A Matlab Realization of Shor's Quantum Factoring Algorithm

S. Jha, P. Chatterjee, A.Falor and M. Chakraborty, *Member IEEE*
Department of Information Technology
Institute of Engineering & Management
Kolkata, India
email: mohuyacb@yahoo.com

Abstract— **Quantum cryptography or quantum key distribution uses quantum mechanics to guarantee secure communication. It enables two parties to produce a shared random bit string known only to them, which can be used as a key to encrypt and decrypt messages. An important and unique property of quantum cryptography is the ability of the two communicating users to detect the presence of any third party trying to gain knowledge of the key. A third party trying to eavesdrop on the key must in some way measure it, thus introducing detectable anomalies. By using quantum superposition or quantum entanglement and transmitting information in quantum states, a communication system can be implemented which detects eavesdropping. If the level of eavesdropping is below a certain threshold a key can be produced which is guaranteed as secure, otherwise no secure key is possible and communication is aborted. In this paper authors have realized Shor's Quantum Factoring Algorithm using Matlab. Simulation results using quantum bits verify the possible eavesdropper's presence, changing the state of the system and hence disrupting the whole process.**

***Index Terms*—Classical computing, quantum computing, quantum cryptography, photon, polarization, quantum key exchange, Shor' algorithm.**

I. Introduction

Quantum Computing is a new area of research that has only recently started to emerge. Quantum Computing and Quantum Cryptography were born out of the study of how quantum principles might be used in performing computations. In 1982 Richard Feynman, the Nobel Laureate observed that certain quantum mechanical phenomena could be simulated efficiently on a classical computer. He suggested that using quantum mechanics to do computations that are impossible on classical computers could perhaps reverse the simulation. Feynman did not present any examples of such devices, and only recently has there been progress in constructing even small versions. Quantum cryptography is used solely to produce and distribute a key and not to transmit any message data. This key can then be used with any chosen encryption algorithm to encrypt (and decrypt) a message, which can then be transmitted over a standard communication channel [1].

The algorithm most commonly associated with quantum key distribution (QKD) for symmetric cryptosystems is the one-time pad, as it is probably secure when used with a secret random key. This cryptosystem consists of creating a secret-key of random bits the same length as the plaintext. The secret-key is then modularly added bit by bit to the plaintext to produce the ciphertext. The one-time pad however suffers three major draw-backs: the secret-key must be as long as the plaintext, the secret-key must be discarded after each use, and the secret-key must be truly randomly generated [2].

Asymmetric cryptography schemes are based on the concept of one-way functions, which are mathematically non-invertible functions. Encryption and decryption keys are different unlike symmetric cryptosystems, which use a single key for both encryption and decryption [3].

Symmetric quantum cryptosystems while having the potential for a perfectly secure system currently are plagued with a variety of technical problems as discussed above. Comparing this to an asymmetric cryptosystem (classical or quantum), at least three times as much information must be sent, and hence a decrease in the security of the system.

Many current asymmetric cryptosystems depend upon classical intractable functions (such as the factoring of large integers with two prime factors in the case of RSA encryption [4]) that have been shown to be solvable in polynomial time on a quantum computer and hence are not quantum intractable. Because of this, quantum intractable functions must be found for quantum asymmetric cryptosystems to be secure. Currently very few quantum intractable functions have been proposed, and those that have are based upon a quantum algorithm proposed by Peter Shor of MIT [5] in 1994 for the factorization of large integers with two coprimes.

The organization of the paper is as follows. After the introduction in section I, an overview of quantum key exchange is provided in section II. This is followed by the

mathematical aspects (both classical and quantum) of Shor's algorithm in section III. Section IV provides the simulation environment and results. The paper is concluded in section V with some highlights on future works.

II. Quantum Key Exchange

Quantum communication involves encoding information in quantum states, or qubits, as opposed to classical communications use of bits. Usually, photons are used for these quantum states. Quantum cryptography exploits certain properties of these quantum states to ensure its security. There are several different approaches to quantum key distribution, but they can be divided into two main categories depending on which property they exploit [2].

A. Prepare and Measure Protocols

Unlike in classical physics, the act of measurement is an integral part of quantum mechanics. In general, measuring an unknown quantum state will change that state in some way. This is known as quantum indeterminacy, and underlies results such as the Heisenberg Uncertainty Principle, Information-Disturbance Theorem and No Cloning Theorem [6]. This can be exploited in order to detect any eavesdropping on communication (which necessarily will involve measurement), and more importantly calculate the amount of information which has been intercepted [7].

B. Entanglement Based Protocols

The quantum states of two (or more) separate objects can become linked together in such a way that they must be described by a combined quantum state, not as individual objects. This is known as Entanglement and means, for example, performing a measurement on one object will affect the other. If an entangled pair of objects is shared between two parties, anyone intercepting either particle will alter the overall system, allowing their presence (and the amount of information they have gained) to be determined [7].

These two approaches can both be further subdivided into three families of protocols; discrete variable, continuous variable and distributed phase reference coding. Discrete variable protocols were the first to be invented, and they remain the most widely implemented. The other two families are mainly concerned with overcoming practical limitations of experiments. Described below are the two protocols that use discrete variable coding.

C. Photon Polarization States

Any two pairs of conjugate states can be used for the protocol, and many optical fibre based implementations use phase encoded states. The sender (traditionally referred to as Alice) and the receiver (Bob) are connected by a quantum communication channel which allows quantum states to be transmitted. In the case of photons this channel is generally either an optical fibre or simply free space. In addition they communicate via a public classical channel, for example using radio waves or the internet. Neither of these channels need to be secure; the protocol is designed with the assumption that an eavesdropper (referred to as Eve) can interfere in any way with both. The security of the protocol comes from encoding the information in non-orthogonal states. Quantum indeterminacy means that these states cannot in general be measured without disturbing the original state (see No cloning theorem) [6]. There are two pairs of states, with each pair conjugate to the other pair, and the two states within a pair orthogonal to each other. Pairs of orthogonal states are referred to as a basis. The usual polarization state pairs used are either the rectilinear basis of vertical (0°) and horizontal (90°), the diagonal basis of 45° and 135° or the circular basis of left- and right-handedness. Any two of these bases are conjugate to each other, and so any two can be used in the protocol. Fig. 1 shows the rectilinear and diagonal bases that are used.

Basis	0	1
+	↑	→
X	↗	↘

Fig. 1. Rectilinear and diagonal bases

The first step is quantum transmission. Alice creates a random bit (0 or 1) and then randomly selects one of her two bases (rectilinear or diagonal in this case) to transmit it in. She then prepares a photon polarization state depending both on the bit value and basis, as shown in the table to the left. So for example a 0 is encoded in the rectilinear basis (+) as a vertical polarization state, and a 1 is encoded in the diagonal basis (x) as a 135° state. Alice then transmits a single photon in the state specified to Bob, using the quantum channel. This process is then repeated from the random bit stage, with Alice recording the state, basis and time of each photon sent.

Quantum mechanics (particularly quantum indeterminacy) says there is no possible measurement that will distinguish between the 4 different polarization states, as they are not all orthogonal. The only measurement possible is between any two orthogonal states (a basis), so for example measuring in the rectilinear basis will give a result of horizontal or vertical. If the photon was created as horizontal or vertical (as a rectilinear eigenstate) then this will measure the correct state, but if it was created as 45° or 135° (diagonal eigenstates) then the rectilinear measurement will instead return either horizontal or vertical at random. Furthermore, after this measurement the photon will be polarized in the state it was measured in (horizontal or vertical), with all information about its initial polarization lost [8].

As Bob does not know the basis the photons were encoded in, all he can do is select a basis at random to measure in, either rectilinear or diagonal. He does this for each photon he receives, recording the time, measurement basis used and measurement result. After Bob has measured all the photons, he communicates with Alice over the public classical channel. Alice broadcasts the basis each photon was sent in, and Bob the basis each was measured in. They both discard photon measurements (bits) where Bob used a different basis, which will be half on average, leaving half the bits as a shared key as shown in Table I.

TABLE I
PHOTON POLARIZATION STATES PROTOCOL

Alice's random bit	0	1	1	0	1	0	0	1
Alice's random sending basis	+	+	X	+	X	X	X	+
Photon polarization Alice sends	↑	→	↘	↑	↘	↗	↗	→
Bob's random measuring basis	+	X	X	X	+	X	+	+
Photon polarization Bob measures	↑	↗	↘	↗	→	↗	→	→
Shared secret key	0		1			0		1

D. Intercept and Resend

To check for the presence of eavesdropping Alice and Bob now compare a certain subset of their remaining bit strings. If a third party has gained any information about the photons polarization it will have introduced errors in Bobs measurements. If more than p bits differ they abort the key and try again, possibly with a different quantum channel, as the security of the key cannot be guaranteed. p is chosen so that if the number of bits known to Eve is less than this, privacy amplification can be used to reduce Eve's knowledge of the key to an arbitrarily small amount, by reducing the length of the key.

The simplest type of possible attack is the intercept-resend attack, where Eve measures the quantum states (photons) sent by Alice and then sends replacement states to Bob prepared in the state she measures [9]. This will produce errors in the key shared between Alice and Bob. As Eve has no knowledge of the basis a state sent by Alice is encoded in, she can only guess which basis to measure in, in the same way as Bob. If she chooses correctly then she will measure the correct photon polarization state as sent by Alice, and will resend the correct state to Bob. However if she chooses incorrectly then the state she measures will be random, and the state sent to Bob will not be the same as the state sent by Alice. If Bob then measures this state in the same basis Alice sent he will get a random result, as Eve has sent him a state in the opposite basis, instead of the correct result he would get without the presence of Eve as shown in Table 2.

TABLE II
INTERCEPT AND RESEND PROTOCOL

Alice's random bit	0	1	1	0	1	0	0	1
Alice's random sending basis	+	+	X	+	X	X	X	+
Photon polarization Alice sends	↑	→	↘	↑	↘	↗	↗	→
Eve's random measuring basis	+	X	+	+	X	+	X	+
Polarization Eve measures and sends	↑	↗	→	↑	↘	→	↗	→
Bob's random measuring basis	+	X	X	X	+	X	+	+
Photon polarization Bob measures	↑	↗	↗	↘	→	↗	↑	→
Shared secret key	0		0			0		1
Errors in key	✓		✘			✓		✓

III. SHOR'S ALGORITHM

A. Overview

Shor's algorithm, named after mathematician Peter Shor, is a quantum algorithm (an algorithm which runs on a quantum computer) for integer factorization discovered in 1994 [5]. Informally it solves the following problem: Given an integer n, the algorithm finds the prime factors of n.

The algorithm was viewed as important because of the difficulty of factoring large numbers, which is used in most

cryptography systems like RSA [4]. If an efficient method of factoring large numbers is implemented most of the current encryption schemes would become worthless. While it has not been proven that factoring large numbers can not be achieved on a classical computer in polynomial time, the fastest algorithm publicly available for factoring a large number *n* (whose representation has *log n* bits) runs in:

$O(\exp((\log n)^{1/3} * (\log \log n)^{2/3}))$, or exponential time. In contrast Shor's algorithm runs in $O((\log n)^2 * (\log \log n))$ on a quantum computer, and then must perform $O(\log n)$ steps of post processing on a classical computer. Overall then this time is polynomial. This discovery propelled the study of quantum computing forward and as such an algorithm is much sought after. The effectiveness of the algorithm lies in the efficiency of the quantum Fourier transform, and modular exponentiation by squaring.

Shor's algorithm is important because it can, using a quantum computer, be used to break the widely used public-key cryptography scheme known as RSA. RSA is based on the assumption that factoring large numbers is computationally infeasible. So far as is known, this assumption is valid for classical (non-quantum) computers; no classical algorithm is known that can factor in polynomial time. However, Shor's algorithm shows that factoring is efficient on a quantum computer, so an appropriately large quantum computer can break RSA. It was also a powerful motivator for the design and construction of quantum computers and for the study of new quantum computer algorithms. It has also facilitated research on new cryptosystems that are secure from quantum computers, collectively called post-quantum cryptography [10].

B. Procedure

The problem to be solved is: given a composite number *n*, find an integer *p*, strictly between *1* and *n* that divides *n*. Shor's algorithm consists of two parts:

- A reduction, which can be done on a classical computer, of the factoring problem to the problem of order-finding.
- A quantum algorithm to solve the order-finding problem.

Classical Part

1. Pick a pseudo-random number $a<N$.
2. Compute $gcd(a, N)$. This may be done using the Euclidean algorithm.
3. If $gcd(a, N) \neq 1$, then there is a nontrivial factor of *N*. Stop.
4. Otherwise, use the period-finding subroutine (below) to find *r*, the period of the following function:
   $f(x) = a^x \bmod N$,
   i.e. the smallest integer *r* for which $f(x + r) = f(x)$.
5. If *r* is odd, go back to step 1.
6. If $a^{r/2} \equiv -1 \pmod N$, go back to step 1.
7. The factors of *N* are $gcd(a^{r/2} \pm 1, N)$. Stop.

Quantum Part

1. Start with a pair of input and output qubit registers with $log_2 N$ qubits each, and initialize them to
   $$N^{-1/2} \sum_x |x\rangle |0\rangle$$
   where *x* runs from 0 to N - 1.
2. Construct $f(x)$ as a quantum function and apply it to the above state, to obtain
   $$N^{-1/2} \sum_x |x\rangle |f(x)\rangle$$
3. Apply the quantum Fourier transform on the input register. The quantum Fourier transform on *N* points is defined by:
   $$U_{QFT} |x\rangle = N^{-1/2} \sum_y e^{2\pi i x y/N} |y\rangle$$
   This leaves us in the following state:
   $$N^{-1} \sum_x \sum_y e^{2\pi i x y/N} |y\rangle |f(x)\rangle$$
4. Perform a measurement. We obtain some outcome *y* in the input register and $f(x0)$ in the output register. Since *f* is periodic, the probability to measure some *y* is given by
   $$N^{-1} \left| \sum_{x:\, f(x)=f(x_0)} e^{2\pi i x y/N} \right|^2 = N^{-1} \left| \sum_b e^{2\pi i (x_0 + rb) y/N} \right|^2$$
   Analysis now shows that this probability is higher, the closer yr/N is to an integer.
5. Turn y/N into an irreducible fraction, and extract the denominator *r'*, which is a candidate for *r*.
6. $f(x) = f(x + r')$. Stop.

IV. Simulation Environment and Results

A. Simulation Environment

Although experimental demonstration of Shor's algorithm is important for the study of quantum computers, however it has proved to be elusive [11]. In this paper we present a realization of the simplest instance of Shor's algorithm using Matlab Version: 7.5.0.342(R2007b). There are four restrictions for Shor's algorithm. They are:

1. The number to be factored must be >= 15
2. The number to be factored must be odd
3. The number must not be prime
4. The number must not be a prime power

B. Results

The results of Matlab simulation of Shor's algorithm is presented below. Fig. 2 shows the simulation results of classical part of the algorithm when all the above conditions are met for N = 63. Fig. 3 shows the simulation results of quantum part of the algorithm. Fig. 3a, 3b, 3c, 3d and 3e show the plots of steps 1, 2, 3, 4 and situation when the information is intercepted by eavesdropper and no outcome is obtained (encircled in Fig. 3e) respectively.

```
Command Window
>> start()
         WELCOME TO THE SIMULATION OF SHOR'S ALGORITHM
There are four restrictions for shor's algorithm:
I. The number to be factored must be >= 15.
II. The number to be factored must be odd.\
III. The number must not be prime.
IV. The number must not be a prime power.
Enter the number you want to factor:63
The number is not prime
Yes!!! The number can be factorised
The value of last generated number is :
      8

The value of period is :
      2

The factors are :
      9

      7

>>
```

Fig. 2. Simulation results of classical part of Shor's algorithm

```
Command Window
File  Edit  Debug  Desktop  Window  Help
>> quantum(3)

d2 =

    64    64


d3 =

    64    64


ans =

 0.15468|> + -0.030824-0.10009i|1> + 0.0043079+0.10793i|10> +
0.17829+0.10995i|11> + 0.11937-0.056099i|100> + 0.10941-0.060345i

a =

     1


p =

     1

     1

>>
```

Fig. 3a. Input and output qubit register initialization

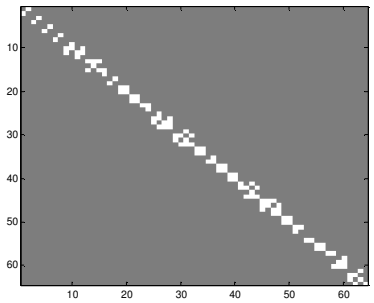

Fig. 3b. Absolute value of Quantum Function of qubits

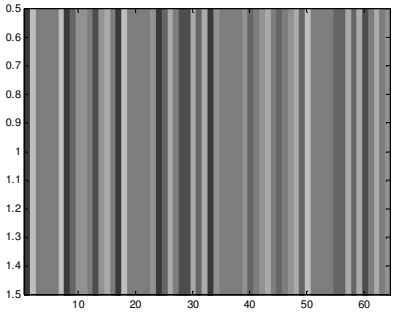

Fig. 3c. Absolute value of Quantum Fourier Transform of qubits

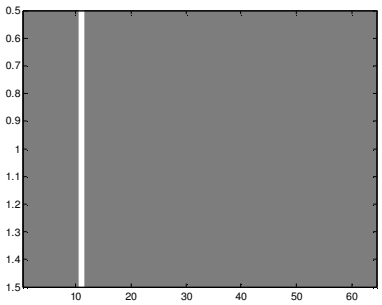

Fig. 3d. Output measurement

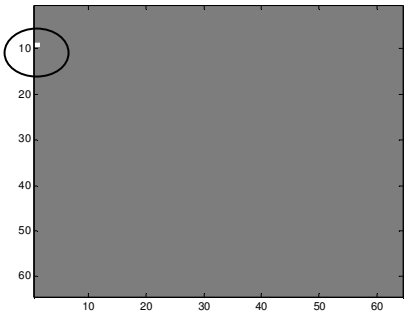

Fig. 3e. Information getting intercepted

Fig. 3. Simulation results of quantum part of Shor's algorithm

V. Conclusion

Quantum Computing is a new area of research that has very recently emerged. As explained Shor's quantum factoring algorithm is no doubt a very efficient way of factorizing a composite number and has a lot of advantages over classical approach with regard to time of calculation. However it is not devoid of limitations. The computers that we use are not enough to do huge quantum computations. We may not be able to implement this algorithm in case of large numbers. The processor strength should be much more than the classical computers we generally use.

The future prospect of quantum cryptography is huge. Taking into account the massive use of internet in the modern days, it is going to play a mammoth part in securing the

transfer of information over a network. This is only a new field of research and we are yet to see many more innovations coming our way in this area. There have been many modifications to Shor's algorithm. For example, whereas, an order of twenty to thirty runs are required on a quantum computer in the case of Shor's original algorithm, and with some of the other modifications, in the case of the modification done by David McAnally at the University of Queensland an order of only four to eight runs on the quantum computer is required [12]. Though this will need larger resources, strong processors, much more memory than the classical computers, still we can surely say that it is not impossible. We are about to witness a huge revolution in the field of network security.

REFERENCES

[1] Ranjan Bose, Information Theory, Coding and Crypotography, MH

[2] Trappe & Washington, Introduction to Cryptography with Coding Theory, 2/e, Pearson.

[3] Michael Welschenbach, Cryptography in C and C++ 2/e, Apress.

[4] Steve Burnett & Stefan Palne, RSA Security's Official Guide to Cryptography, RSA Press.

[5] Shor, P. (1994). "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *Proceedings of the 35th Symposium on Foundations of Computer Science, 35*, 124-134.

[6] William K. Wootters, Wojciech H. Zurek, "No Cloning Theorem", Physics Today, Feb 2009, pp 76-77.

[7] Mobin Javed, Khurram Aziz, "A Survey of Quantum Key Distribution Protocols", *Proceedings of the 6th International Conference on Frontiers of Information Technology FIT '09,* 2009.

[8] Jihang Ye Daqing Wang Yu Liu, "Remote Preparation of Photon Polarization State via Eintein-Podolsky-Rosen Channel", Proceedings of *Symposium on Photonics and Optoelectronics, 2009* during 14-16 Aug. 2009, pp- 1 – 4.

[9] Marcos Curty, Norbert Lütkenhaus, "Intercept-Resend Attacks in the Bennett-Brassard 1984 Quantum-Key-Distribution Protocol with Weak Coherent Pulses", Phys. Rev. A 71, 062301 (2005).

[10] Bernstein, Daniel J., Buchmann, Johannes, Dahmen, Erik (Eds.), "Post-Quantum Cryptography", 2009, IX, 245 p. 25 illus., Hardcover, Springer.

[11] Vandersypen, L. et al (2001). "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," Nature, 414, 883-887.

[12] David McAnally, "A Refinement of Shor's Algorithm", arXiv:quant-ph/0112055v4.