Unveiling the Power of Shor's Algorithm: Cryptography in a Post Quantum World

Dylan Phares

Follow this and additional works at: https://scholarship.claremont.edu/cmc_theses

Part of the Applied Mathematics Commons, and the Number Theory Commons

Recommended Citation
https://scholarship.claremont.edu/cmc_theses/3669

This Open Access Senior Thesis is brought to you by Scholarship@Claremont. It has been accepted for inclusion in this collection by an authorized administrator. For more information, please contact scholarship@claremont.edu.
Unveiling the Power of Shor’s Algorithm:
Cryptography in a Post Quantum World

submitted to

Professor Helen Wong

By
Dylan Russell Phares

for
Senior Thesis
2024
4/22/24
Acknowledgements

A brief thank you to Professor Wong for her technical help throughout.
Additionally thank you to my mom for her motivation and encouragement throughout.
## Contents

1 **Introduction** 3  
   1.1 Overview of Cryptography 3  
   1.2 The Need for Stronger Encryption Techniques 3  
   1.3 Introduction to Shor’s Algorithm 3  
   1.4 Thesis Statement 4  

2 **Fundamentals of Classic Cryptography** 5  
   2.1 Historical Development of Classical Cryptography 5  
   2.2 RSA Encryption Scheme 5  
   2.3 RSA Example 6  
   2.4 Decrypting using the period of message 7  
   2.5 Factoring $N$ using the period of message 8  

3 **Introduction to Quantum Computing** 9  
   3.1 Qubits, Superposition and Entanglement 9  
   3.2 Quantum Gates and Circuits 9  
   3.3 Quantum Measurement 10  

4 **Shor’s Algorithm: The Breakthrough** 12  
   4.1 Origin and Background of Shor’s Algorithm 12  
   4.2 Key Steps and Operations of Shor’s Algorithm 12  
   4.3 An In Depth Review of Shor’s Algorithm 14  
   4.4 Shor’s Algorithm Example 1 16  
   4.5 Shor’s Algorithm Example 2 19  

5 **Conclusion** 24  
   5.1 Recapitulation of Key Points 24  
   5.2 Implications for RSA Encryption 24  
   5.3 Future Outlook and Recommendations 25  
   5.4 Closing Remarks 25  

6 **Bibliography** 26
1 Introduction

1.1 Overview of Cryptography
Cryptography is the process of sending and receiving encoded messages with the goal being that only you and your pal will be able to decode the message. In typical Cryptography problems we have a sender Alice who sends an encoded message to a receiver Bob, however there is a third party, Eve, whose goal is to steal and then decode the encoded message. The goal of cryptography is to make this as hard as possible for Eve and to ensure that Alice and Bob’s messages remain safe.

1.2 The Need for Stronger Encryption Techniques
In the current landscape, cybersecurity and cryptography are hyper important in assuring sensitive information is safely concealed and secured. However current encryption techniques will not remain satisfactory forever. Even current methods are technically breakable but the computing power needed to solve these problems is not easy to get access too. What will change the cryptography landscape for good will be the creation of the first sufficiently large and hyper-efficient Quantum Computer. This would allow RSA encryption to be easily decrypted, as shown by Shor’s algorithm, which is the basis of most if not all cryptography used over the internet by people and companies alike. Most old cryptography techniques are likely be rendered useless. This is why some form of Quantum Cryptography needs to be discovered and successfully implemented. Classical cryptography will likely be phased out sooner rather than later. All we can hope for is that we are able to figure out Quantum Cryptography before someone or some entity is able to construct a sufficiently large Quantum Computer.

1.3 Introduction to Shor’s Algorithm
In 1994 Mathematician Peter Shor developed a Polynomial-time algorithm for factoring large integers on quantum computers. Factoring large numbers is what many cryptography schemes rely on, particularly RSA. By Polynomial-time all that means is that as the problem grows the time it takes to solve only increases polynomially rather than exponentially. So in this instance a typical classical computer would take $2^n$ time to compute whereas a quantum computer would take $n^2$. Where $n$ is the number of digits in the integer that needs to be factored. Which as $n$ increases the classical computer ends up computing for a much longer time. Despite the lack of quantum computers, Shor’s algorithm tells us that RSA can be decrypted with quantum. Shor’s algorithm is extremely important now and will have evergrowing impotence in the future as it proves that given a functioning Quantum Computer, factoring large composite numbers can be done in half the time or less when compared to classical computers.
1.4 Thesis Statement

This paper will discuss Shor’s algorithm in length and explain the process of and anything associated with completing the algorithm, including a brief overview of current cryptography techniques and some basics of Quantum computing. Understanding Shor’s algorithm can be an in depth and lengthy process due to the nature of the algorithm utilizing both quantum and classical techniques. However this task is not insurmountable. We will explore the ins and outs of Shor’s algorithms including how the algorithm works step by step and why all these steps are necessary to ensure the accuracy of the algorithm. The aim of this thesis is to educate and inform on Shor’s Algorithm in an easy to digest manner.
2 Fundamentals of Classic Cryptography

2.1 Historical Development of Classical Cryptography

Cryptography is almost as old as man, for as long as man has wanted to keep secrets from one another some form of cryptography has been utilized. Going back to the Greeks and Egyptians, they would utilize some form of encoded message to keep the message safe. However cryptography really began to accelerate first during World War 1 then seriously during World War 2. World War 1 still mainly had people manually encrypting and decrypting encoded messages. Then in World War 2 thanks to rotor machines these calculations could be done much faster and with much more accuracy.[3] Eventually the National Bureau of Standards realized that a standard encryption system would be extremely useful for businesses and governments to keep their information safe online. This led IBM to develop the Data Encryption Standard (DES) where two users would share a secret key with each other in order to decrypt. However this is extremely inefficient and is easily breakable by the discovery of the secret key. This in turn led to people trying to find a way to share keys without getting them stolen. This secure method is called the Diffie-Hellman key exchange and utilizes modular exponentiation. However more importantly Diffie came up with the idea of public key cryptography. Which in turn leads to the discovery of RSA encryption by Rivest, Shamir, and Adleman. However while these men are credited with the invention of RSA, it was actually being utilized years earlier by intelligence communities as they had discovered it in secret. [4]

2.2 RSA Encryption Scheme

RSA is unique to many other cryptographic techniques in that it employs a public key that is open to the world as well as a private key for the person who sends the message. The public key encodes the message while the private key decodes them. This is in contrast to Symmetric Key cryptography where both users agree on a certain decode key that both will know and use. Given both methods, the public key system is a much safer one to employ and is much less likely to be able to be decoded by a malicious third party attack. So here’s an in depth view of how RSA works:

1. **Key Generation:**

   1. - Select two large prime numbers, \( p \) and \( q \), typically of similar bit lengths.
   2. - Compute their product, \( (N = pq) \), which serves as the modulus for both the public and private keys.
   3. - Choose an exponent \( (e) \) relatively prime to \( ((p-1)(q-1)) \) and compute its modular multiplicative inverse \( (d) \) modulo \( ((p-1)(q-1)) \). \( (e) \) is part of the public key as the encryption exponent, while \( (d) \) is kept private as the decryption exponent.
2. **Encryption:**

1. To send a message \( m \), the sender obtains the recipient’s public key \((N, e)\).
2. The sender raises \( m \) to the power of \( e \) modulo \( N \) to obtain the ciphertext \( c \):
   \[
   (c \equiv m^e \pmod{N})
   \]
3. The ciphertext \( c \) is sent to the recipient.

3. **Decryption:**

1. The recipient, possessing the private key \((N, d)\), raises the ciphertext \( c \) to the power of \( d \) modulo \( N \) to recover the original message \( m \):
   \[
   (m \equiv c^d \pmod{N})
   \]

RSA’s security relies on the difficulty of factoring the modulus \( N \) into its prime factors \((p, q)\). Generally somewhere around one thousand bit integer or more will be extremely intensive to crack. That is, it will take multiple days to years depending on the number for a classical decoding method to work itself out. As of now, RSA remains secure against classical computing methods due to the computational complexity of factoring large numbers. Shor’s algorithm leverages the quantum Fourier transform and the periodicity of certain mathematical functions to efficiently find the prime factors of a composite number. However, Shor’s algorithm poses a potential threat if large-scale, non-faulty quantum computers become a reality, Shor’s algorithm could significantly hinder the usability and durability of RSA and other cryptographic schemes based on integer factorization and discrete logarithms. [5][6]

2.3 **RSA Example**

1. Choose \( p \) and \( q \) (remember both have to be prime!) say \( p \) is 17 and \( q \) is 19, generally these values would be much larger but for the sake of example we will use smaller numbers.
2. The product of \( p \cdot q = N = (17 \cdot 19) = 323 \)
3. Now we need to find an exponent \( e \) s.t. \( e \) is relatively prime to \((p-1)*(q-1)=288 \) and \( 1 < e < (p - 1) * (q - 1) \) For this case lets say \( e=11 \)
4. Next find a value \( d \) s.t \((d * e) \pmod{(p - 1) * (q - 1)} = 1\), so in our example \((d * 11) \pmod{288} = 1\) so \( d = 131 \)
5. So the public key would be \((e,N)\) or \((11,323)\) and the private key would be \((d,N)\) which is \((131,323)\)
6. - To encrypt a message $m=2$ would be of the form

$$c = 2^{11} \pmod{323}$$

$$c = 110$$

7. - Then to decrypt a message $c=257$ would be of the form

$$m = (110^{131}) \pmod{323}$$

$$m = 2$$

As we can see in the example above the decryption key ($d$) is what fuels this encryption method to work. The most important part of RSA encryption is that the decryption key remains hidden to everyone but the original key creator. If the decryption key is discovered then anyone could decrypt any messages sent using that encryption. This is why ability to factor the composite number $N$ is crucial in keeping this decryption key secret. If the number $N$ is successfully factored then a bad actor could utilize their knowledge of modulo arithmetic (in particular using the following formula)

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

to find the decryption key. Once the decryption key is found no message utilizing that key is safe. Since $N$ is made up of two prime numbers these are the only two possible factors for any given $N$. Given this one would be inclined to think that RSA is rather easy to break, however as the size of $N$ increases it is known that the time it takes to factor scales exponentially. So, factoring an extremely large $N$ could take upwards of 100 years with a classical computer. Knowing this mathematicians aimed to forge a new way to be able to factor these gargantuan numbers that are over 1000 bits long. That method involves finding the period of the encoded message which then allows us to quickly determine the factors. However the one caveat of this method (Shor’s) is that it only works under the assumption of a perfectly operating quantum computer which currently does not exist.

### 2.4 Decrypting using the period of message

Next we will show how one could decrypt RSA without factoring $N = pq$ but instead by finding the period of the encoded message Once the period is found we can use what we know of Euclid and Fermat’s Little Theorem by utilizing a few key formulas:

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

$$m \equiv c^d \pmod{N}$$

$$c \equiv m^e \pmod{N}$$

$$c^r \equiv 1 \pmod{N}$$

$$ed^r \equiv 1 \pmod{r}$$
Where $e$ is the encryption key, $d$ is the decryption key, $m$ and $c$ are the encoded messages, $r$ is the period, and $N$ is the composition of $p$ and $q$ where $p$ and $q$ are what we are trying to find. Let’s say for example that we want to work out our example in reverse where

$$e = 11, m = 110$$

say we find the period of $110 \pmod{323}$ to be $r = 72$ using modular arithmetic, then we know that

$$110^r \equiv 1 \pmod{323}$$

$$11d' \equiv 1 \pmod{72} \quad d' \equiv 59 \pmod{72}$$

then plug $d'$ in to retrieve original message $c \equiv 110^{59} \pmod{323}$ we get that

$$c = 2$$

Which is correct. In this case we were able to find $r$ using a classical computer due to the relatively small intensity of the problem, however extremely large numbers like those used in RSA encryption would require a quantum computer to crack. [3] Also note that $d' = 59$ only decodes $m = 110$ in particular. If we had a different original message then we would need a different decryption key. This is different than our previous decryption key ($d$) in sections 2.2 and 2.3 as ($d$) remains the same no matter what the encoded messages are

### 2.5 Factoring $N$ using the period of message

Once the period of the function has been found this information can be used to factor the large composite integer. The period reveals information about factors of $N$, specifically it says that about $a^r \pm 1$ where $r$ is the period. From here in order to find the factors take the $\gcd(a^r \pm 1, N)$ Finally, so long as the Greatest Common Denominators are prime numbers we are done and these are the factors, if not we must go back and choose a new $a$. Now using the $a$ that we have figured in our example let’s plug in the numbers to find the GCD’s which should in turn be our prime factors of the modulus $N$.

$$\gcd(11^6 + 1, N) = \gcd(1771562, 14) = 2$$

$$\gcd(11^6 - 1, N) = \gcd(1771560, 14) = 7$$

Thus we know that the prime factors for our $N=14$ must be 2 and 7, which in this case is rather elementary, but we can see how this might be useful in factoring larger composite numbers where the factors don’t immediately jump out at us.[3] All in all there are three ways to successfully decrypt RSA encryption.

1. The first being simply having or guessing the decryption key
2. Second factoring $N$ by hand or via other methods to find $p$ and $q$ which gives $(p-1)(q-1)$, from which we can find the decryption key $d$
3. Finally finding the period $r$ of the encoded message mod $N$, which is the smallest $r$ s.t. $b^r \equiv 1 \pmod{N}$. This allows us to find the decryption key without factoring $N$. 

8
3 Introduction to Quantum Computing

3.1 Qubits, Superposition and Entanglement

A quantum bit, or qubit, is the fundamental unit of quantum information in quantum computing. Unlike classical bits, which can represent either a 0 or a 1, qubits can be both a 0 and a 1, thanks to the principles of quantum mechanics. This property is what allows quantum computers to perform certain calculations much more efficiently than classical computers. At its core, a qubit can be any two-level quantum system. The key to understanding how qubits work lies in their ability to exist in multiple states simultaneously. This phenomenon is known as superposition. While a classical bit must be either 0 or 1 at any given time, a qubit can be in a combination of both states simultaneously. Mathematically, this superposition is represented by a complex linear combination of the basis states (0 and 1), where the coefficients of each state describe the probability amplitude of finding the qubit in that state upon measurement. Qubits can also be represented as more than a 2-bit linear combination but rather a combination of n-bits where \( n \geq 2 \).

Another essential property of qubits is entanglement. When qubits become entangled, the state of one qubit becomes dependent on the state of another, regardless of the distance between them. This phenomenon allows quantum computers to perform certain computations in parallel and can lead to significant speedups in certain algorithms. [9] Entanglement and Superposition are what makes qubits useful for Quantum Computing allowing quantum systems to be much more reliable and quicker than their classical counterparts. An example of a qubit might look as follows:

\[ \alpha |0\rangle + \beta |1\rangle \]

Where \( |\alpha|^2 \) and \( |\beta|^2 \) are the probability of the qubit measuring to either 1 or 0. [9][1]

3.2 Quantum Gates and Circuits

The operations that act on qubits are called quantum gates and a combination of these gates is called a circuit. The Quantum gates are simply unitary matrices that exhibit some sort of transformation on the qubits. The Pauli Matrices are an example of some widely used gates for quantum operations. The three Pauli Gates operate as follows:

Pauli X

\[ |0\rangle \rightarrow |1\rangle \text{ and } |1\rangle \rightarrow |0\rangle. \]

Pauli Y

\[ |0\rangle \rightarrow i|1\rangle \text{ and } |1\rangle \rightarrow -i|0\rangle. \]

Pauli Z

\[ |0\rangle \rightarrow |0\rangle \text{ and } |1\rangle \rightarrow -|1\rangle. \]
These are examples of a one qubit gates i.e. they act on one qubit while other gates are able to transform multiple qubits. In general, a qubit is a complex linear combination of $2^n$ n-bit strings, while gates exhibit unitary transformations, and we can understand them as $2^n \times 2^n$ matrices.

The main gate utilized by Shor’s is the quantum fourier transformation that finds the period of the function and transforms a superposition of $2^n$ qubits. However the algorithm itself is simply a mix of 1-qubit and 2-qubit gates. There are many other gates and circuits used like the Hadamard and Controlled gates. When applied these gates transform qubits from the standard basis into the basis of the transformation, so after applying the Hadamard gate to a qubit in the standard basis, that qubit can now be represented in the Hadamard basis. Each basis has it’s own uses and functions however for the purpose of this paper we will mainly focus on the Quantum Fourier Transformation as this is instrumental in the process of Shor’s algorithm.

3.3 Quantum Measurement

Quantum Bits have the unique ability to be both 0 and 1, until they are measured. Then once measurement has occurred the Qubit effectively becomes what it is measured at, at that time. As mentioned above the qubits have probabilities based off of $\alpha$ and $\beta$. Measuring in Quantum is not the same as measuring in classical. In a classical measurement if we know all of the determining and surrounding factors of the measurement we can say with a fair bit of certainty that it will measure one way or another. Now given this same situation but measuring in quantum, despite knowing everything that we should need to know to determine the measurement, the final outcome could be anything in quantum. The measured result only exists once it has been measured in quantum and is unaffected by outside factors. In classical the thing being measured remains that thing even before it is measured. If you are still confused please observe the following comic strip: [10]

![Figure 1: Explanation of Superposition](image-url)
It is important to note that the choice of measurement basis can significantly impact the outcomes and efficiency of quantum algorithms. Certain quantum algorithms, such as those based on Quantum Fourier transforms or phase estimation, naturally operate in non-standard bases. By choosing an appropriate measurement basis, one can often simplify the computational tasks or extract relevant information more efficiently.\[1\] Here is an example of how one might measure a qubit that is in superposition in the standard basis. Say we have a qubit where

$$|\psi\rangle = \frac{1}{2} |0\rangle + \frac{\sqrt{3}}{2} |1\rangle$$

We can check that

$$|\alpha|^2 + |\beta|^2 = 1$$

Since we know that $\alpha = \frac{1}{2}$ and $\beta = \frac{\sqrt{3}}{2}$, $(\alpha)^2 = \frac{1}{4}$ and $(\beta)^2 = \frac{3}{4}$.

The probability of measuring the state as $|0\rangle$ is 25% and the probability of measuring the state as $|1\rangle$ is 75%. As mentioned earlier the measurement process is actually responsible for the collapse of the system, so the only way to collapse a system is to measure it, and doing this many times would be how one determines the general probabilities of the system. This process also changes the state of the qubit and it is no longer a superposition of states, but rather one post measurement $|\psi\rangle = \text{to either } |0\rangle \text{ or } |1\rangle$.

This time let’s use a system where Number of Registers = 2 and Number of Qubits = 4. For this example we will only measure the right register and determine the probabilities for them. Say this system is of the form:

$$|\psi\rangle = \frac{1}{\sqrt{3}} |00\rangle + \frac{1}{\sqrt{3}} |01\rangle + \frac{1}{\sqrt{3}} |11\rangle$$

So to measure the right register we take the probabilities of each qubit for the right register only. So we can rewrite

$$|\psi\rangle = (\frac{1}{\sqrt{3}})[00] + (\frac{1}{\sqrt{3}})(|0\rangle + |1\rangle) \otimes |1\rangle$$

This means that we measure the following probabilities for our system

$$\frac{1}{\sqrt{3}}|00\rangle \text{ with probability of } \frac{1}{3} \text{ and } \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle \text{ with probability of } \frac{2}{3}$$

In this example since we only measured the right register then only the right register collapses and the left register remains in a superposition. So knowing this we can see that if we were to measure the left qubit then $|00\rangle$ stays as $|00\rangle$, but $\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)$ could be either $|01\rangle$ or $|11\rangle$. In the above example we seemingly did not include the qubit $|10\rangle$ however the qubit was there but it had a 0% chance of being measured because the probability of measuring any of the other three registers already summed up to 1.

\[1\] Note that if $|\alpha|^2 + |\beta|^2 \neq 1$ then something is wrong.
4 Shor’s Algorithm: The Breakthrough

4.1 Origin and Background of Shor’s Algorithm

In 1994 a mathematician by the name of Peter Shor discovered a new algorithm that would change the view of quantum computing for the rest of the world. Published in his paper “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. This paper showed that the factoring of large numbers could be done quite easily and efficiently on quantum computers. At the time of its release the algorithm was rather useless given that it required a sufficient quantum computer to complete the algorithm in less time than a classical computation, which at the time of his discovery the field of quantum mechanics and computing was much smaller and little to no focus was done in that area. Despite this fellow mathematicians were able to understand the significance of Shor’s algorithm almost immediately. [5]

4.2 Key Steps and Operations of Shor’s Algorithm

Shor’s Algorithm is a fairly simple process, that utilizes both quantum and classical computing. The algorithm also utilizes rudimentary number theory to in order to prove or check the accuracy of the period. Here’s how it breaks down on a step by step basis:

1. Function Evaluation: Shor’s algorithm aims to find the period \( r \) of a function \( f(x) \) that maps integers to integers. In the context of integer factorization, this function is

\[
(f(x) = a^x \mod N)
\]

where \( a \) is the message we want to decode between \( 2 \) and \( N-1 \), and \( N=pq \) is the number to be factored.

2. Quantum Fourier Transform: Shor’s algorithm starts by preparing a quantum superposition of all possible input values of the function \( f(x) \). This superposition is achieved using quantum parallelism. To apply the QFT to this superposition we must measure to find all \( x \) where \( a^x = x \). The QFT is then applied to this superposition.

3. Period Finding: The QFT effectively amplifies the components of the superposition that correspond to the periods of the function \( f(x) \). Due to the nature of the QFT, the output of the QFT will have peaks at positions corresponding to the periodicities of \( f(x) \).

4. Measurement: After applying the QFT, a quantum measurement is performed on the output state. The measurement collapses the quantum state to one of the basis states, each corresponding to a possible period of the function.
5. Classical Post-processing: Classical post-processing techniques are employed to extract the period from the measurement outcome. This typically involves using continued fractions or other mathematical methods to determine the period from the measurement result.

6. Period Verification: Once a candidate period is obtained, it needs to be verified classically to ensure its correctness. This verification step is essential to ensure the accuracy of the algorithm’s output.

One caveat of period finding utilizing Shor’s algorithm is that once r has been found it is simpler to analyze when \( \frac{2^n}{r} \) is an integer otherwise a much more complicated formula is needed. I will give an example of both of these instances, starting with the cleaner instance where \( \frac{2^n}{r} \) is an integer.

For example, Say we want to find the period of the message \( b = 5 \pmod{14} \) Then \( f(x) = 5^r \) we want \( x \) s.t. 
\[
f(x) = 5^r \equiv 1 \pmod{14}
\]

As we can see this is the setup for finding the period where \( x = r \) in this scenario, so to find the period we must find \( x \) \( r \equiv 1 \pmod{14} \). After plugging in some values for \( x \) we can see that \( x=6 \) or \( r=6 \) (same thing!) works for the equation:

\[
5^2 = 25 \equiv 11 \pmod{14}
\]
\[
5^4 = 121 \equiv 9 \pmod{14}
\]
\[
5^6 = 5^4 * 5^2 = 9 * 11 = 1 \equiv 11 \pmod{14}
\]

This is a classical way to find a rather elementary period. However in our later examples we will show how a quantum computer would be utilized to find the period of the encoded message as well.
4.3 An In Depth Review of Shor’s Algorithm

The Quantum Fourier Transformation can be described as the following transformation of $2^n$-qubits:

$$Q_{FT}|x_n\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n}|y_n\rangle$$

For example say $n=1$ then the number of qubits would be two, so a Unitary Quantum Fourier Transformation would look as follows:

$$Q_{FT}|0\rangle = \frac{1}{\sqrt{2}}(e^{2\pi i 0}|0\rangle + e^{2\pi i 0}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$Q_{FT}|1\rangle = \frac{1}{\sqrt{2}}(e^{2\pi i 0}|0\rangle + e^{2\pi i /2}|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

For our purposes we will utilize a 4-qubit Quantum Fourier transformation in two examples. This QFT is of the form seen in Figure 2.

Figure 2: Quantum Fourier Transform of 4 qubits.
The process of using the Quantum Fourier Transformation to find the period of a function can be described generally as follows:

1. Create superposition of the form
\[ \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x| f(x) \]

2. Measure right register to create on the left register
\[ \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + Kr \rangle \]

Where \( x_0 \) is what we get from measuring the right register, \( Kr \) is an integer multiplied by the period, and \( m \) is the smallest integer s.t. \( mr + x_0 \geq 2^n \). In other words this means that \( m = \frac{2^n}{r} \) or \( m = \frac{2^n}{r} + 1 \)

3. Apply QFT
\[ QFT( \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + Kr \rangle ) = \]
\[ \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} QFT |x_0 + Kr \rangle = \]
\[ \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} \frac{1}{2^n} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i (x_0 + Kr) y}{2^n}} |y_n \rangle = \]
\[ \frac{1}{\sqrt{m}} \frac{1}{2^n} \sum_{k=0}^{m-1} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i x_0 y}{2^n}} e^{\frac{2\pi i Ky}{2^n}} |y_n \rangle = \]
\[ \frac{1}{\sqrt{m}} \frac{1}{2^n} \sum_{y=0}^{2^n-1} e^{\frac{2\pi i x_0 y}{2^n}} \sum_{k=0}^{m-1} e^{\frac{2\pi i Ky}{2^n}} |y_n \rangle = \]

4. We measure again to get \( |y \rangle \) with probability of \( p(y) \) where
\[ p(y) = \frac{1}{m 2^n} \left| \sum_{k=0}^{m-1} e^{\frac{2\pi i Ky}{2^n}} \right|^2 \]

5. Finally compute \( r \) using classical number theory where It can be shown that \( p(y) \) is largest when index \( y \) is close to \( \frac{2^n}{r} J \)

15
4.4 Shor’s Algorithm Example 1

Let us look at the simpler situation where $r$ must divide $2^n$ in order for $\frac{2^n}{r}$ positions of peak periodicity. So let’s look at an example where that is true take $13^x \equiv 1 \pmod{14}$ for example. This is an example of a 4-qubit system as well because our $N$ is the same. We know that the expansion of this series looks like:

$$\frac{1}{\sqrt{16}} \sum_{x=0}^{16-1} |x\rangle|13^x\rangle = \frac{1}{\sqrt{16}} ([0]|1\rangle + [1]|13\rangle + [2]|1\rangle + [3]|13\rangle + [4]|1\rangle + [5]|13\rangle$$


After measuring the right register, the left register can collapse into one of two possible vectors. One corresponds to when $13^x = 1$ and is:

$$|v_1\rangle = \frac{1}{\sqrt{8}} ([0] + [2] + [4] + [6] + [8] + [10] + [12] + [14]) = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$
The other is when $13^x = 13$ and is

$$|v_1\rangle = \frac{1}{\sqrt{8}}(|1\rangle + |3\rangle + |5\rangle + |7\rangle + |9\rangle + |11\rangle + |13\rangle + |15\rangle) = \frac{1}{\sqrt{8}}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Both of these vectors are multiplied by $\frac{1}{\sqrt{8}}$ to make sure they are normalized. Now we take our superposition where the right register has been measured to be either 1 or 13. We apply the Quantum Fourier Transform (see Figure?) to both of these vectors in order to understand where the peaks of periodicity are.

$$Q_{FT}v_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$
\[
QFT_{v_{13}} = \begin{pmatrix}
1/\sqrt{2} \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
1/\sqrt{2}
\end{pmatrix}
\]

From this we can see that the large points are at 0, 8 respectively. Measurement would yield either \(|0\rangle\) or \(|8\rangle\) with \(p(y) = 1\) probability or \(p(y) = 0\) probability for all other values. In this situation we know that \(\frac{2^n}{r}J = 0\) or 8 exactly, since \(2^n = 16\) we know that \(r = 2\).

Breaking down our formula for \(p(Y)\) we can see why \(\frac{2^n}{r}J\) is equal to 8 or more generally why the Quantum Fourier formula says that \(\frac{2^n}{r}J\) is equal to \(m\)

\[
\sum_{k=0}^{m-1} e^{2\pi i k y (r)} = m, \text{ if } y = 0 \text{ or } y \left(\frac{r}{2^n}\right) \text{ is an integer}
\]

and

\[
\sum_{k=0}^{m-1} e^{2\pi i k y \left(\frac{r}{2^n}\right)} = 0, \text{ if } y \left(\frac{r}{2^n}\right) \text{ is not an integer}
\]

so we set

\[
\frac{2^n}{r}J = 8 \text{ since we know } 2^n = 16 \text{ we can say } 8r = 16 \text{ so } r = 2
\]

Now we have found the period \(r=2\) which does divide \(2^n\) and therefore we are able to verify this is the period classically. To double check whether our period is correct we can see if it satisfies the formula \(13^x \equiv 1 \pmod{14}\) where \(x = r\)

\[13^2 = 169 \equiv 1 \pmod{14}\] Since \(14 \times 12 = 168 + 1 = 169\) it is true \(13^2 \equiv 1 \pmod{14}\)
4.5 Shor’s Algorithm Example 2

Now suppose we wish to revisit our earlier example of \( f(x) = 5^x \equiv 1 \pmod{14} \). We must use the \( Q_{FT} \) on the superposition of \( f(x) \) such that all values of \( 5^x \) are encoded for \( x = 0, 1, 2...2^n - 1 \) we get that

\[
\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle |f(x)\rangle = \frac{1}{\sqrt{16}} \sum_{x=0}^{15} |x\rangle |5^x\rangle =
\]

\[
\frac{1}{\sqrt{16}} (|0\rangle |5^0\rangle + |1\rangle |5^1\rangle + ... |15\rangle |5^{15}\rangle ) =
\]

\[
\frac{1}{\sqrt{16}} \sum_{x=0}^{15} |x\rangle |13^x\rangle = \frac{1}{\sqrt{16}} (|0\rangle |1\rangle + |1\rangle |5\rangle + |2\rangle |11\rangle + |3\rangle |13\rangle + |4\rangle |9\rangle + |5\rangle |3\rangle

+ |6\rangle |1\rangle + |7\rangle |5\rangle + |8\rangle |11\rangle + |9\rangle |13\rangle + |10\rangle |9\rangle + |11\rangle |3\rangle

+ |12\rangle |1\rangle + |13\rangle |5\rangle + |14\rangle |11\rangle + |15\rangle |13\rangle )
\]

In this situation we can see that \( n=4 \) because \( 8 < 14 < 16 \) so \( 2^4 \) or 4 qubits is the minimum number of qubits needed to perform this calculation. Since we know that a quantum computer has all the capabilities of a classical computer we are also able to assume that a quantum computer would be able to create this superposition as a classical computer would be able to. Measuring the right side of this function will tell you one of 1, 5, 11, 13, 3, 9, but the left side is an equal superposition of \( x \) s.t. \( 5^x = \) what was measured on the right. Once measured this superposition can be expressed as one of the following vectors

\[
v_{5^x=1} = \frac{1}{\sqrt{3}} (|0\rangle + |6\rangle + |12\rangle ) = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}
\]

\[
v_{5^x=3} = \frac{1}{\sqrt{2}} (|5\rangle + |11\rangle ) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}
\]

19
\begin{align*}
    v_{5^e=5} &= \frac{1}{\sqrt{3}} (|1\rangle + |7\rangle + |13\rangle) = \frac{1}{\sqrt{3}} \\
    v_{5^e=9} &= \frac{1}{\sqrt{3}} (|4\rangle + |10\rangle) = \frac{1}{\sqrt{2}} \\
    v_{5^e=11} &= \frac{1}{\sqrt{3}} (|2\rangle + |8\rangle + |14\rangle) = \frac{1}{\sqrt{3}} \\
    v_{5^e=13} &= \frac{1}{\sqrt{3}} (|3\rangle + |9\rangle + |15\rangle) = \frac{1}{\sqrt{3}}
\end{align*}

Now in order to find the period we must take these vectors and multiply them by our Quantum Fourier Transformation. As seen in Figure 2. Multiplying this matrix by our vectors gives us Figures 3-8. Note that a few of these figures have been manipulated beyond just applying the QFT,\(^2\) however the important information still remains.

\(^2\)These manipulations are simply just absolute values and squares post QFT
Figure 3: Vector of $5^x = 1$ post QFT

Figure 4: Vector of $5^x = 5$ post QFT

Figure 5: Vector of $5^x = 11$ post QFT
Using the vectors from Figures 3-8 we can see that the period peaks at positions 3, 5, 8, 11, 13. So the probability of measuring 3, 5, 8, 11, 13 is large, at least relative to the other positions.

After we measure, with high probability we will get one of 3, 5, 8, 11, 13. The output will be close to $2^n J$ for some integer $J$, but we don’t know what exact integer $J$ is right off the hat. For example, we might get $16j/r = 5$, but we don’t know what $r$ or $j$ is.

One method is to make guesses for $J$, which would give us some possible periods $r$. We can use a classical computer to verify if a guess for $r$ is correct (by computing whether $5^r = 1 \mod 14$).

Alternatively, to find $J$ more precisely, we can run the quantum algorithm more times, and then we would get a collection of which are close to integer multiples of $16/r$. There are classical algorithms based on continued fractions.
to deal with this scenario. It is beyond the scope of this paper. \[^3\] [3]

Since we know \(2^n = 16\) we have that

\[
\frac{16}{r} J = 0, \quad \frac{16}{r} J = 3, \quad \frac{16}{r} J = 5, \quad \frac{16}{r} J = 8, \quad \frac{16}{r} J = 11, \quad \frac{16}{r} J = 13
\]

\(\frac{16}{r} J = 0\) Doesn’t tell us much for \(r\)

\(\frac{16}{r} J = 3\) or \(3r = 16\) (1) so \(r\) is between 5 and 6

\(\frac{16}{r} J = 5\) or \(5r = 16\) (2) so \(r\) is about 6

\(\frac{16}{r} J = 8\) or \(8r = 16\) (3) so \(r\) is equal to 6

\(\frac{16}{r} J = 11\) or \(11r = 16\) (4) so \(r\) is between 5 and 6

\(\frac{16}{r} J = 13\) or \(13r = 16\) (5) so \(r\) is about 6

So we have somewhat verified that the period is 6.

\[^3\text{See pages 79-82 in Quantum Computer Science: An Introduction}\]
5 Conclusion

Shor’s algorithm is able to quickly and efficiently factor large numbers via the use of a Quantum Fourier Transformation and period finding. Once the period of an encoded message is found and verified, malicious actors can use the period to find the factors and subsequently gain access to encrypted messages from their own decryption key (d’).

5.1 Recapitulation of Key Points

1. RSA encryption relies on the fact that extremely large composite numbers (N) are difficult to factor via current classical computing methods.
2. Shor’s Algorithm utilizes the mechanics of a quantum computer so that it can reliably and quickly factor large integers compared to classical.
3. Shor’s Algorithm utilizes the Quantum Fourier Transformation to find the period of an encoded message.
4. Shor’s utilizes knowledge of modulo division, and our new found knowledge of the period so that we can find the factors (p) and (q) of (N).
5. These factors are found by utilizing the formula gcd(a^r ± 1, N) where the results should yield two prime factors of (N).

5.2 Implications for RSA Encryption

The creation of extremely powerful quantum computers is simply a matter of time at this point with companies and governments around the world vying to be the first to achieve a true quantum computer that is able to complete problems in half or less than that a classical computer. As the days go by this inevitability becomes closer and closer which scares many people in the cryptography realm because once this is our reality, then many things we do on a daily basis through the internet may no longer be considered safe. One example is that of https. https is a web protocol that encrypts all information the website shares with the user and vice versa. In a post quantum world these encryptions might be worthless. In a post quantum world passwords that are encrypted via certain classical techniques will be vulnerable to attack. Any and all old data will be vulnerable to any person able to access this technology. This may not seem like to big a deal, however things like credit cards or social security numbers which don’t change will be available for malicious actors. While this information is already somewhat accessible to those who know how to access it. Shor’s algorithm will simply expedite and accelerate this process. So while there are some classical cryptography methods that remain resistant to quantum processing, a majority of these processes are in fact extremely vulnerable to algorithms like Shor’s. [8]
5.3 Future Outlook and Recommendations

Despite all the doom and gloom I believe people should look forward to a post quantum world. We will be able to make calculations and run simulations that were considered impossible beforehand, and while classical cryptography is vulnerable to quantum attacks, a new form of quantum cryptography will be born. In fact many cryptographers today are solely focused on creating different quantum cryptography techniques in order to prepare for the coming advancements in quantum computing. These cryptographic techniques will be so good that even the most efficient conceivable quantum computer would take hundreds of years to break them. All in all while some concern is warranted, so long as bad actors don’t get their hands on this technology first as well as figure how to properly utilize it. The world at large should have enough time to adapt to these quantum advances with quantum advances of their own. Many governments and companies are already likely moving on from basic RSA encryption techniques. However any data previously encrypted in the old system would still remain vulnerable to Shor’s. Finally I would be amiss if I didn’t mention a recent article where a mathematician has apparently developed an algorithm that will potentially factor large numbers in $n^{1.5}$ compared to Shor’s $2^n$.[9] This technique utilizes less gates then Shor’s which allows it to theoretically factor large numbers faster then Shor’s.

5.4 Closing Remarks

The exploration of Shor’s algorithm within this paper aimed to provide valuable insights into the realm of quantum computing and cryptography. By highlighting the underlying principles and mechanics of this algorithm, we have not only deepened our understanding of quantum processes but also highlighted the immense potential it holds for revolutionizing cryptography and computational mathematics. Shor’s algorithm stands as a testament to the remarkable potential of quantum computing to solve problems that were previously deemed impossible by classical means. As we continue to advance our knowledge and capabilities in the field of quantum computing, the profound implications of Shor’s algorithm serve as a beacon for guiding our exploration of the quantum frontier. With further research and development, we will unlock even greater applications and possibilities. Eventually Shor’s will be remembered as a stepping stone for truly great quantum processes. Despite this Shor’s has undoubtedly laid the groundwork for future quantum explorers to further and improve upon his endeavors in the cryptography space.
6 Bibliography


