Indian Institute of Technology Madras

Chennai, India

Attacking cryptosystem using Quantum algorithms

Students

- Chaganti kamaraja Siddhartha: EP20B012
- T Mani Venkata Krishna: CS20B082

Guide

- Professor Chester Rebeiro

May 8, 2023

Contents

1 Introduction

Quantum computing has emerged as a revolutionary technology that has the potential to significantly impact a wide range of applications, including cryptography. Cryptography is an essential aspect of modern-day communication and plays a crucial role in ensuring the security of our sensitive data. However, the advent of quantum computing threatens to render many of the existing cryptographic techniques vulnerable.

In this project, we aimed to explore the potential of quantum computing in breaking two widely used cryptographic algorithms, the Even Mansour cipher and RSA algorithm. We used two of the most prominent quantum algorithms, Simon's algorithm and Shor's algorithm, respectively, to break these algorithms.

To simulate the attacks, we used the IBMQ Aer simulator, which is a state-of-the-art quantum simulator capable of emulating quantum circuits with a high degree of accuracy. The simulator enabled us to run large-scale simulations and study the impact of different parameters on the performance of our quantum algorithms.

To understand the functioning of these algorithms, we first delved into the fundamentals of quantum computing, including quantum states, gates, and quantum parallelism. We also described the Deutsch-Josza algorithm, which is a well-known quantum algorithm used for solving a particular type of problem known as a "black box" problem. The algorithm is widely used to demonstrate the power of quantum computing and serves as a building block for more complex quantum algorithms.

We then moved on to describe Simon's algorithm, which is used for finding the period of a function. The algorithm has important implications in cryptography and can be used to break certain types of cryptographic schemes, including the Even Mansour cipher. We explained the algorithm in detail, including the steps involved and the mathematical principles underlying it.

Finally, we discussed Shor's algorithm, which is perhaps the most well-known quantum algorithm used for breaking RSA encryption. The algorithm exploits the fact that the problem of finding the prime factors of a large number is extremely hard for classical computers but can be solved efficiently using a quantum computer. We explained the algorithm in detail, including the quantum circuit used to implement it and the steps involved in finding the prime factors.

Overall, this project aimed to demonstrate the potential of quantum computing in breaking widely used cryptographic algorithms. By using state-of-the-art quantum algorithms such as Simon's algorithm and Shor's algorithm, we were able to simulate attacks on the Even Mansour cipher and RSA algorithm, respectively, and demonstrate the power of quantum computing in cryptography. The project highlights the need for developing new cryptographic techniques that can withstand the threat posed by quantum computing and paves the way for further research in this area.

2 Overview of Crypto Algorithms

2.1 Even-Mansour

Shimon Even and Yishay Mansour designed this block cipher. We break the message into blocks and we apply encrytion. Let us call one block as 'M'. In encryption we use a prewhitening key K_2 and postwhitening key K_2 an a function **F**. Here K_1,K_2 are private info and **F** is known to all.
Let ENC(M) be the overall encrytion function and C be the reulting cipher. Here the circuit of F and F^{-1} is known to everyone but K_1 and K_2 are only known to sender and reciever.

$$C = ENC(M) = F(K_1 \oplus M) \oplus K_2$$

We use above encryption function on block of message 'M' and generate cipher 'C'. Below attached picture depicts the same.

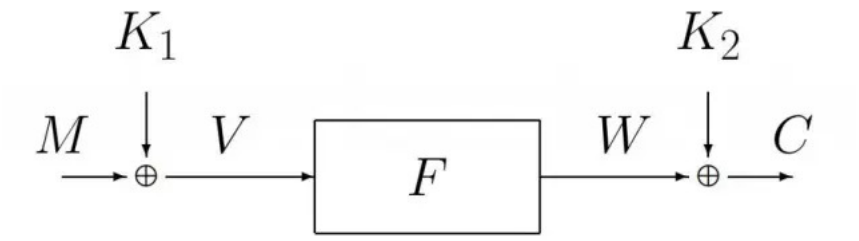


Figure 1: Even Mansour

For decryption we calculate $K_2 \oplus \mathbf{W}$ followed by passing it through F^{-1} and the output will be **V** and we the message **M** will be $K_1 \oplus \mathbf{V}$.
Even Mansour is a minimal block cipher. It is the simplest possible block cipher with formal proof of security. It is necessary to know K_1 and K_2 to break the cipher.

2.2 RSA

RSA is developed by Rivest, Shamir and Adleman. This encyption scheme is public key encryption scheme. Security because of high cost of factoring of very large numbers.
Generate two very large prime numbers. Let us name them p, q. Calculate N=p*q. And a small number 'e' such that it is coprime to (p-1)*(q-1). And let 'd' be the modulo inverse of e modulo (p-1)*(q-1). 'e' is the public key. 'd' is the private key.
Now break the message in segments. Take a segment of data that needs to be

encrypted let it be 'x'. Let 'e' be the public key of reciever.
To encrypt, sender calculates c by using below formula. And transmits the cipher(c).

$$c = x^e (mod\, N)$$

To decrypt, reciever uses his private key(d) to calculate 'x' using below formula.

$$x = c^d (mod\, N)$$

Below is a depiction of how RSA works.

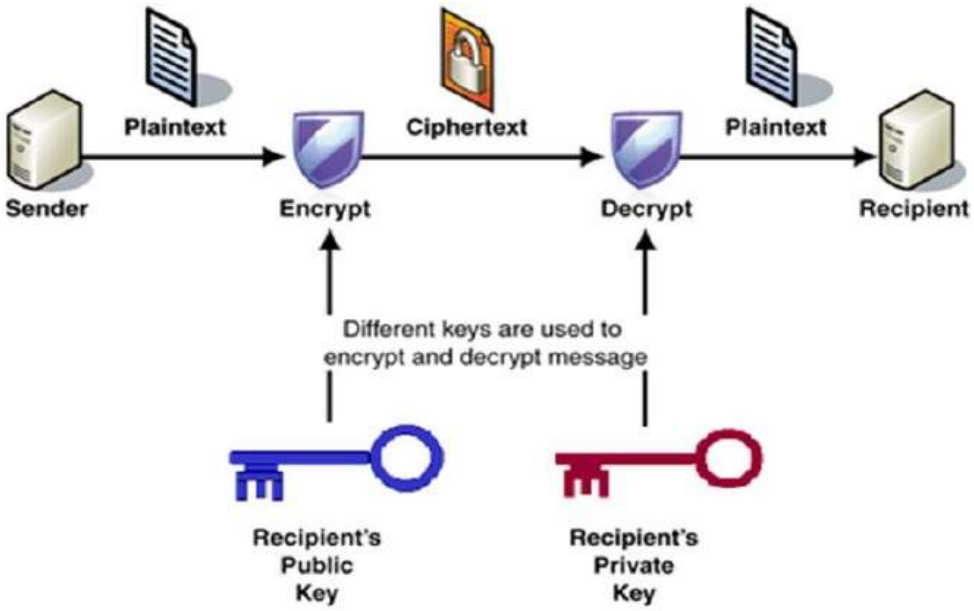


Figure 2: RSA

For breaking RSA we need to factorise **N**. But in classical approach it takes exponential time complexity in number of bits. We use **Shor's Algorithm** which uses quantum approach for solving **Order finding problem**. We can prove that by solving order finding problem in polynomial time we will be able to solve factorisation problem in polynomial time.

2.2.1 Order finding problem

Given two integers **a** and **N** finding the smallest **r** such that $a^r = 1\ (mod\, N)$ is the order finding problem.
Example: for $N = 10$ and $a = 3$, $order = 4$ as $3^4 = 1\ (mod\, 10)$

2.2.2 Solving factorisation problem using order finding problem

Let us start with the assumption that we can solve order finding problem in polynomial time. For simplicity, let us say we have to factor **N** which is product of two distinct primes p_1 and p_2.
We pick a random integer **a** from 2 to $N-1$ and compute $gcd(a, N)$ this can be done in polynomial time. If gcd is not 1 then it must either be equal to p_1 or p_2 in this case the factorisation problem is solved. So let us say $gcd(a, N) = 1$.
Let **r** be the order of $a \; modulo \; N$ which can be evaluated in polynomial time.
We repeat the above steps until we find **r** that is even for some **a**. There is very significant fraction of **a**'s which have even order for a **N**. $a^r - 1$ is a multiple of **N**. If r is even we can write $a^r - 1$ as,

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$$

$(a^{r/2} - 1)$ cannot be multiple of **N** because if that were the case then order would be $r/2$ not r.
case(i):
$(a^{r/2} + 1)$ is not a multiple of **N**. That means $(a^{r/2} - 1)$ and $(a^{r/2} + 1)$ are not multiple of **N** but their product is. That implies p_1 is a prime factor of $(a^{r/2} - 1)$ and p_1 is a prime factor of $(a^{r/2} + 1)$ or vice versa. In this case we can find p_1 by just calculating $gcd(a^{r/2} - 1, N)$ similarly p_2 by just calculating $gcd(a^{r/2} + 1, N)$.
case(ii):
$(a^{r/2} + 1)$ is a multiple of **N**. In this case we cannot do anything but choose another **a** but these cases are very less frequent.

Hence we can say that by solving order finding problem we can solve factorisation problem.

3 Quantum Computing -Summary

To keep things simple for a reader with no background in Quantum Mechanics or Quantum Computing, In this section we discuss how an attacker can get access to authorized information without much involvement of Quantum Computing and Display the results of our simulation.
For readers who don't want to make their hands dirty with quantum computing please take the following points as granted and proceed further.

1. Quantum states are represented using state $|0\rangle$ and $|1\rangle$ similar to classical states 0 and 1.

2. Unlike Classical states which only exist in either 0 or 1, Quantum states exists in superposition of states. For example an arbitrary quantum state is represented as

   $$a\,|0\rangle + b\,|1\rangle$$

   where, a, b are complex numbers and $a^2 + b^2 = 1$.

3. Even though Quantum states exists in superposition when measured in computational basis they only output either 0 or 1. Please note that they output classical states 0 and 1 and not quantum states $|0\rangle, |1\rangle$.

4. If a two - one function $f(x)$ with period s is given there exists a quantum algorithm called Simon's algorithm which can find the period s in polynomial time.

5. If a Number which is product of two primes is given there exists a quantum algorithm called Shor's algorithm which makes use of order-finding and outputs the prime factors of given number.

For more detailed explanation please refer to **Appendix.**

4 Attack on Even-Mansour

Recall from, section 2.1 that Even mansour encryption is given by

$$Enc(M) = F(k_1 \oplus M) \oplus K_2$$

The secret keys k_1 and k_2 are known only to sender and receiver to encrypt and decrypt message M.
The attacker has no access to k_1, k_2 but have access to the box which encrpyts that is he can get output Encrypted text for a given input but he don't know the secret keys k_1, k_2 stored in the box. The goal of the attacker is to find the secret keys k_1, k_2.
The attacker also have access to funtion box $F(x)$ that is he can compute $F(x)$ for a given input x. He also have access to function box $F^{-1}(y)$ that is given an input y it can compute $F^{-1}(y)$.
Summarizing the assumptions and goal of attacker:
Resources of attacker:

1. Can compute $Enc(M) = F(M \oplus k_1) \oplus k_2$ given M.

2. Can compute $F(x)$ given x.

3. Can compute $F^{-1}(y)$ given y.

Goal: To find k_1, k_2 using given resources.

Attack

Kuwakado and Morii (2012) showed that there is a function $f(M)$ that can be computed from given resource where the function $f(M)$ is periodic with period k_1. The function given by Kuwakado and Morii (2012) is

$$f(M) = Enc(M) \oplus F(M)$$

$$f(M) = F(M \oplus k_1) \oplus k_2 \oplus F(M)$$

$$f(M \oplus k_1) = F(M \oplus k_1 \oplus k_1) \oplus k_2 \oplus F(M \oplus k_1)$$
$$f(M \oplus k_1) = F(M) \oplus k_2 \oplus F(M \oplus k_1)$$
$$f(M \oplus k_1) = f(M)$$

Therefore, we can clearly see that $f(M)$ is periodic with period k_1. Now, if k_1 is found we can xor $Enc(M)$ with $F(M \oplus k_1)$ to find k_2. Now, from point 4 from section 3 using Simon's algorithm we can find k_1 and k_2.
The important point to note here is finding a periodic function which can be used to compute secret information. Luckily Kuwakado and Morii (2012) did that work for us. What if there is something new?
Canale et al. (2022) authored a code available at period-search. (n.d.). GitHub. Retrieved May 7, 2023, from https://github.com/rub-hgi/period-search.git which can be used to generate periodic function like $f(x)$ as shown above for different ciphers like Even-Mansour, Fiestel-3,4,5 rounds and Misty-5k ciphers. The output of their code generating the function by taking encryption and resources as input is shown in results section.

5 Results

5.1 Even-Mansour

Canale et al. (2022) wrote a function to take input of all resources and considered them as gates which compute the function of given input, and give the output. Their method is to brute force check all the functions that can be computed from given resources and check wether the composite function of all these is periodic or not. Since, brute force takes long time, they defined concepts like Equivalence circuits, ordered cirucits, Depth of a node, etc and using these they constructed the brute force attack because of these conditions most of the unwanted cases are elminated. The output of this for even-mansour cipher is shown in figure [1]. In the figure [1] shown blue color nodes represent inputs. zero on blue color node represent Message M. Blue color nodes represent gates. Zero on orange color node represent xor function. one on orange color node represent Encryption function $Enc(M)$, two on orange color node represent $F(M)$ function. Edges with number one represent input number one. Edges with number two represent input with number two. Edges with number three represent the gate requires only one input.
Note :The arrows in the diagram are in reverse direction and we tried to reverse the order but unable to do so. Therefore consider the direction of arrow to be reversed.
Now, the output shows blue color node zero is input to orange color node one and two. That means the outputs of edges one and two are $Enc(M)$ and $F(M)$. These two are input to orange color node zero that means the output is xor of one and two edges that is $Enc(M) \oplus F(M)$ this is exactly same as the function given by Kuwakado and Morii (2012). Let this function $f(x)$ can be realised by a quantum circuit between first 2 barriers given in figure [**??**]. Now, the simon

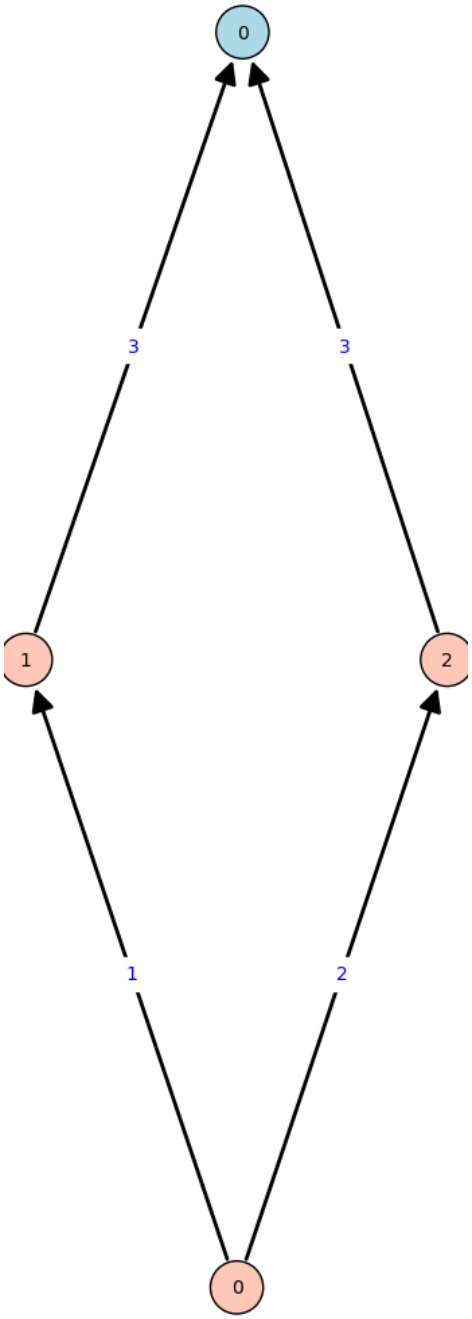


Figure 3: periodic function for even mansour attack

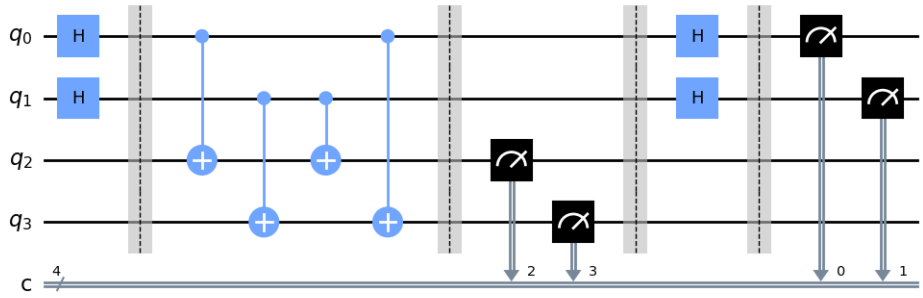


Figure 4: Even mansour simon

algorithm is applied and the output is shown in figure [5]. The output of first two qubits is either 00 or 11 therefore, if the key is 'ab' then

$$a + b = 0$$

$$\implies ab = 00 \text{ or } 11$$

but it is periodic with non-zero period therefore, period $s = 11$. Please refer to simon algorithm in appendix for more detailed explanation of how and why simon algorithm works.

6 RSA

From Section three point number six we know that shor's algorithm can be used to factor a number which is product of two primes. To factor fifteen which is product of three and five, we used Shor's algorithm described in Appendix. The circuit and phase estimation results are explained clearly in the appendix. Recall from Section 2.2 the number a is taken to be $a = 7$ and the order of a^r mod 15 is 4. Therefore, the factors can be easily described using a and r. Thex figure [6] shows the result of the shor's algorithm.

Appendix

7 Quantum States and Qubits

Quantum states and qubits are fundamental concepts in quantum computing and quantum information theory. A qubit is a unit of quantum information that can exist in two distinct states, often referred to as $|0\rangle$ and $|1\rangle$, similar to classical

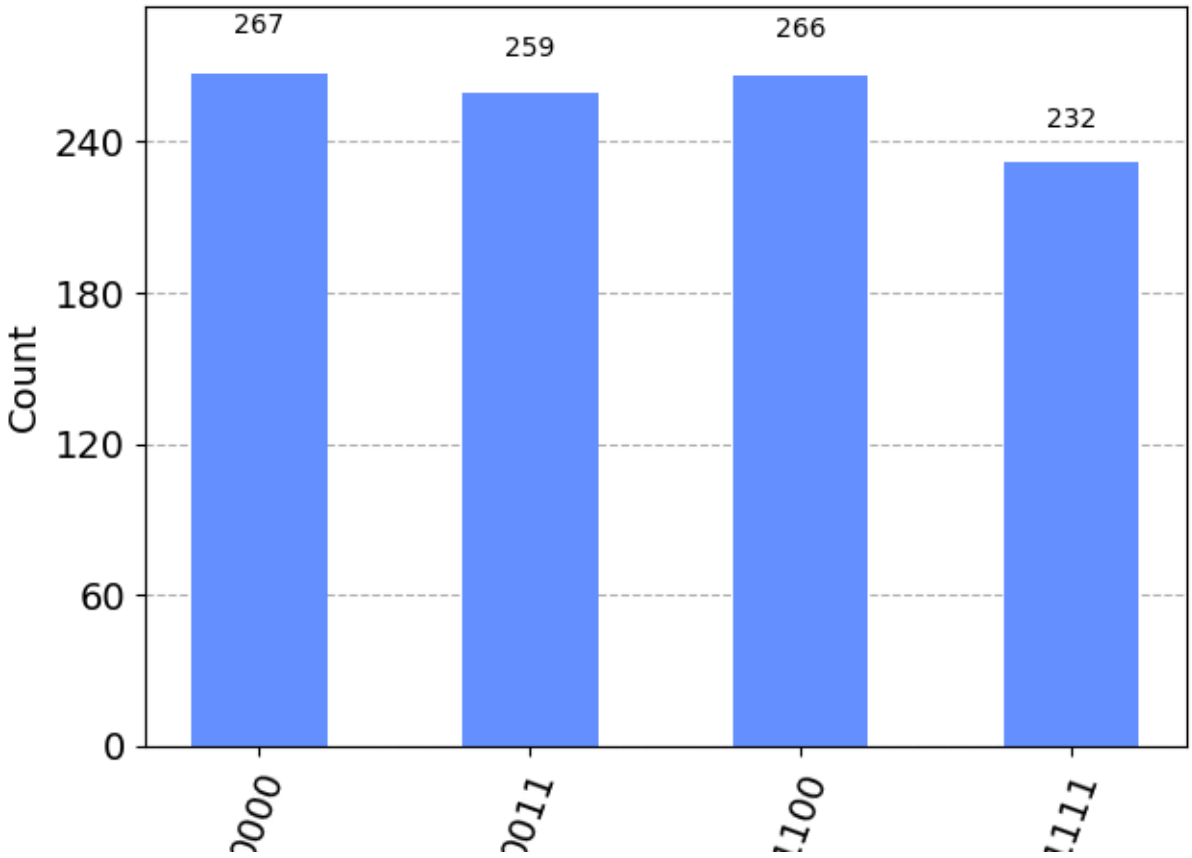


Figure 5: results of even mansour simon

```
ATTEMPT 1:
Register Reading: 01000000
Corresponding Phase: 0.25
Result: r = 4
Guessed Factors: 3 and 5
*** Non-trivial factor found: {guess} ***
*** Non-trivial factor found: {guess} ***
```

Figure 6: Result with a = 7 for N = 15

bits. However, unlike classical bits, qubits can exist in superpositions of these states, meaning they can be in a combination of both states simultaneously. This property allows for quantum computers to perform certain calculations exponentially faster than classical computers.

Quantum states are described using quantum mechanicsNielsen and Chuang (2011), which is a branch of physics that studies the behavior of particles on a microscopic level. In quantum mechanics, the state of a system is described using a wave function, which gives the probability amplitude of each possible state that the system can be in. This wave function can be manipulated using quantum gates, which are the building blocks of quantum circuits.

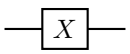
One of the most famous quantum algorithms that exploits the power of quantum states and qubits is Shor's algorithmShor (1994), which can factor large integers in polynomial time, a problem that is believed to be exponentially hard for classical computers. This algorithm has potential applications in cryptography, as many modern encryption techniques rely on the difficulty of factoring large numbers.

In summary, quantum states and qubits are essential concepts in quantum computing and quantum information theory, with potential applications in cryptography, chemistry, and optimization problems.

8 Quantum Gates

As we know we use classical gates to change the state of classical bits, In quantum computing we use quantum gates to change quantum bits(qubits) from one state to another. Some important quantum gates are listed below,

8.1 X-gate

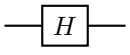


Similar to classical NOT gate this gate when applied changes the state in following way

$$|0\rangle \rightarrow |1\rangle$$

$$|1\rangle \rightarrow |0\rangle$$

8.2 Hadamard gate



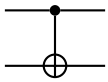
This gate is called Hadamard gate and it changes the quantum state in the following way

$$|0\rangle \rightarrow \frac{1}{2}(|0\rangle + |1\rangle)$$

$$|1\rangle \rightarrow \frac{1}{2}(|0\rangle - |1\rangle)$$

Counterintutive to classical computing as we already know quantum states exists in superposition of $|0\rangle$ and $|1\rangle$ state. Here, the state $\frac{1}{2}(|0\rangle + |1\rangle)$ is represented by $|+\rangle$ and $\frac{1}{2}(|0\rangle - |1\rangle)$ is represented as $|-\rangle$.

8.3 C-NOT gate



This gates takes two qubits as inputs one is control qubit and the other is target qubit. In the above figure the qubit with dark circle is control qubit and the qubit with X gate is target qubit. This gates applies not gate on the target qubit only when control qubit is set to 1. This is shown below with an example.

$$|00\rangle \rightarrow |00\rangle$$
$$|01\rangle \rightarrow |01\rangle$$
$$|10\rangle \rightarrow |11\rangle$$
$$|11\rangle \rightarrow |10\rangle$$

This gate is used to entangle two qubits. Entanglement is only shown by quantum states and is one of the main property like superposition which is exploited in quantum algorithms and together both of them gives so called "Quantum Advantage". Fore more detailed explanation please refer to chapter 1 of Nielsen and Chuang (2011). All the gates shown above are reversible and when applied twice gives the initial state. This is one of the very important property of quantum gates which are hermitian. Some quantum gates are not hermitian and when applied twice do not give the input but for every quantum gate there exists a conjugate gate when applied in composition with gate gives the input. So, all quantum gates are Unitary.

9 Quantum Algorithms

So far we have studied what are quantum states and how we can go from one state to another state using quantum gates. In this section we will see how we can use the above knowledge to create quantum algorithms which give advantage over classical algorithms. First we will learn about Deutsch-Josza Algorithm which is very basic but one of the great algorithms which clearly shows quantum advantage. Then we will see Simon's Algorithm which is used to find period of periodic functions. This algorithm is used to break ciphers like Even-mansour, Fiestel, and Misty. Then we will see Shor's Algorithm this is used to break RSA in polynomial time. If you have a basic knowledge of quantum computing and you know these algorithms feel free to skip ahead to next section which discuss how different crypto systems work.

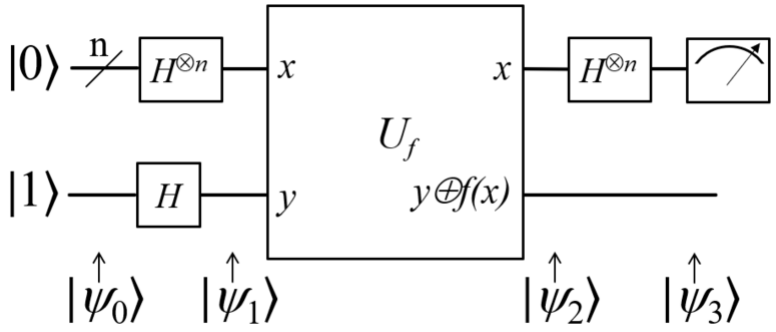


Figure 7: Duetsch-Josza Circuit. //Source: https://upload.wikimedia.org/wikipedia/commons/b/b5/Deutsch-Jozsa-algorithm-quantum-circuit.png

9.1 Deutsch-Josza Algorithm

Why should we attend parties? When physicist John Archibald Wheeler hosted a party at Austin in 1981, many physicists attended and one of them is David Deutsch. In the party a group discussed the foundations of computing and Deutsch immediately saw quantum theory would give an improvent to it. Few years later in 1985 he published Deutsch algorithm which clearly showed the advantage of quantum computing over fastest classical algorithm.Mar (2016).
Deutsch problem Given a function f from set of $N = 2^n$ bit strings x and returns 0 or 1. Given function is either **Constant** (All outputs are same.) or **Balanced** (Half of the input bitstrings give 0 and remaining half gives 1.). The goal is to decide wether the function is Constant or Balanced. Classically we need to compute the function $2^{n-1} + 1$ times to determine wether the function is Constant or Balanced in worst case.
Quantum Solution. Let us assume there is a box which contains circuit for $f(x)$. The box also contains extra circuits to satisfy the unitary property of quantum Operations. It takes input quantum states $|x\rangle$ and $|y\rangle$ and outputs quantum states $|x\rangle$ and $|y \oplus f(x)\rangle$. So, if we apply same box again the outputs become $|x\rangle$ and $|y \oplus f(x) \oplus f(x)\rangle = |y\rangle$. Therefore, the box is untiary. From now we will call this box as $U_f(x)$.

Let us take n qubits initialized to the state $|0\rangle$ then the n states combined represented as $|0\rangle^{\otimes n}$ and if we apply n Hadamard gates on them we get the state $\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n}$. Let this be the input state $|x\rangle$. Let us take state $|y\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. This can be obtained by applying "X-gate" and Hadamard gate on state $|0\rangle$. Let us send this inputs into the box and see what happens.

The quantum states in the circuit are as follows.

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$$

$$|\psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Let us consider $n = 1$ state for simplicity. Then,

$$|\psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\psi_1\rangle = \frac{1}{2}(|0\rangle |0\rangle + |1\rangle |0\rangle - |0\rangle |1\rangle - |1\rangle |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{2}(|0\rangle |0 \oplus f(0)\rangle + |1\rangle |0 \oplus f(1)\rangle - |0\rangle |1 \oplus f(0)\rangle - |1\rangle |1 \oplus f(1)\rangle)$$

$$|\psi_2\rangle = \frac{1}{2}(|0\rangle (|f(0)\rangle - \left|\bar{f}(0)\right\rangle) + |1\rangle (|f(1)\rangle - \left|\bar{f}(1)\right\rangle))$$

We rewrite it as

$$|\psi_2\rangle = \frac{1}{2}((-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle)(|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = \frac{1}{2}(-1)^{f(0)}(|0\rangle + (-1)^{f(1)-f(0)} |1\rangle)(|0\rangle - |1\rangle)$$

You can check if $f(x)$ is constant the value $(-1)^{f(1)-f(0)} = 1$ and if $f(x)$ is balanced $(-1)^{f(1)-f(0)} = -1$. Therefore, if $f(x)$ is constant

$$|\psi_2\rangle = \frac{1}{2}(-1)^{f(0)}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$

after applying Hadamard,

$$`\,|\psi_3\rangle = \frac{1}{\sqrt{2}}(-1)^{f(0)} |0\rangle (|0\rangle - |1\rangle)$$

If $f(x)$ is balanced,

$$|\psi_2\rangle = \frac{1}{2}(-1)^{f(0)}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)$$

after applying Hadamard,

$$`\,|\psi_3\rangle = \frac{1}{\sqrt{2}}(-1)^{f(0)} |1\rangle (|0\rangle - |1\rangle)$$

After measuring input state, if we get 0, then the function is constant and if we get 1, the function is balanced. Here, only $n = 1$ state is shown but same ideas can be extended to arbitrary n. his clearly took only one call to function $f(x)$. Therefore, we can clearly see how quantum algorithms give advantage over classical algorithms.

9.2 Simon's Algorithm

The first quantum algorithm to demonstrate an exponential speed-up vs the best classical algorithm in addressing a particular issue was Simon's algorithm, which was first presented in Simon (1997). This served as the inspiration for the quantum algorithms built on the quantum Fourier transform, including the most well-known one, Shor's factoring algorithm.

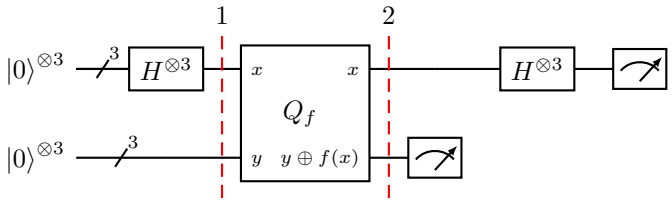
Simon's problem. If a function $f(x)$ is $2:1$ such that for every input x_1 in 2^n there exists x_2 which satisfy the condition $x_1 \oplus s = x_2$ and $f(x_1) = f(x_2)$. If the function $f(x)$ is given in a black box how many calls to the black box it will take to determine s. Here, we call s as period of function $f(x)$.

Example Let us consider the following function $f(x)$

$$f(000) = f(111) = 000$$
$$f(001) = f(110) = 001$$
$$f(010) = f(101) = 010$$
$$f(011) = f(100) = 011$$

Classical solution. Using brute force It takes $2^{n-1} + 1 = 2^{3-1} + 1 = 5$ calls to black box to determine s.

Quantum solution.



Starting State

$$|\psi_0\rangle = |0\rangle^{\otimes 3} |0\rangle^{\otimes 3}$$

State after First Hadamard Transforms

$$|\psi_1\rangle = \frac{1}{\sqrt{2^3}} \sum_{x \in \{0,1\}^3} |x\rangle |0\rangle^{\otimes 3}$$

State after applying the oracle

$$|\psi_2\rangle = \frac{1}{\sqrt{2^3}} \sum_{x \in \{0,1\}^3} |x\rangle |f(x)\rangle$$

State after measuring the second register

If the measurement gave $|001\rangle$

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$$

where,

$$f(x) = f(x \oplus s) = 001$$

State after final Hadamard

$$|\psi_3\rangle = \frac{1}{\sqrt{2^7}} \sum_{z \in \{0,1\}^3} [(-1)^{x.z} + (-1)^{(x \oplus s).z}] \, |z\rangle$$

Measurement of first 3 qubits of final state

Measurement of first 3 qubits of final state give information about s because, It will give output only if

$$(-1)^{x.z} = (-1)^{(x \oplus s).z}$$

which means:

$$\begin{aligned} x.z \bmod 2 &= (x \oplus s).z \bmod 2 \\ x.z \bmod 2 &= x.z \oplus s.z \bmod 2 \\ \implies s.z &= 0 \bmod 2 \end{aligned}$$

A string z will be measured, whose inner product with s = 0. Thus, repeating the algorithm $\approx$ n times, we will be able to obtain n different values of z and the following system of equation can be written:

$$\begin{cases} s.z_1 = 0 \bmod 2 \\ s.z_2 = 0 \bmod 2 \\ \vdots \\ s.z_n = 0 \bmod 2 \end{cases}$$

From which s can be determined, for example by Gaussian elimination.

Determining s

If first run gives the output $z_1 = 011$ then, Let s = abc
$z_1 = 011$

$$s.z_1 = 0 \bmod 2$$

$$b + c = 0 \bmod 2$$

either, bc = 00 or bc = 11.
If the second run gives the output $z_2 = 101$ then, $z_2 = 101$

$$s.z_2 = 0 \bmod 2$$

$$a + c = 0 \bmod 2$$

either ac=00 or ac =11.
If c = 0 $\implies$ a = 0 and b = 0 then s = 000 but we know s $\neq$ 000 therefore, c = 1 $\implies$ a = 1, b = 1 $\implies$ s = 111. We can clearly see, Simon's algorithm determined the period s in polynomial steps.

9.3 Shor's Algorithm

Shor's algorithm is a quantum solution to the problem of integer factorization, which is thought to be exponentially difficult for classical computers. Shor's technique is a ground-breaking method for number theory and encryption because it effectively factors huge integers into their prime factors in polynomial time by harnessing the power of quantum states and qubits. The quantum Fourier transform and period discovery are used in the algorithm to identify the period of a function associated to the integer to be factored. The number can then be effectively factored using the period. Shor's algorithm marks a significant advancement in the science of quantum computing and has the potential to revolutionise encryption and other fields of mathematics, even if it has not yet been applied on a broad scale due to the existing constraints of quantum hardware.
Algorithm As we have seen in section 2.2 how factorisation problem can be solved using order finding. We use Quantum algorithm to find the order.
Let us assume there exists a Unitary operator U when acted up on any state $|y\rangle$ results in the following state.

$$U|y\rangle = |ay \mod N\rangle$$

We call this unitary opeartor as modular multiplication operator from Markov and Saeedi (2015) as shown in figure [8]
For example. If $y = 1$, $a = 5$, $N = 21$ then,

$$\begin{aligned}
U|1\rangle &= |5 \mod 21\rangle = |5\rangle \\
U^2|1\rangle &= |25 \mod 21\rangle = |4\rangle \\
U^3|1\rangle &= |125 \mod 21\rangle = |2O\rangle \\
U^4|1\rangle &= |625 \mod 21\rangle = |16\rangle \\
U^5|1\rangle &= \left|5^5 \mod 21\right\rangle = |17\rangle \\
U^6|1\rangle &= \left|5^6 \mod 21\right\rangle = |1\rangle
\end{aligned}$$

Therefore, we can clearly see after applying the gate 6 times we get same $|1\rangle$ therefore, order of $5^r \mod 21$ is 6.

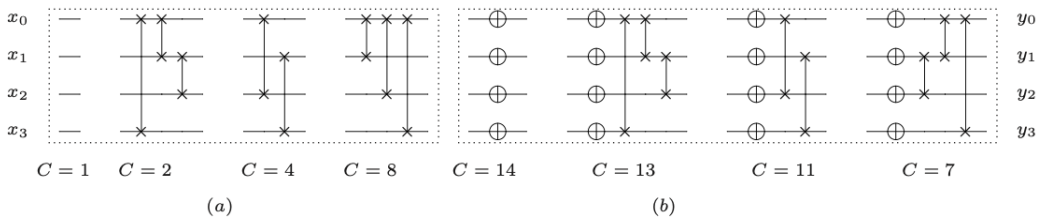


Figure 8: Circuit for $Cx \mod 15$.Source: Markov and Saeedi (2015)

If there is a quantum state which is created using superposition of states as shown below,

$$|u_0\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left|a^k \mod N\right\rangle \tag{1}$$

Now, if we apply the Unitary U on this state, $U|u_0\rangle = |u_0\rangle$ therefore, the state $|u_0\rangle$ is an eigen state of U with eigen value equal to one. Let us create an arbitrary state $|u_s\rangle$ as shown below,

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi iks/r} \left|a^k \mod N\right\rangle \tag{2}$$

It can be easily verified that the state $|u_s\rangle$ is also an eigen state of U with eigen value equal to $e^{2\pi is/r}$ that is

$$U|u_s\rangle = e^{2\pi is/r}|u_s\rangle$$

. Now, let us create a superposition of all the states of the form $|u_s\rangle$ that is

$$|\varphi\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle$$

As we know,

$$\sum_{k=0}^{r-1} e^{2\pi iks/r} = 0, \ \forall s \neq 0.$$

We can show that,

$$|\varphi\rangle = |1\rangle$$

and when Unitary is operated on this state $|1\rangle$,

$$U|1\rangle = e^{2\pi is/r}|1\rangle$$

where, s is a random integer between 0and $r-1$ because, the state $|1\rangle$ is superposition of all the states of the form $|u_s\rangle$ where s varies from 0 to $r-1$ refer to

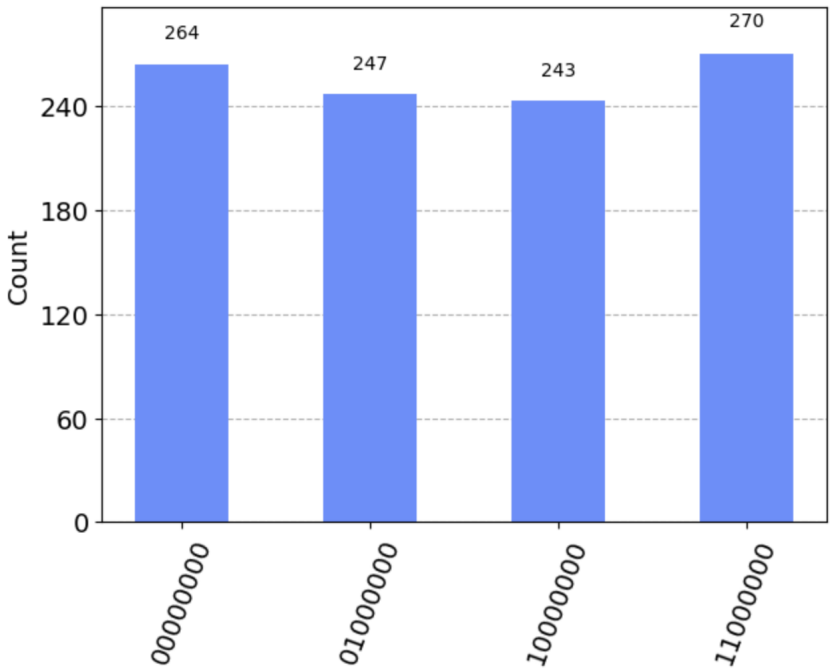


Figure 9: Output of shor's simulation

figure[9] which shows the output of the cirucit in [10].
Now, we measure the phase $\phi = \frac{s}{r}$ using Quantm Phase estimation algorithm D'Ariano et al. (1998) see figure [10] (which is one of the first algorithms we learn in quantum computing and can be easily learn from Nielsen and Chuang (2011) but beyond the scope of this report.) we can find r from ϕ and from that as shown in section 2.2.2 we can find the factors.

$$
\begin{aligned}
&\textit{RegisterOutput} && \textit{Phase} \\
&000000000(bin) = 000(dec) \rightarrow 000/256 = 0.00 \\
&111000000(bin) = 192(dec) \rightarrow 192/256 = 0.75 \\
&201000000(bin) = 064(dec) \rightarrow 064/256 = 0.25 \\
&310000000(bin) = 128(dec) \rightarrow 128/256 = 0.50
\end{aligned}
$$

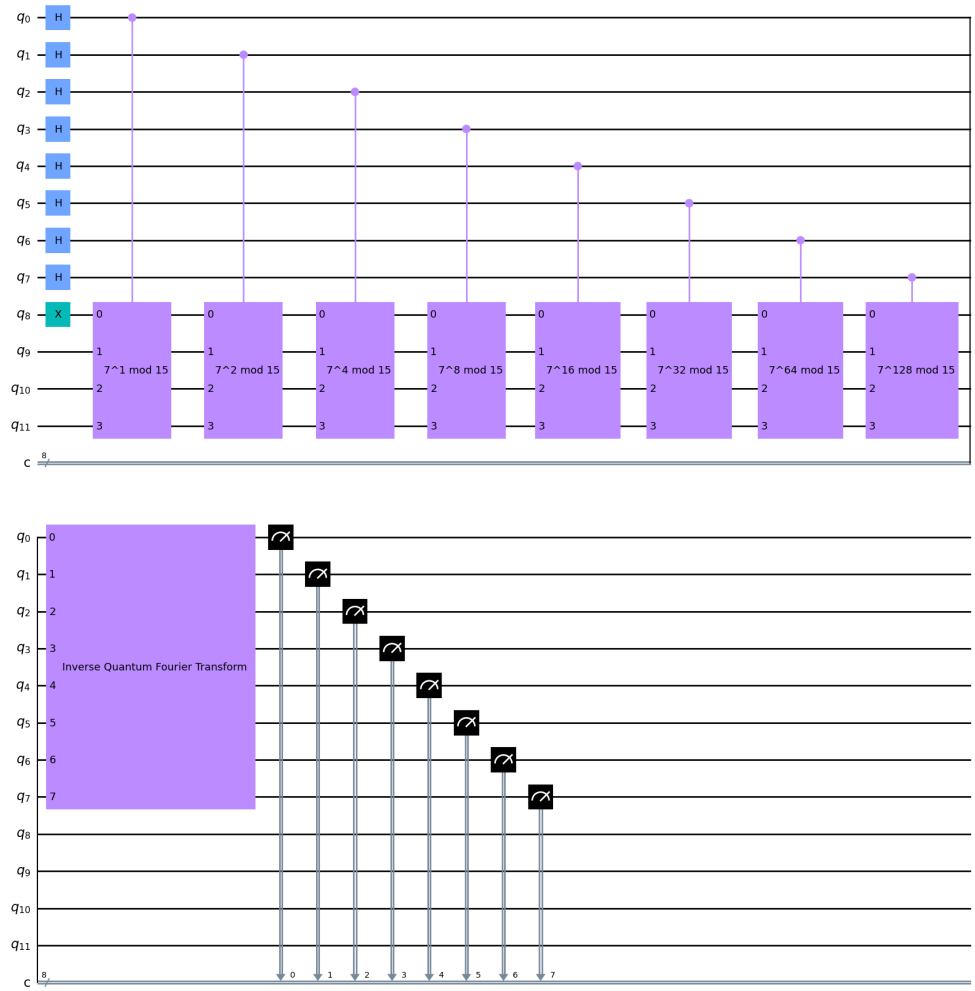


Figure 10: Quantum Phase estimation circuit for RSA

References

(2016). Preface to the second edition. In *Quantum Information Theory*, pages xi–xii. Cambridge University Press.

Canale, F., Leander, G., and Stennes, L. (2022). Simon's algorithm and symmetric crypto: Generalizations and automatized applications. Cryptology ePrint Archive, Paper 2022/782. `https://eprint.iacr.org/2022/782`.

D'Ariano, G., Macchiavello, C., and Sacchi, M. (1998). On the general problem of quantum phase estimation. *Physics Letters A*, 248(2):103–108.

Kuwakado, H. and Morii, M. (2012). Security on the quantum-type even-mansour cipher. pages 312–316.

Markov, I. L. and Saeedi, M. (2015). Constant-optimized quantum circuits for modular multiplication and exponentiation.

Nielsen, M. A. and Chuang, I. L. (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

Shor, P. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134.

Simon, D. R. (1997). On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483.

List of Figures