cryptography algorithms are based on the fundamental process of factoring large integers into their primes, which is said to be "intractable". Nullification of this assumption is possible using a system that acts faster than adversary capability. Lastly, the encryption procedures involve transmitting part of the encryption key alongside the encrypted message and part is maintained by third party on a so call secure server such as public key infrastructure (PKI). The security of our information therefore depends on the degree of security of third party server which we cannot tell.

To secure information for the future, the experts have introduced quantum physics into cryptography, which leads to evaluation of quantum cryptography. While the most well-known primitive of this discipline is quantum key distribution (QKD), there exist others like quantum coin tossing, quantum money, quantum copy protection, quantum private channels, blind quantum computation and quantum public key encryption (Anne & Christian, 2015). Unlike traditional cryptography, where the security is usually based on the fact that an adversary is unable to solve a certain mathematical problems such as one-way and trapdoor function, QKD achieves security through the laws of quantum physics. More precisely, it is based on the fact that an eavesdropper, trying to intercept the quantum communication, will inevitably leave traces which can thus be detected. In this case, the QKD protocol aborts the generation of the key.

Consequence to the development of Quantum cryptography, European Union members have announced their intention to invest in the research and development of a secure communications system based on this technology (Sheila, 2011). The system, known as SECOQC (Secure Communication based on Quantum Cryptography), will serve as a strategic defense against the Echelon intelligence gathering system. In addition, a handful of quantum information processing companies, including MagiQ Technologies and ID Quantique, are implementing quantum cryptography

solutions to meet the needs of businesses, governments, and other institutions where preventing the unauthorized disclosure of information has become a critical success factor in maintaining a competitive advantage over adversaries.

Similarly, Quantum computing is a worthwhile theoretical area to study (Madeline, 2015) as it will also have a radical effect on the usability of some of the algorithms employed today such as Shor's Algorithms for Integer Factorization and Discrete which include Fourier Transform, Integer Factorization, and Finding Discrete Logarithms, all which can be solved in polynomial time.

II. LITERATURE REVIEW

2.1 Non-quantum Cryptography

The modern, non-quantum cryptography, the ones implemented on classical computers is based on a gap between efficient algorithms for encryption for the legitimate users versus the computational infeasibility of decryption for the adversary; it requires that one have available primitives with certain special kinds of computational hardness properties. Of these, perhaps the most basic is a one-way function. Informally, a function is one-way if it is easy to compute but hard to invert. Other primitives include pseudo-random number generators, and pseudorandom function families, from such primitives; it is possible to build secure encryption schemes.

According to Shafi and Mihir (2008), modern cryptography abandons the assumption that the adversary has available infinite computing resources, and assumes instead that the adversary's computation is resource bounded in some reasonable way. Similarly, the encryption and decryption algorithms designed are probabilistic and run in polynomial time. The running time of the encryption, decryption, and the adversary algorithms are all measured as a function of a security *parameter* k which is a parameter which is fixed at the time the cryptosystem is setup". Thus, when we say that the adversary algorithm runs in polynomial time, we mean time bounded by some polynomial function in k .

2.2 Limiting Factors of Non-quantum Encryption Scheme

The security of any cryptosystem depends on keeping some or all of the elements of the cryptovvariable(s) or key(s) secret (Feryal, 2003) and effective security are maintained by manipulating the size (bit length) of the keys and by following proper procedures and policies for key management. The factors are discussed below;

A. Key Distribution Factor

In key distribution, one must worry about two (related) issues: the lack of authenticity and key secrecy (Shafi and Mihir, 2008). One of the challenges in symmetric encryption is that both the sender and the recipient must have the secret key. Also, if either copy of the key falls into the wrong hands, messages can be decrypted by others and the sender and intended receiver may not know the message was intercepted. The primary challenge of symmetric key encryption is getting the key to the receiver, a process that must be conducted out of band (meaning through a channel or band other than the one carrying the ciphertext) to avoid interception.

The problem with asymmetric encryption is that holding a single conversation between two parties requires four keys. Moreover, if four organizations want to exchange communications, each party must manage its private key and four public keys. In such scenarios, determining which public key is needed to encrypt a particular message can become a rather confusing problem, and with more organizations in the loop, the problem expands. This is why asymmetric encryption is sometimes regarded by experts as inefficient.

B. Computing Power Factor

Roel, (2015) noted that the strength of a cryptographic algorithm is expressed in the total amount of computations an adversary needs to perform to recover the secret key. It is often referred to as the computational complexity of the cipher. Obviously the strength of many encryption applications and cryptosystem is measured by the key size or key space. Typically, the length of the key increases the number of random guesses that have to be made in order to break the code. Creating a larger universe of possibilities increases the time required to make guesses, and thus a longer key directly influences the strength of the encryption. Contrarily, the increasing rate of computing power has already been observed in 1965 by Gordon Moore, co-founder of Intel. The Moore's law observes that computers get faster and faster. Moore noticed a pattern that processor speed doubles every 18 months e.g PC in 1987, speed was 1.5 Mhz, up till 2002 when the speed became 1.5 GHz. Similar pattern also occurs for memory and storage. But this doubles every 12 months e.g. Maximum Hard disk requirement in 1991 was 20 Mega bytes but rose in 2002 to 30 Giga bytes.

Similarly, Cheng-Jing (2008) in his work prefers to call cipher "computationally secure" rather than "unconditionally secure". The latter is true if no matter how much the ciphertext is intercepted, there is not enough information to determine the corresponding plaintext uniquely. By the way, we have to realize that all ciphers are breakable if given

unlimited resources, therefore, the latter is countered. So generally speaking, the former is sometimes more meaningful according to Cheng-Jing (2008), which means if it can be broken by systematic analysis with available limited resources.

Computationally secure is established with the two criteria meet at the same time:

- i. the cost of breaking the cipher exceeds the value of the encrypted information.
- ii. the time required to break the cipher exceeds the useful lifetime of the information.

The observations are well explanatory and also take us close to the point. If information has no value to the hacker, he will not border spending his resources on hacking it. In other words, if the value on the information matters to the hacker he will go to any length to get a system with high computation power to break it within a reasonable time.

Furthermore, in a lecture notes on encryption by Shafi and Mihir (2008), they argued that in modern cryptography, we speak of the infeasibility of breaking the encryption system and computing information about exchanged messages whereas historically one spoke of the impossibility of breaking the encryption system and finding information about exchanged messages. They noted that the encryption systems which they described and claimed "secure" with respect to the new adversary are "not secure" with respect to a computationally unbounded adversary.

C. Third Party System Insecurity Factor

According to Cloud Security Alliance (2016), the digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to trust that signatures or assertions made by the private key (that correspond to the public key) are certified. In this model, a certificate authority (CA) is a trusted third party that is trusted by both the subject (owner) of the certificate and the party relying upon the certificate. CAs are characteristic of many public key infrastructure (PKI) schemes. If a central authority ensures that a unique certificate is assigned to each entity in a system, then an attacker cannot fake multiple identities. A trusted certificate is the only reliable method to defend against Sybil attacks.

Ordinarily, Public Key Infrastructure (PKI) is employed to solve the challenge of trust that the public key used in communications with a person really is the public key of that person. Normally, PKI consists of a trusted third party. The challenge is how to be sure the PKI system itself is not vulnerable. If the system itself is vulnerable it means the information on it such as the keys managed on the system are not secure.

2.3 Quantum Information and Cryptography

Quantum cryptography is the art and science of exploiting quantum mechanical effects in order to perform cryptographic tasks (Anne & Christian, 2015). One of the primitives is QKD. The relationship between quantum information and cryptography is almost half-a-century old (Anne & Christian, 2015) Still today, the two areas are closely intertwined: for instance, two of the most well-known results in quantum information stand out as being related to cryptography: quantum key distribution (QKD) (Bennett & Brassard, 1984) and Shor's factoring algorithm (Shor, 1994).

The predictions of quantum mechanics defy our everyday intuition and are partly responsible for the bewildering possibilities in the quantum world (Anne & Christian, 2015). The functionality is beyond what classical physics can offer: since any digital record can be copied, classical information simply cannot be used for uncloneability. Concepts such as;

- superposition - a particle can be in multiple places or states at the same time,
- entanglement - particles are correlated beyond what is possible classically and so the state cannot be described individually and
- quantum uncertainty -observing one property of a particle intrinsically degrades the possibility of observing another

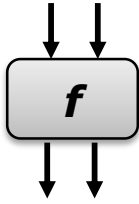
2.4 Quantum Information Representation

Quantum cryptosystem takes advantages of full parallelism provided by superposition quality of quantum system as illustrated in equation 2.1-2.3. The system evaluates entire function in a single run as shown in Figure 1. A qubit (or QUantum BIT) is similar in concept to a standard 'bit' - it is a memory element. It can hold not only the states '0' and '1' but a linear superposition of both states, $\alpha|0\rangle + \beta|1\rangle$. Therefore, the system takes advantage of parallelism architecture to process information faster than conventional computers.

$$00, 01, 10, 11 \tag{2.1}$$

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle \tag{2.2}$$

$$|\Psi\rangle = \sum_{i_1 i_2 \dots i_N} c_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle \quad (2.3)$$

$$\frac{1}{2} [|00\rangle + |01\rangle + |10\rangle + |11\rangle]$$


$$\frac{1}{2} [f(00)|00\rangle + f(01)|01\rangle + f(10)|10\rangle + f(11)|11\rangle]$$

Figure 1: Quantum parallelism. All functions evaluations in a single run

A. Quantum Key Distribution (QKD)

Quantum key distribution (QKD) is a primitive quantum cryptography (QC) that offers unconditional security between two remote parties that employ one-time pad encryption to encrypt and decrypt messages using a shared secret key, even in the presence of an eavesdropper with infinite computing power and mathematical genius (Mario et. al, 2017). While quantum computing remains speculative, QKD systems have already been realized in several commercial and research settings worldwide (Mario et. al, 2017).

A QKD link obviously requires the existence of quantum computer or detector, (Sheila, 2011), communication protocol, a quantum channel, used to transmit the prepared qubits, but also needs a classical authenticated channel to extract a secret key from a set of raw detections (Jesus et. al., 2013). Figure1 illustrates a notional QKD system configured to securely generate the secure shared key K, which is used to encrypt/decrypt sensitive data, voice, or video communications (Logan et.al, 2017). The QKD system consists of a sender “Alice”, a receiver “Bob”, a quantum channel (i.e., an optical fiber or direct line of sight free space path), and a classical channel (i.e., a conventional networked connection). Alice is shown with a laser source configured to generate and prepare single photons, known as quantum bits or “qubits”. The encoded photons are then transmitted over the quantum channel to Bob, whom measures them using specialized single photon detectors.

B. Conjugate Coding

It is also referred to as quantum coding. It is based on the principle that we can encode classical information into conjugate quantum bases. This primitive is extremely important in quantum cryptography in fact, the vast majority of quantum cryptographic protocols exploit conjugate coding in one way or another. The significance of conjugate coding to cryptography is summarized (Anne and Christian, 2015) by two key features that were, remarkably, already mentioned and exploited in Wiesner (1983):

- Measuring in one basis irrevocably destroys any information about the encoding in its conjugate basis.
- The originator of the quantum encoding can verify its authenticity; however, without knowledge of the encoding basis, and given access to a single encoded state, no third party can create two quantum states that pass this verification procedure with high probability.

The most well-known QKD protocols are the prepare-and-measure-based Bennett-Brassard-84 (BB84) and Bennett-92 (B92) protocols and the entanglement based Ekert-91 (E91) protocol described in (Sheila, 2011). Figure 2 shows conjugate coding and key distribution, using BB84 protocol, with the applications of the two key features summarized above.

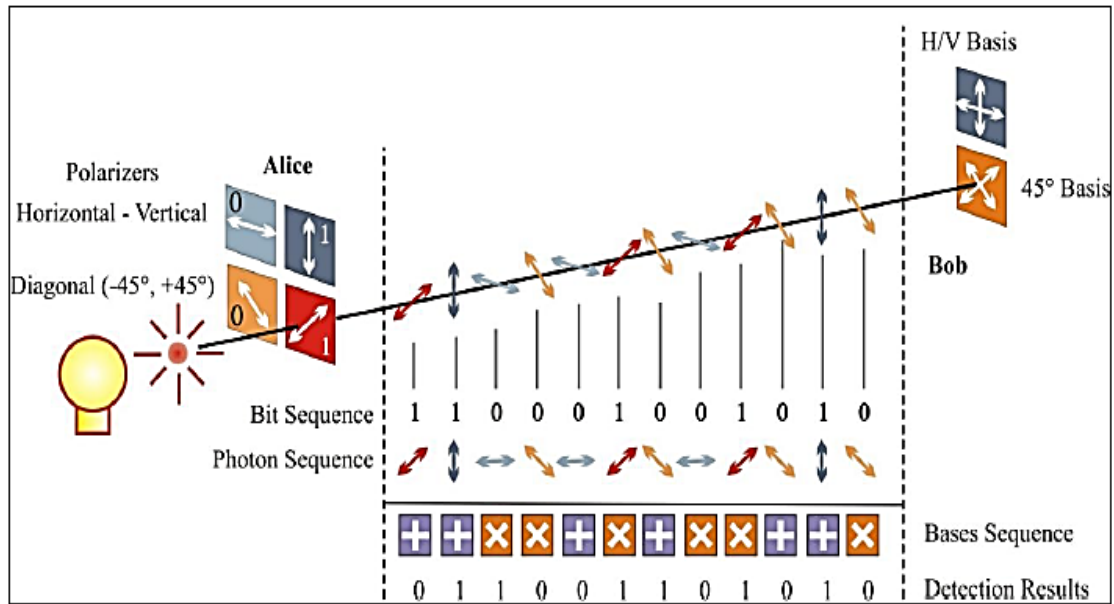


Figure 2: Raw key exchange using BB84 protocol
Source: Sheila, 2011

III. RESEARCH METHOD

The research survey used cover some mathematical formalism of quantum information as it pertains to quantum cryptography. The plan is aimed at showing the limits to traditional cryptosystem and then we can prove the strengths of quantum cryptography taking the candidates of one-way function as case study. For the purpose of the work, effort is concentrated on analyzing algorithms which is fundamental to cryptosystem. Today’s public key exchange schemes such as Diffie-Hellman and encryption algorithms like RSA respectively rely on the computational hardness of solving the discrete log problem and factoring large primes. We therefore make attempt to survey these cryptographic algorithms and show their run time complexity on both classic and quantum computing.

Algorithms for QKD

Quantum computers are particularly efficient at finding hidden cyclic subgroups in key spaces; using Shor’s algorithm (Medline, 2015) and this significantly reduce the time taken to factor large numbers, for example. The enhanced parallel processing potential and the ability to use “quantum methods” will reduce the time needed to solve the hard mathematical problems underlying the security of RSA, ElGamal algorithms, and their equivalent Elliptic Curve versions.

A. The Quantum Fourier Transform (QFT)

Linear complexity plays a very important role in the theory of stream ciphers so it is not surprising that the Discrete Fourier Transform (DFT) has been applied to problems in advanced cryptology. DFT is an algorithm that takes a signal and determines the frequency content of the signal (sequence of numbers). The equation for the DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi kn/N} \tag{3.1}$$

where we sweep k from 0 to N-1 to calculate all the DFT coefficients. When we say 'coefficient' we mean the values of X(k), so X(0) is the first coefficient, X(1) is the second etc. It is calculating the correlation between a signal x(n) and a function $e^{-i2\pi kn/N}$.

Using the following identity

$$e^{-i\theta} = \cos\theta - i\sin\theta \tag{3.2}$$

With a little bit of algebra we can turn it into this:

$$X(k) = \sum_{n=0}^{N-1} x(n)\cos\left(\frac{2\pi kn}{N}\right) - i \sum_{n=0}^{N-1} x(n)\sin\left(\frac{2\pi kn}{N}\right) \tag{3.3}$$

Quantum Fourier Transform (QFT) is exactly analogous to the classical Discrete Fourier Transform (DFT), which is typically used to examine the periodicity and other properties of an n-component vector of complex numbers (Madeline,

2015). For a given input vector $\alpha = (\alpha_1, \alpha_2 \dots \alpha_{n-1})$ and the resultant output $\beta = (\beta_1, \beta_2 \dots \beta_{n-1})$ of the DFT is specified by (3.4)

$$\beta_y \equiv \frac{1}{\sqrt{N}} \sum_{x=0}^{n-1} \alpha_x \exp\left(2\pi i \frac{xy}{n}\right) \quad (3.4)$$

Similarly, the QFT, whose gate is denoted U_{DFT} , linearly operates on an arbitrary $N = \log(n)$ -qubit quantum state $|\psi\rangle_{in} = \sum_{x=0}^{n-1} \alpha_x |x\rangle$

$$U_{DFT} |\psi\rangle = \sum_{x=0}^{n-1} \alpha_y |y\rangle \quad (3.5)$$

where the α_y are the discrete Fourier transforms of the α_x .

QFT can be solved in polynomial time (Mario et. al, 2017) by applying Shor’s algorithm on a quantum computer (Madeline, 2015, Adrian, 2015). The problem of finding the period r of the function $f(a) \equiv x^a \pmod{n}$, where x is any integer that is coprime (sharing no common factors) with n . It is possible to show that once r is found, the factors of n can be computed in polynomial time as $\gcd(x^{r/2} + 1)$ and $\gcd(x^{r/2} - 1)$.

The QFT can be implemented in $O(N \log N + \log 1/\epsilon)$, and all measurement steps are constant in time. The algorithm therefore has an overall complexity $\epsilon O(N^3)$. Classical implementations of the DFT, such as the fast Fourier transform algorithm, require $O(N^2)$ gates. In contrast, the QFT can be implemented as an $O(N^2)$ algorithm.

B. Large Integer Factorization

The most basic primitive for cryptographic applications is a one-way function which is “easy” to compute but “hard” to invert. Several candidates which seem to posse the above properties have been proposed such as Discrete Logarithm Problem (DLP), Finite Field Discrete Logarithm Problem (FFDLP) etc

The most obvious way of cracking RSA is to factor a user’s $n = pq$ into the primes p and q . When we talk about the problem of factoring, we assume that we are looking for a single non-trivial factor of a number n , so we can assume n is odd. Factoring: The function $f: (x; y) \rightarrow xy$ is conjectured to be a one way function. The asymptotically proven fastest factoring algorithms to date are variations on Dixon’s random squares algorithm. It is a randomized algorithm with running time

$$L(n)^{\sqrt{2}} \quad (3.6)$$

$$\text{where } L(n) = e^{\sqrt{\log n \log \log n}}$$

The number field sieve by Lenstra, Lenstra, Manasee, and Pollard with modifications by Adleman and Pomerance is a factoring algorithm proved under a certain set of assumptions to factor integers in expected time

$$e^{((c+o(1))(\log n)^{\frac{1}{2}}(\log \log n)^{\frac{2}{3}})} \quad (3.7)$$

Example of Shor’s algorithm for Prime factorization problem, given an integer N , find its prime factors

$$15 = 3 \times 5 \quad 15 = 3 \times 5$$

$$9999999942014077477 = 3162277633 \times 3162277669$$

Time to solve:

$$\text{Classical} \quad - \quad 2^{O[\ln(N)^{\frac{1}{3}}]} \quad (3.8)$$

$$\text{Quantum} \quad - \quad O[\ln(N)^2] \quad (3.9)$$

All algorithms proposed for factoring are intractable, for example Trial Division, Fermat Factorization, Pollard rho Factorization, Brent’s Factorization Method, Pollard p-1 Factorization

C. The Discrete Logarithm Problem (DLP)

The DLP is the problem of finding a such that $g^a \equiv x \pmod{p}$, where g is a generator of the multiplicative group modulo a prime p .

The function $f: (p, g, x) \rightarrow (g^x \pmod{p}, q, x)$ where p is a prime and g is a generator for Z_p^* is conjectured to be a one-way function. Computing $f(p, g, x)$ can be done in polynomial time using repeated squaring. However, the fastest known proved solution for its inverse, called the discrete log problem is the index-calculus algorithm, with expected running

$\text{time}L(n)^{\sqrt{2}}$. An interesting problem is to find an algorithm which will generate a prime p and a generator g for Z_p^* in polynomial time.

The algorithm requires 3 quantum registers of length N such that $p \leq 2^N < 2p$, each initialized to the state 0. The first step is to apply the $H^{\otimes N}$ gate to the first two registers (again to take advantage of parallel computation) in order to put the system in the global state

Next, the third register is used to compute the modular exponent $f(y, z) = g^y x^z \pmod{p}$, We now apply the inverse of the QFT gate onto the first two registers, at which point the state becomes approximately

$$|\psi\rangle \rightarrow \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \left| \frac{al}{r} \right\rangle |l/r\rangle |\hat{\chi}(al, l)\rangle \quad (3.10)$$

With this algorithm, the DLP can therefore be solved in polynomial time.

D. Finite Field Discrete Logarithm Problem (FFLDP)

Let F_q be a finite field. Let g generate F_q^* . Let $b \in F_q^*$. Then $g^i = b$ for some positive integer $i \leq q - 1$. Determining i given F_q, g and b is the finite field discrete logarithm problem (FFDLP), which is (to our current knowledge) as hard as factoring (Edward, 2010).

Let us take for example, 2 generates F_{101}^* . So we know $2^i = 3$ (*i.e.* $2^i \equiv 3 \pmod{101}$) has a solution. It is $i = 69$. Similarly, we know $2^i = 5$ has a solution; it is $i = 24$. How can we solve such problems faster than brute force? Some solutions were presented in (Edward, 2010) faster than brute force. But they are nonetheless not fast.

For cryptographic purposes, consider $10^{300} < q < 10^{600}$ where q is a (large) prime or of the form 2^d . Notation, if $g^i = b$ then we write $\log_g(b) = i$. In the above example, for $q = 101$ we have $\log_2(3) = 69$ (*Since* $e^{69} \equiv 3 \pmod{101}$).

The best known algorithms for solving the FFLDP take as long as those for factoring, and so are sub-exponential.

In Quantum computing, the algorithm requires 3 quantum registers of length N such that $p \leq 2^N < 2p$, each initialized to the state 0. The first step is to apply the $H^{\otimes N}$ gate to the first two registers (again to take advantage of parallel computation) in order to put the system in the global state such as

$$|\psi\rangle = \frac{1}{2^N} \sum_{y=0}^{2^N-1} \sum_{z=0}^{2^N-1} |y\rangle |z\rangle |0\rangle \quad (3.11)$$

Next, the third register is used to compute the modular exponent $f(y, z) = g^y x^z \pmod{p}$, by applying the inverse of the QFT gate onto the first two registers, at which point the state becomes approximately.

$$|\psi\rangle \rightarrow \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} \left| \frac{al}{r} \right\rangle |l/r\rangle |\hat{\chi}(al, l)\rangle \quad (3.12)$$

Measuring the first two registers then gives values for $al=r$ and $l=r$ modulo p (where l is variable), from which the desired discrete logarithm a may be deduced by applying a continued fractions algorithm (Madeline, 2015)

IV. ALGORITHMS EVALUATION

We proceed to present the result of the comparison of runtime complexity of the algorithms on both classical and quantum cryptography. Our comparison starts with information representation between the both schemes then the results follow. The results show that the algorithms, most especially the candidates of one-way function, take polynomial time with quantum cryptography while they are intractable with classical counterpart.

Table 4.1: Summary of run time complexity of Classical and Quantum Cryptography

	Classical Cryptosystem	Quantum Cryptosystem
Long Integer Factorization	$e^{\sqrt{\log n \log \log n}} \sqrt{2}$ $2^{O\left[\ln(N)^{\frac{1}{3}}\right]}$	$O(N^3)$ $O[\ln(N)^2]$
Fourier Transform	$O(N^{2^N})$	$\epsilon O(N^2), O(N^2)$
Finding Discrete Logarithms	$e^{((c+o(1))(\log n)^{\frac{1}{2}}(\log \log n)^{\frac{2}{3}})}$	$O(N^3)$

V. DISCUSSION

The implementation of the algorithms proved the breaking of traditional cryptography which are intractable and infeasible to implement on classical computer due to its limited computing power. Intractable algorithms on classical cryptosystem are now easy to manage on quantum cryptosystem since they run in polynomial time. Such algorithms are fundamental to cryptography, therefore reducing their time complexity indicates that any encryption scheme based on them will be easily broken. These algorithms are all factoring-based problems such as Finite Field Discrete Logarithm Problem (FFLDP), Large Integer Factorization Problem (LIFP), and Discrete Fourier Transform (DFT) which are implemented in scheme like DES and RSA.

VI. CONCLUSION

Quantum computer breaks today's cryptosystem through superposition quality; ability to take full advantage of parallelism to solving factoring-based problems which are fundamental to modern cryptographic techniques. Similarly, quantum key distribution (QKD) secures privacy and confidentiality of information even in the presence of adversary using entanglement quality such that tampering with it would be known quickly. Asymmetric cryptography as we know it now will be no more, and different asymmetric schemes will have to be devised. Symmetric cryptography will need to use keys which are twice as long as now, to achieve the same level of security. Quantum computers therefore force cryptography into a new age.

Quantum cryptography is still in its infancy and so far looks very promising. This technology has the potential to make a valuable contribution to e-commerce and business security, personal security, and security among government organizations. If quantum cryptography turns out to eventually meet even some of its expectations, it will have a profound and revolutionary effect on all of our lives.

REFERENCES

- [1] Shor P. W., (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.
- [2] Logan O. M., Michael R. G., Douglas D. H., Ryan E., Colin M. and Gerald B. (2017). Modeling, Simulation, and Performance Analysis of Decoy State Enabled Quantum Key Distribution Systems. *Applied Science Article*
- [3] Mario M., Chen F., Lei M. Z., Glenn G. P., (2017). Key Reconciliation with Low-Density Parity-Check Codes for Long-Distance Quantum Cryptography. *arXiv:1702.07740v2 [quant-ph] 17*
- [4] Martinez-Mateo J., David E. & Vicente M., (2013). Key Reconciliation for High Performance Quantum Key Distribution. *Scientific Reports 3 : 1576 - DOI: 10.1038/srep01576*
- [5] Shafi G. & Mihir B. (2008). *Lecture Notes on Cryptography. MIT Computer Science and Artificial Intelligence Laboratory, Cambridge & Department of Computer Science and Engineering, University of California*
- [6] Mohannad M. A. (2013). Research Methodologies in Computer Science and Information Systems. *Alquds Open University College of Technology & Applied Science*
- [7] Asmoredjo S.M. (2005). Masters' Thesis on a Probabilistic Model for Cyber Attacks and Protection
- [8] Edward S. (2010). *A Lecture Note On "An introduction to cryptography and cryptanalysis". Santa Clara University*
- [9] Sheila C. (2011). Quantum Key Distribution Protocols and Applications. *Technical Report RHUL. Department of Mathematics Royal Holloway, University of London Egham, Surrey TW20 0EX, England*
- [10] Madeline M. (2015). A Review of the Development and Current Status of Quantum Key Distribution. *A project work submitted to Pomona College Department of Physics and Astronomy.*
- [11] Deborah C., Paul E., Firdaus A. & Ian H. (2005). Commercial Prospects for Quantum Information Processing. *Quantum Information Processing Interdisciplinary Research Collaboration*

- [12] Gobby C., Yuan Z. L. & Shields A. J. (2004). Unconditionally secure quantum key distribution over 50km of standard telecom fibre. *Toshiba Research Europe Ltd, Cambridge Research Laboratory Cambridge, CB4 0WE, UK*
- Michael E. W. & Herbert J. M. (2012). Principles of Information Security, Fourth Edition, *Course Technology, Cengage Learning. ISBN-13: 978-1-111-13821-9*
- [13] Bennett C. and Brassard G., (1984). Quantum Cryptography: Public Key Distribution and Coin Tossing. *International Conference on Computers, Systems, and Signal Processing, Bangalore, India.*
- [14] Lütkenhaus N. (1999). Estimates for practical quantum cryptography. *Physical Review A 59.5:3301.*

