AN ADVANCED TECHNIQUE FOR PRIMALITY TEST WITH QUANTUM COMPUTATION AND ITS RELEVANCE IN CRYPTOGRAPHY

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology
in
Computer Technology

Submitted by

Ranadeep Bhattacharjee
Registration Number: 154186 of 2020-2021,
Examination Roll Number: M6TCT23019
Session: 2020-2023

Under the Supervision of

Prof. Debotosh Bhattacharjee
Department of Computer Science and Engineering
Jadavpur University,
188, Raja S.C. Mallick Rd,
Kolkata - 700032,
West Bengal, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that this is a bonafide record of the project entitled **"AN ADVANCED TECHNIQUE FOR PRIMALITY TEST WITH QUANTUM COMPUTATION AND ITS RELEVANCE IN CRYPTOGRAPHY"**, submitted by Ranadeep Bhattacharjee (University Registration No.: 154186 of 2020-2021, Examination Roll No.: M6TCT23019) is hereby approved of a creditable study of a technological subject carried out under my supervision and presented in a manner satisfactory to warrant its acceptance for partial fulfilment of the requirements of the degree of Master of Technology in Computer Technology. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Supervisor

. . . . . .

Prof. Debotosh Bhattacharjee
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

Countersigned

. . . . . .

Prof. Nandini Mukherjee
Head, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-32, India

. . . . . .

Prof. Ardhendu Ghoshal
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata-32, India

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval[1]

This is to certify that the thesis entitled "**AN ADVANCED TECHNIQUE FOR PRIMALITY TEST WITH QUANTUM COMPUTATION AND ITS RELEVANCE IN CRYPTOGRAPHY**", is a bonafide record of work carried out by Ranadeep Bhattacharjee in partial fulfilment of the requirements of the degree of Master of Technology in Computer Technology in Department of Computer Science and Engineering, Jadavpur University during the period of September 2021 to June 2023. It is understood that by this approval the undersigned do not necessarily endorse any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

. . . . . .
Signature of Examiner 1
Date:

. . . . . .
Signature of Examiner 2
Date:

Only in case thesis is approved[1]

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "**AN ADVANCED TECHNIQUE FOR PRIMALITY TEST WITH QUANTUM COMPUTATION AND ITS RELEVANCE IN CRYPTOGRAPHY**" contains literature survey and original research work by the undersigned candidate, as part of my Degree of Master of Technology in Computer Technology.

All information has been obtained and presented in accordance with academic rules an ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Ranadeep Bhattacharjee

Registration No.: 154186 of 2020-2021

Examination Roll No.: M6TCT23019

Thesis Title: AN ADVANCED TECHNIQUE FOR PRIMALITY TEST WITH QUANTUM COMPUTATION AND ITS RELEVANCE IN CRYPTOGRAPHY

. . . . . .
Signature with date

ACKNOWLEDGEMENT

I wish to express deep sense of gratitude to Prof. **Debotosh Bhattacharjee** for his continuous encouragement and guidance to accomplish my thesis work.

I desire to convey special thanks to all faculty members for extending their cooperation to perform the project work.

I am deeply indebted to my parents for their constant encouragement and enthusiasm time to time.

I am also thankful to my wife and child for their continuous support and encouragement.

I am extremely grateful to the eminent authors whose work I had the privilege to quote.

My sincere thanks to those who are directly or indirectly associated and extended their help and co-operation for bring out this thesis.

……

Ranadeep Bhattacharjee
Registration No.: 154186 of 2020-2021
Examination Roll No.: M6TCT23019
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32, India

TABLE OF CONTENTS

LIST OF ABBREVIATION USED

- Deutsch-Jozsa algorithm : DJ Algorithm
- LOCC : Local operations and classical communication
- U_{CNOT} : Controlled Not Gate
- H : Hadamard gate
- P : Phase Shift Gate
-  : Measurement
- tr [A] : Trace of Matrix A
- $|0\rangle$: $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- $|1\rangle$: $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

ABSTRACT

Several classical algorithms exist to detect prime numbers. All such algorithms are NP-hard. In the Quantum Computation domain also, a few algorithms like Shor's Algorithm exist, which are mainly based on the quantum version of Discrete Fourier Transformation. In this thesis a different approach (i.e. other than Fourier Transformation) has been made to detect Safe prime and Sophie Germain prime by establishing a correlation between balanced - constant function & prime number. Here we use the concept of balanced and constant function i.e. promise algorithm or more precisely, a type of Deutsch Jozsa (DJ) algorithm, a generalized version of Deutsch's algorithm. Shor's algorithm has been integrated with the quantum concept of Phi function to overcome its limitations. We have concentrated on detecting the prime property of a number i.e. **'a given number is prime or not'** without having any interest in identifying its factors.

Keywords

Safe prime and Sophie Germain prime; Balanced and Constant function; Quantum values of Euler's phi function; Safe Prime and Cryptography; Quadratic Residue; Primality Test;

OBJECTIVE(S):

1. Establishing a correlation between balanced -constant function & prime number
2. Developing a Quantum Algorithm to detect Safe prime and Sophie Germain prime
3. Overcome Limitation of Shor's Algorithm in case of odd order

CHAPTER-1 : RELEVANCE OF QUADRATIC RESIDUE IN PRIMALITY TEST

The basic idea behind such cryptography algorithm design is to develop a task for which the algorithm can provide an answer in polynomial time in a forward direction. But during decoding, i.e., computing the task in the reverse direction, no known algorithm exists to find an answer quickly, but if one is provided with information known as a password showing what the answer is, the answer can be easily verified. Classical Cryptographic algorithms are mainly based on randomly selecting one or more very large prime numbers and multiplying them to have a large composite number. While reversing the problem, we are faced with the task of determining whether a given large number is prime or not. There are no simple yet efficient means of accomplishing this task till now. So such cryptographic algorithms are safe yet.

In this chapter, we will develop a novel function that will operate based on the quadratic residue properties of the input number. In number theory, an integer, say q, is called a quadratic residue modulo n if it is congruent to a perfect square modulo n. Otherwise, q is called a quadratic nonresidue modulo n. The output of the developed function will be either balanced or constant, indicating the primality of the input number.

1.0 INTRODUCTION

A prime number is a whole number greater than 1 whose only factors are 1 and itself. It can't be expressed as a product of two natural numbers, essentially smaller than the number itself. The property of being prime is called **primality**. A natural number greater than 1 that is not prime is called a composite number. It may be noted that the natural number 1 does not fit in either of the two conditions. Hence, 1 is neither prime nor composite. Apart from the concept of prime numbers, there is another important property – relatively prime. Two numbers will be defined as relatively prime if and only if they do not share any common factor except 1. Notably, prime numbers are always relatively prime with any other numbers.

To better understand the further section, we will define three basic terms: the first is from Number theory, and the third is from Quantum Computation. These terms will be frequently used in subsequent sections and the next two chapters.

Definition 1: In number theory, Euler's totient function counts the positive integers up to a given integer *n* that are relatively prime to *n*. In other words, it is the number of integers *k* in the range $1 \leq k \leq n$ for which the greatest common divisor gcd(*n*, *k*) is equal to 1. The integers *k* of this form are sometimes referred to as totatives of *n*.

Definition 2: We say that $a \in Z$ is a quadratic residue mod n if there exists $b \in Z$ such that

$$a \equiv b^2 \bmod n.$$

If there is no such b we say that a is a quadratic non-residue mod n.

Example- Modulo 13, the quadratic residues are 1, 4, 9, 3, 12, and 10, while the quadratic nonresidues are 2, 5, 6, 7, 8, and 11.

Definition 3: Let a function say f(x) that takes as input x (n-bit binary string) and returns either 0 or 1 depending on the computation of the function for a given input.

If f(x) takes the same value on all inputs x, then it is a constant function.

If f(x) takes value 0 on exactly half of the inputs and 1 on the rest half of the inputs, then it is a balanced function.

Example- Let n be an odd number. x = {Positive integer less than n}.

Then f(x) = [(x / n)] is a balanced function with value = 0.

Then f(x) = (x % 2) is a balanced function. Say n = 5, then f (x) = 0 for 2 , 4 and f (x) = 1 for 1 , 3.

2.0 LITERATURE SURVEY:

There exist many tests to detect the primality of a number. Some of these are mentioned below.

Pocklington–Henry Cabourn Pocklington and Derrick Henry Lehmer invented the Lehmer primality test (1914).

Michael Rabin proposed the Miller-Rabin test (1976), slightly modifying a test by Mille. The test requires O(log n) arithmetic operations and is polynomial time.

Solovay-Strassen Test was proposed by Solovay and Strassen (1977). The test requires O(log n) arithmetic operations and is polynomial.

Agrawal, Kayal, and Saxena (2004) proposed AKS Test. The algorithm works in polynomial time.

Also, there are a lot more like the Fermat primality test Frobenius primality test, Baillie–PSW primality test, Adleman–Pomerance–Rumely primality test, elliptic curve primality testing, Lucas primality test, Lucas–Lehmer test (LLT), etc. But all such tests are probabilistic in nature.

In 1736, Leonhard Euler published a proof of Fermat's little theorem, which is the restriction of Euler's theorem to the case where n is a prime number. Subsequently, Euler presented other proofs of the theorem, culminating with his paper of 1763, in which he proved a generalization to the case where n is not prime. The theorem is known as **Euler's theorem** (also known as the Fermat–Euler theorem or Euler's totient theorem). The theorem states that if n and a are coprime positive integers, then $a^{\Phi(n)} = 1 \bmod (n)$

In 1748 he discovered and stated the famous **Eluer's criterion**. It states, " Let p be an odd prime and gcd (a,p) =1. Then a is a quadratic residue of p if and only if $a^{(p-1)/2} = 1 \pmod p$. This theorem has an important corollary which will be used later in this chapter. The **Corollary** states that - Let p be an odd prime and gcd (a,p) =1. Then a is a quadratic residue / non-residue of p if $a^{(p-1)/2} = 1 / -1 \pmod p$.

In 1798, Adrien-Marie Legendre defined an important symbol with the help of Quadratic residue. Let p be an odd prime and gcd (a,p) =1. The **Legendre symbol** (a/p) is defined by

$$\left(\frac{a}{p}\right) = \begin{cases} 1 \text{ if a is a quadratic residue of p} \\ -1 \text{ if a is a quadratic nonresidue of p} \end{cases}$$

An important application of the **Legendre symbol** (a/p) is based on the following **Corollary.** If p is an odd prime and a ∈ Z with gcd (a, p) = 1, then $x^2 \equiv a \bmod p$ either has no solutions or two incongruent solutions mod p. If we square all of {1, 2, ..., p − 1} mod p, we will get values in {1, 2, ..., p − 1}. We know that each result will occur twice and so there will be (p − 1)/2 quadratic residues. The remaining will be the quadratic nonresidues.

From the above discussion, we conclude an important theorem mentioned below.

Theorem 1.1: If p is an odd prime, then

$$f(p) = \sum_{a=1}^{(p-1)} \left(\frac{a}{p}\right) = 0$$

i.e. there are precisely (p-1)/2 quadratic residues and (p-1)/2 quadratic nonresidues of p. In other words, f (p) is a balanced function.

Bézout's identity (1779) States that- Let a and b be integers with the greatest common divisor d. Then there exist integers x and y such that ax + by = d. Moreover, the integers of the form az + bt are exactly the multiples of d.

Séroul (2000, p. 10) uses Bézout's theorem for the following two theorems.

1. Let a,b in Z be any two integers, then there exist u,v in Z such that au+bv=GCD (a, b).

2. Two integers a and b are relatively prime if there exist u,v in Z such that au+bv=1.

Now we will quote with proving two well-known theorems.

Theorem 1.2: If gcd (a,p) > 1, then for any $k \geq 1$; $a^k = 1 \bmod (p)$; cannot be held.

Proof: Let us assume that $a^k = 1 \bmod (p)$ for some k. then $a.a^{k-1} = 1 \bmod (p)$, which means

$a.a^{k-1} + pv = 1$. For some integer v. So, two numbers, namely a^{k-1} (say u) and v, such au + pv = 1 hold. So by Séroul's study (2000, p. 10) using Bézout's theorem as stated above, gcd (a,p) = 1 which is a contradiction, hence the result.

Theorem 1.3: For n >2, Φ(n) is an even integer.

Proof: Let k be a totative of *n* (In number theory, a totative of a given positive integer n is an integer k such that $0 < k \leq n$ and k is coprime to n), then by definition gcd (n,k)=1. Let, gcd(n, n−k) =d

So, d|n−(n−k) i.e. d|k. So we have d|k and d|n. But gcd(n,k)=1, so d=1. Thus, gcd(n,n−k)=1

This means if k is the totative of n, then (n – k) is also a totative of n. Thus totatives occur in pairs and hence the results.

3.0 OBJECTIVE :

1. Develop a function whose value could be either constant or balanced based on the Quadratic residue property of a given number

2. Setting the domain of the developed function

3. Establishing a relation between the primality of the input number and output type of the developed function

4. Validation of developed function with example

4.0 RATIONALE OF THE STUDY :

David Deutsch (1985) proposed a simple algorithm to explore the potentially greater computational power of a quantum computer as compared to a classical computer. The problem is defined as follows:-

Consider a function that takes as input n-bit strings and returns 0 or 1. Suppose we are promised that the function is either a constant function or a balanced function. The objective is to decide whether it is constant or balanced by making as few function evaluations as possible.

In the worst case, we will get $(2^n / 2) = 2^{(n-1)}$, continuous 0 / 1. And after that, if we receive the same value, the function is constant; otherwise, it is balanced. So, classically, it requires $(2^{(n-1)} + 1)$ function evaluations in the worst case. Using Quantum Computation, the question can be answered with just one function evaluation.

So if we succeed in developing our objective function, Quantum Computation can be successfully used to explore the primality problem and will add a new dimension to the cryptography of existing communication systems.

5.0 FUNCTION DEFINITION AND PROOF:

Given a number p for input. Let us define a function f (a,p) as

$= 1$; if $a^{(p-1)/2} \bmod (p) = 1$

$= -1$; else

with an added constraint **(p-1)/2 = prime greater than 2**.

and domain of a : **a ϵ [2, p-2]**.

We have excluded 1 and p-1 intentionally for our proof. From simple calculation it follows that, $a^{(p-1)/2} \bmod (P) = 1$ for a = 1, and -1 for a = p-1.

In the next paragraph, we will prove that the defined function meets our objective(s).

Let Φ(p) be the Eluer Phi function with an upper limit (p-1). Given that(p-1)/2 is prime. Hence if we compute **g.c.d. (Φ(p), (p-1)/2) then two cases may arises only; g.c.d. (Φ(p), (p-1)/2) =1 or (P-1)/2.**

Case 1:g.c.d. (Φ(p), (p-1)/2) =1; Here two cases can arise :-

a) **if gcd (a,p) = 1** then by **Euler's theorem** (1736) $a^{\Phi(p)} = 1 \bmod (p)$. But we have g.c.d. (Φ(p), (p-1)/2) =1. Let r be the order of a mod p. Then r is a factor of Φ(p). Now, if $a^{(p-1)/2} = 1 \bmod (p)$, then r is also a factor of (p-1)/2. But g.c.d. (Φ(p), (p-1)/2) =1**. So,** $a^{(p-1)/2} \neq 1 \bmod (p)$

b) **if gcd (a,p) ≠ 1,** then by virtue of Theorem 2 $a^{(p-1)/2} \neq 1 \bmod (p)$

Combining two cases f(p) = -1 for all values of a ϵ [2, p-2]. **So f (p) will be a constant function in this case and clearly, p is composite.**

Case 2: g.c.d. (Φ(p), (p-1)/2) = (p-1)/2; Here two cases can arise :-

a) Φ(p) = (p-1)/2 : In this case, half of the numbers in domain [2, p-2] are relatively prime to p. But (p-1)/2 is a prime number greater than 2, which is always odd. But by virtue of theorem 3, Φ(p) is even. So this case is not possible.

b) Φ(p) = (p-1) : In this case, p is prime and by virtue of Eluer's criterion (1748) for quadratic residue in combination with Theorem 1, $a^{(p-1)/2} = 1 \bmod (p)$ for half of the values in the range [2, p-2] and $a^{(p-1)/2} = -1 \bmod (p)$ for other values in the range [2, p-2]. **So, f (p) will be a balanced function in this case.**

So, once we find the function is constant, we can immediately conclude that the input number p is not a Prime. If we find the function is balanced, then the number p is Prime. The limitation is (p-1)/2 has to be prime and greater than 2.

6.0 VALIDATION OF THE FUNCTION WITH EXAMPLE:

For case 1:

Let us take p = 15, as an example. Clearly, (15 -1)/2 =7 is a Prime Number greater than 2. Now our task is to find if 15 is prime or not.

Table 1 : *Validation of Function with Example (Case 1)*

a	**$a^{(p-1)/2}$**	**$a^{(p-1)/2}$ mod (p)**	**f (a, p) = f (15)**
1	1	1	1
2	128	8	-1
3	2187	12	-1
4	16384	4	-1
5	78125	5	-1
6	279936	6	-1
7	823543	13	-1
8	2097152	2	-1
9	4782969	9	-1
10	10000000	10	-1
11	19487171	11	-1
12	35831808	3	-1
13	62748517	7	-1
14	105413504	14	-1

From the above table, we find that the values of f (p) where a ϵ [2, p-2] is constant (-1), i.e., f (p) is a constant function. So we immediately conclude that 15 is not a Prime Number.

For case 2:

Let us take p = 11, as an example. Clearly, (11-1)/2 =5 is a Prime greater than 2. Now our task is to find if 11 is prime or not.

Table 2 : *Validation of Function with Example (Case 2)*

a	a^(p-1)/2	a^(p-1)/2 mod (p)	f(p) = f (15)
1	1	1	1
2	32	10	-1
3	243	1	1
4	1024	1	1
5	3125	1	1
6	7776	10	-1
7	16807	10	-1
8	32768	10	-1
9	59049	1	1
10	100000	10	-1

From the above table [Table 3 : Validation of Function with Example (Case 2)], we find that the values of f (p) where a ϵ [2, p-2] is -1 for p = 2, 6, 7, 8 & is +1 for p = 3, 4, 5, 9 i.e. f (p) is a balanced function. So we can conclude that 11 is a prime.

Chapter-2 : QUANTUM NUMBERS FOR EULER PHI FUNCTION - A MODIFIED VERSION OF SHOR'S ALGORITHM

Several classical algorithms exist to detect prime numbers. Some are probabilistic, like the Fermat primality test, Miller-Rabin primality test,t, etc., while others are based on trial division, but all such algorithms are NP-hard.

The classical algorithm can efficiently detect prime numbers from a given range of natural numbers. Mostly these algorithms are based on Sieve approaches, i.e., identification of composite numbers and then eliminating them from a given set of numbers to mark the rest as primes.

In the Quantum Computation domain, primality algorithms like Shor's Algorithm and modification are mainly based on the quantum version of Discrete Fourier Transformation. In this chapter, we have tried to explore the power of Quantum Computation to break the irreversibility feature of the multiplication property. We have concentrated on detecting the prime property of a number, i.e., whether **a given number is prime or not,** without having any interest in identifying its factors.

In number theory, Euler's phi function counts the positive integers up to a given integer n that are relatively prime to n. In other words, it is the number of integers k in the range $1 \leq k \leq n$ for which the greatest common divisor $\gcd(n, k)$ is equal to 1. The integers k of this form are sometimes referred to as totatives of n. Leonhard Euler introduced the function in 1763. It has many applications in primality tests by classical computation. Though in a quantum domain, such application is not yet studied. Quantum computers have the potential to provide computational power on a scale that traditional computers cannot ever match, mainly due to the concepts of superposition and entanglement. They can solve complex problems. The more complex a problem, the harder it is for even a supercomputer to solve. Some Basic Quantum Terminologis used in this paper are described below.

Qubits

Traditional computers are built on bits. These bits (short for binary digits) are the basic units of information in computing, where two distinct configurations can be measured. They can be thought of as on or off, up or down, or encoded in binary as either 0s or 1s.

In quantum computing, quantum bits or qubits form the basics of how these computers work. These qubits can be made from quantum-mechanical systems that can have two states. For example, the spin of an electron can be measured as up or down, or a single photon is either vertically or horizontally polarised.

Superposition

Unlike traditional computing bits, which can be either 0s or 1s, qubits can exist as either 0s or 1s or a mix of both simultaneously. This phenomenon, known as a state of superposition, means that all combinations of information can exist at once.

When qubits are combined, this ability to hold all possible information configurations simultaneously means that complex problems can be represented as far more manageable than traditional computing methods.

An outline of Shor's Algorithm is mentioned here for ready reference. Shor's algorithm factors an integer N in two steps. The quantum step computes the order of a mod N where a is relatively prime to N. The classical step uses this order to factor N. Descriptions of the classical step require the order, r, to be even and that $a^{r/2} = 1 \bmod N$. If r is odd or $a^{r/2} = -1 \bmod N$, the quantum step is repeated.

1. LITERATURE SURVEY:

Bastos, D. C., Kowada, L.A.B.(2021) explained the possibility of the success of Shor's Algorithm if given the prime factorization of a composite N.

Maria Sabani, Ilias Galanis, Ilias Savvas, and Georgia Garani (2021) Consider some reliability issues of quantum devices in order to explore the potentiality of Shor's algorithm.

Anna M. Johnston 2017 observed that using any odd prime divisor of the order in the classical step minimizes the need to repeat the quantum step. Not only is there no need to throw out odd orders returned by the quantum step, but a single returned order, depending on its factorization, allows multiple attempts at factoring N.

Edward Gerjuoy (2005) explains, in a fashion comprehensible to the nonexpert, the RSA encryption protocol; the various quantum computer manipulations constituting the Shor algorithm; how the Shor algorithm performs the factoring; and the precise sense in which a quantum computer employing Shor's algorithm can be said to accomplish the factoring of very large numbers with less computational effort than a classical computer.

In 1736, Leonhard Euler published a proof of Fermat's little theorem, which is the restriction of Euler's theorem to the case where n is a prime number. Subsequently, Euler presented other proofs of the theorem, culminating with his paper of 1763, in which he proved a generalization to the case where n is not prime. The theorem is known as **Euler's theorem** (also known as the Fermat–Euler theorem or Euler's totient theorem).The theorem states that if n and a are coprime positive integers, then $a^{\Phi(n)} = 1 \bmod (n)$

Jie Fang and Chenglian Liu in 2018 observed that if $p > 1$ is a composite number , then $\Phi(p) \leq p - \sqrt{p}$

As per the general procedure of Shor's Algorithm, we must first select a number, say a1, which is relatively prime to the given number, say p, and then apply QFT to find out the order of a1, say r1. If r1 is odd or $(a1)^{r1/2} \bmod p = -1$ then we have to choose another number, say a2 and repeat the same procedure to find r2 until an even order is obtained.

2. LIMITATION OF SHOR'S ALGORITHM:

During iteration at the quantum step, if we get an order, say r1, such that either r1 is odd or $(a1)^{r1/2} \bmod p = -1$; iteration of Shor's Algorithm increases.

Let us take an example number, say p = 314191. We select a random a1 = 101. We found gcd (314191, 101) = 1. Using the Quantum algorithm, we found r1 = 4347, which is odd. So we have to check with another relative prime, say a2.

We select another random a2 = 5. We found gcd (314191, 5) = 1. Using the Quantum algorithm, we found r2 = 63, which is odd. So we have to check with another relative prime, say a3. Thus steps of iteration increase without having any outcome.

3. PROPOSED ALGORITHM:

We have used two important theorems to establish the correctness of our algorithm. The theorems are stated and mentioned below:

Theorem 2.1:

Let p be any given number whose primality test is to be done. a < p be a number such that gcd (a, p) = 1. Let r be the order of a mod p. Then value of Φ (p) could not be continuous but a multiple of r.

Proof:

Euler's theorem states that, if p and a are coprime positive integers, then $a^{\Phi(p)} = 1 \bmod (p)$. But r is the order of a mod (p). So $a^{r} = 1 \bmod (p)$. Now if Φ (p) is not a multiple of r then let Φ (p) = kr + c, where k, c are positive integers with 0 < c < r.

Now, $a^{\Phi(p)} = 1 \bmod (p)$, i.e. $a^{kr + c} = 1 \bmod (p)$, i.e. $(a^{r})^{k} . a^{c} = 1 \bmod (p)$. But $a^{r} = 1 \bmod (p)$. So $a^{c} = 1 \bmod (p)$ where 0 < c < r. It contradicts that r is the order of a mod p. Hence Φ (p) is a multiple of r.

Theorem 2.2:

Let p be any given number whose primality test is to be done. a_1, a_2, a_3.. < p be a number such that gcd (a_n, p) = 1. Let r_n be the order of a_n mod p. Then value of Φ (p) could not be continuous but a multiple of lcm ($r_1, r_2, \ldots r_n$).

Proof:

From theorem 1, Φ (p) is a multiple of r_1 for a_1. Similarly, Φ (p) is a multiple of r_2 , r_3 ….. r_n. So $r_1, r_2, r_3 \ldots r_n$ are factors of Φ (p) and $r_1, r_2, r_3 \ldots r_n$ are not necessarily relatively prime to each other. Also they are not exhaustive factors. So we can conclude Φ (p) is a multiple of lcm ($r_1, r_2, \ldots r_n$).

In this chapter, we suggest a slight modification if the outcome of the quantum step i.e. order, say r1 is odd. Instead of rejecting it we can use it to predict Φ (p) i.e. Eluer phi function of p. Here p is the given number or input preciously on which primality test is being done. Modification is done on the classical portion of Shor's algorithm.

Step 1. First, select a number, say 'a', which is relatively prime to the given number, and say 'p'.

Step 2. Apply Quantum Part to find order of 'a'. Let 'r' be the order of 'a' mod 'p.'

Step3. Let us define a flag variable r_i correspond to i^{th} iteration as lcm (r, $r_{(i-1)}$) . $r_{(i-1)}$ is the value of the flag in $(i-1)^{th}$ iteration with intialisation r_0 =1.

Step 3. *Estimation of Quantum nembers for Φ (p)* - Value of Φ (p) could not be continuous but a multiple of r_i (Theorem 2). Thus value of if Φ (p) could be r_i, 2 r_i, 3 r_i …….. . [there is a limitation on the upper bound of Φ (p) . From the definition of the Euler phi function, Φ (p) < p. However, if p is a composite number, then Fang and Liu's study (2018) revealed that the value of Φ(p) is $\leq p - \sqrt{p}$.]

Step 4. Compute gcd (r, n-1). If gcd (r, n-1) ≠ r, we can immediately conclude that p is composite. If r is even or $(a)^{r/2}$ mod p ≠ -1, then we can go with Shor's algorithm to identify the primality of p.

Thus we can predict values of Φ(p) with very high probability and with iteration, the probability increases.

Thus for composite p,

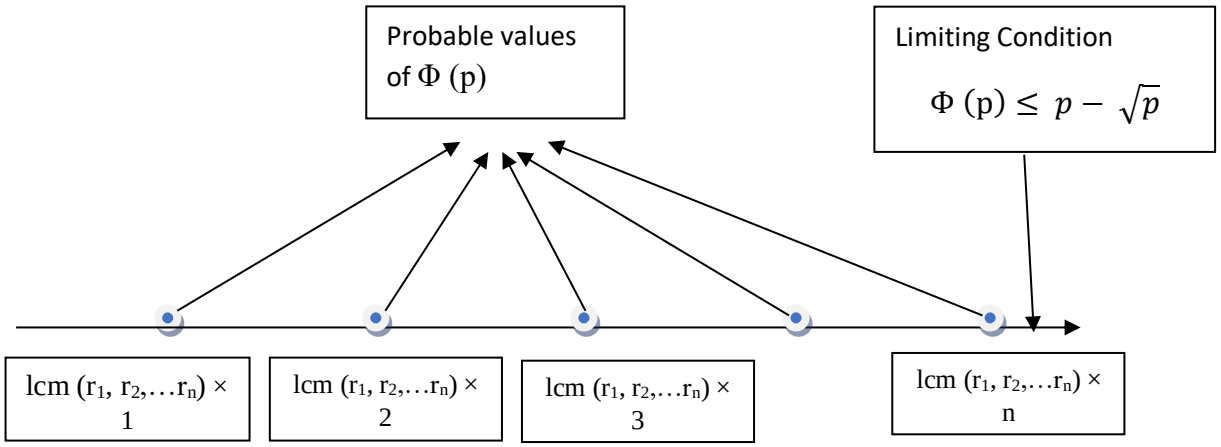


Fig 1 : *Probable Quantum States of Φ (p)*

Example:

In our case, we have taken the previous p = 314191 as an example.

Step 1. We select a number, say 101, which is relatively prime to 314191.

Step 2. Order of 101 mod 314191 is 4347.

Step 3. *Estimation of Quantum nembers for Φ (p)* - we can conclude by theorem 2.2 that Φ(314191) is divisible by 4347. So Φ(314191) can have values = (4347× k) , where k = 1, 2, 3 …… is Quantum Numbers of Orbit with constraint Φ(314191) < 314191.

Step 4. g.c.d. (4347, 314190) = 9 ≠ 4347. So we can verify that the number is not prime and is composite.

So we can update the upper bound Φ(314191) < $314191 - \sqrt{314191}$ = 314191-560 = 313631.

4347

4347×2= 8694

4347×3= 13041

4347×4= 17388

……..

……..

4347×72 = 312984

Probable Values of Φ(314191) in Iteration 1

4347×73= 317331 < 313631 -------→ Limit Exist [Termination]

Fig 2 : *Probable Quantum States of Φ (314191) in First Iteration*

So, the upper bound of n is 72 in the first iteration, i.e., 72 possible values of Φ (314191).

Validation: Indeed 314191 = 829 × 379; 829 and 379 are prime. So Φ (829) = 828 and Φ (379)= 378. So Φ (314191) = 828 × 378 = 312984. This value is divisible by 4347 and satisfied with the quantum number k = 72, i.e. 72 × 4347 = 312984.

Further, we can have another iteration if we want to increase accuracy.

Step 1. We select a number, say 43, which is relatively prime to 314191.

Step 2. Order of 43 mod 314191 is 756.

Step 3. *Estimation of Quantum nembers for Φ (p)* - we can conclude by theorem 2.2 that Φ(314191) is divisible by l.c.m. (756, 4347) = 17388. So Φ(314191) can have values = (17388× k) , where k = 1, 2, 3 …… is Quantum Numbers of Orbit with constraint Φ(314191) < 314191.

17388

17388×2= 34776

17388×3= 52164

17388×4= 69552

……..

……..

17388×18 = 312984

Probable Values of Φ(314191) in Iteration 2

17388 ×19= 330372 < 313631 -------→ Limit Exist [Termination]

Fig 3 : *Probable Quantum States of Φ (314191) in Second Iteration*

So upper bound of n is reduced to 18 from 72 as in the first iteration, i.e., there are only 18 possible values of Φ (314191).

Chapter-3 : QUANTUM ALGORITHM TO DETECT SOPHIE GERMAIN PRIME

Safe primes were introduced by Sophie Germain in her study of Fermat's last theorem (Laubenbacher and Pengelley, 2010). A Sophie Germain Prime is a prime number that satisfies the following property: when we multiply it by 2 and add 1, we get another prime number. Finally, the new prime numbers generated in such a way are called Safe Primes. i.e. Safe Prime = (2 × Sophie Germain Prime) + 1

Safe Prime has a special significance in cryptography. These primes are safer to create a cryptographic key compared to other primes. We will explore it in subsequent sections of this chapter. In this chapter, we established a Quantum Algorithm to detect Safe and Sophie Germain prime. Here we use the concept of balanced and constant function i.e. promise algorithm or more precisely a type of Deutsch Jozsa (DJ) algorithm, a generalized version of Deutsch's algorithm.

1.INTRODUCTION:

Multiplication of two prime numbers is very fast from a computational point of view compared to a given number and to identify the prime numbers that are factors of the given number. Basically it is a NP-hard problem by classical computation. So the multiplication process is NP-hard in the backward direction; in other words, multiplication is deterministic computation in forward direction but non deterministic in the backward direction. An example is illustrated below:

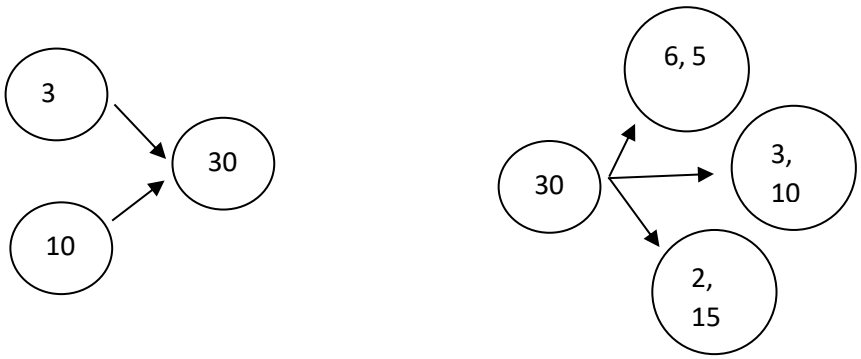


Fig 4: *Illustration of Irreversibility of Multiplication Process*

This actually means that conducting prime factor decomposition of a large number, especially if its prime factors are also large, is not always easy. Many encryption algorithms are based on the irreversibility property of multiplication operations.

2. LITERATURE SURVEY:

John Pollard 1974 described Pollard's p − 1 integer factorization algorithm as only suitable for integers with specific types of factors. This method discovers a prime factor p of an integer n whenever p − 1 has only small prime factors i.e. (p-1) is a x- smooth, where x is a small prime.

Leonard Adleman (1976) gives us an idea about the n-smooth of a number. In number theory, an n-smooth number is an integer whose prime factors are all less than or equal to n. For example- 121 (11×11) is an 11-smooth, whereas a much bigger number, say 220500 ($7^3 \times 3^2 \times 5^2$) is a 7-smooth.

Qingbo Wang (2017) identifies that Prime numbers are universally used in cryptography, but some are safer to use from a cryptographic point of view than others and this does not depend on the digit length of the prime number.

For example, a number, say m, has two prime factors, p and q. As long as (p-1) (or (q-1)) is a product of relatively small primes, Pollard's (p – 1) algorithm allows us to relatively easily find p and q even when p (or q) are themselves large primes. On the other hand, if (p-1) or (q-1) themselves have at least one very large prime factor, the algorithm would not perform well. It means that a prime number, say m, required to be used in cryptography must have the property that (m-1) is an x-smooth, with x being a large prime.

The immediate question arises what is the maximum value of x? If m is an odd prime (indeed, it is for cryptography as only even prime number 2 is too easy to decode) then obviously (m-1) is even. This implies that 2 is a prime factor of (m-1). Let (m-1)/2 = t. or (m – 1) = 2t. Now maximum smoothness for (m-1) is t smooth and occurs when t is a prime. So m is a safe prime if m = 2t + 1, where t is a prime, which is indeed the definition of a safe prime as proposed by Marie-Sophie Germain long before the invention of cryptography!

Thus, safe primes are fundamental in the field of cryptography, which means that the Sophie Germain Primes form the foundation that underlies today's security systems, and a little quantum computation advancement in this field could be the beginning of the end for modern computer encryption.

3. DEUTSCH–JOZSA ALGORITHM

The Deutsch–Jozsa algorithm is a deterministic quantum algorithm proposed by David Deutsch and Richard Jozsa in 1992 with improvements by Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca in 1998.

The Deutsch-Jozsa problem is defined as follows- Consider a function f(x) that takes as input n-bit strings x and returns 0 or 1. Suppose we are promised that f(x) is either a constant function or a balanced function. The goal is to decide whether f is constant or balanced by making as few function evaluations as possible.

Classically, it requires $(2^{n-1}+1)$ function evaluations in the worst case. Using the Deutsch-Jozsa algorithm, the question can be answered with just one function evaluation. We require two quantum registers, one of size n, i.e., composed of 1 q-bit (Input Register), and the other of size 1, i.e., composed of 1 q-bit (Ancilla qubits Register).

The algorithm in a nutshell:-

Step 1. Initialize both registers in the zeros state (Annexure IV).

Step 2. Apply the Hadamard gate to both registers (Annexure-II).

Step 3. Apply the Oracle circuit (Annexure III).

Step 4. Apply the Hadamard gate to the input register (Annexure-II).

Step 5. Measure Input register. Let $y = (y_1, y_2, \ldots y_n)$ be the list of measurement outcomes. We find that f is a constant function if y is the all-zeros string.

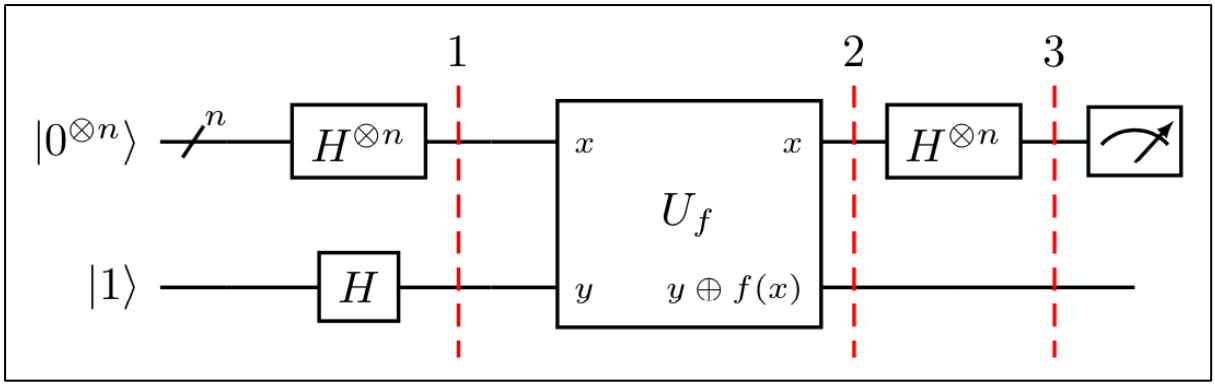


Fig 5 : *Generic circuit for the Deutsch-Jozsa algorithm*

4. EXISTING ALGORITHM TO DETECT SAFE PRIME:

Clasically safe primes are detected by simple primality tests, and an advanced method specifically for safe primes is not available yet. However, Pocklington's criterion can be used to prove the primality of 2p + 1 subject to proven the primality of p with logarithmic time complexity.

In the Quantum domain, Grosshans et all (2017) applied Shor's factoring algorithm for safe semiprimes by improving the classical routines, using simple classical mathematics, which can speed up quantum factoring by reducing the dependence on the quantum circuit. It is against based on QFT and has many limitations. Also complexity is not constant time.

5. PROPOSED ALGORITHM:

Problem Statement – Given a Prime number n, we have to check if n is Sophie Germain Prime or in other words, p = (2n+1) is safe (prime) or not.

Let us define a function f (a) as

= 1; if $a^{(p-1)/2} \bmod (P) = 1$

= 01; else

with an added constraint **(p-1)/2 = prime greater than 2** and domain of a : **a ϵ [2, p-2]**.

From Chapter 1, if this function is constant, we can immediately conclude that the input number p is not a Prime; else (if the function is balanced), then the number p is a Prime.

So the Problem statement is reduced to the identification of the constant or balanced function of f (a).

This new problem statement fits with the **Deutsch-Jozsa problem** defined above. The domain of function f defined above is [2, p-2]. But in the general **Deutsch-Jozsa** algorithm, we take $[0, 2^{(n-1)}]$. So modification in the Quantum circuit is required so that the superposition of the n qubit register starts from 2 (lower limit) to (p-2) (upper limit). An illustration is mentioned below:

EXAMPLE

Case-1

Let us take n= 3 and p= (3×2 + 1) = 7, an example. Clearly 3 is a Sophie Germain Prime or in other words, 7 is a safe prime.

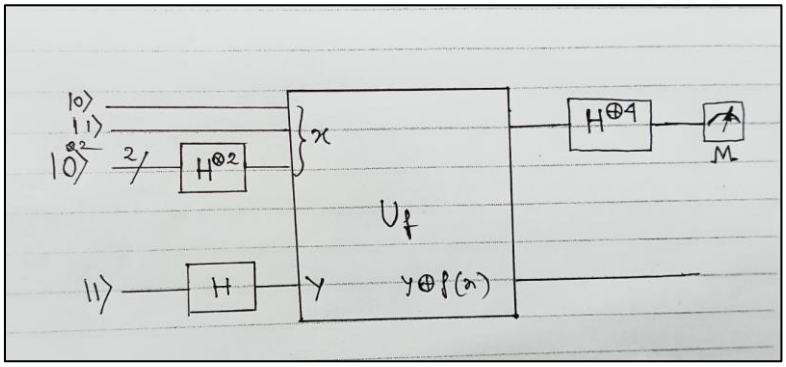


Fig 6: *Quantum Circuit for n =3 i.e. p =7*

In the above circuit, x is the input register (4 qubits), and y is the ancilla register (1 q bit).

Next, we initialize 0 to the first, third, and fourth qubits, 1 to the second qubit of the x registers, and 0 to the y register.

Next, we apply Hadamad transformation to the three and four qubits of the x and y registers. So x register become superimposed with value |0100>, |0101>, |0110> and |0111> , i.e. 2, 3, 4, 5. Just as required for us 2 to (p-2) where p =7.

Next, Oracle evaluation is done, and the result is mentioned in the table below:

Table 3*: Result of Oracle Evaluation for n=3 and p=7*

n=3	**p=7**		
a	**$a^{(p-1)/2}$**	**$a^{(p-1)/2}$ mod (p)**	**f(a)**
1	1	1	1
2	8	1	1
3	27	6	0
4	64	1	1
5	125	6	0
6	279936	6	-1

From the above Table-1, we find that the values of f (a) where a ϵ [2, p-2] is 0 for a = 3, 5 and is 1 for a= 1,4. So f(a) is a balanced function indicating 3 as a Sophie Germain Prime, i.e., 7 as a safe prime.

Case-2

Let us take n= 7 and p= (7×2 + 1) = 15, an example. 7 is not a Sophie Germain Prime, or 15 is not a safe prime, as 15 is a composite number.

Table-4*: Result of Oracle Evaluation for n=7 and p= 15*

n=7	**p=15**		
a	**$a^{(p-1)/2}$**	**$a^{(p-1)/2}$ mod (p)**	**f(a)**
1	1	1	1
2	128	8	0
3	2187	12	0
4	16384	4	0
5	78125	5	0
6	279936	6	0
7	823543	13	0
8	2097152	2	0
9	4782969	9	0
10	10000000	10	0
11	19487171	11	0
12	35831808	3	0
13	62748517	7	0
14	105413504	14	-1

From the above Table-2, we find the values of f (a) where a ϵ [2, p-2] is constant i.e. 0. So we can conclude that seven is not a Sophie Germain Prime, i.e., 15 is not a Safe Prime.

CONCLUSION

Many encryption algorithms are based on the irreversibility property of multiplication operations. Thus a little development in this field could be the beginning of the end for modern computer encryption. In chapter 1, we have proposed a function that can be used as a classifier to identify a given number, say n as an example, is prime or not with a constraint that (n-1)/2 must be a prime with a value greater than 2. In chapter 3, we have discussed that primes which satisfies the said constraint are safe from cryptographic point of view. The function is constant if the input number is Composite, while the function is balanced if the input number is Prime. The above discussion creates a bridge between the Primality problem and the Deutsch-Jozsa algorithm, one of the first quantum algorithms with a nice speedup over its classical counterpart and we conclude chapter 1 with a hope to break the irreversibility feature of multiplication property with the power of Quantum Computation in further advancement which have been explored in Chapter 3.

In chapter 2, we have proposed a modification of Shor's algorithm to predict the quantum values of Euler's phi function. In particular, if we get any relative prime of p, having odd order, then by conventional Shor's algorithm, we have to neglect it. However, it is an important observation and without neglecting it, we can apply a modified version of Shor's algorithm, as stated here, to predict the quantum values of Euler's phi function with a desired level of probability and also indirectly detect its primality.

Chapter 3 describes a novel approach to detect Safe prime by quantum algorithm (Deutsch-Jozsa algorithm). The algorithm can detect a safe prime in a single pass, i.e., the time complexity is constant, having an advantage over existing classical algorithms with logarithmic time complexity. However, efforts are required to develop the quantum circuit, particularly to set the upper and lower limit of the function variable. Further improvement in this direction will make the computation more efficient and feasible.

REFERENCES

1. Bézout, É. (1779), 'Théorie générale des équations algébriques', Paris, France: Ph.-D. Pierres

2. Bastos, D. C., Kowada, L.A.B., 2021, 'How to detect whether Shor's algorithm succeeds against large integers without a Quantum Computer, www.sciencedirect.com

3. D. Deutsch (1985), 'Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer,' Proceedings of the Royal Society of London A, 400, pp. 97–117

4. David M. Burton (2013), 'Elementary Number Theory', Mc Graw Hill Education, Third Edition, pp- 172, 177-178

5. David Deutsch & Richard Jozsa *(1992). "Rapid solutions of problems by quantum computation." Proceedings of the Royal Society of London A.* ***439*** *(1907):*

6. Edward Gerjuoy, 2005, 'Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers', American Journal of Physics 73, 521 (2005); https://doi.org/10.1119/1.1891170

7. Frédéric Grosshans, Thomas Lawson, François Morain, and Benjamin Smith, "Factoring Safe Semiprimes with a Single Quantum Query," (Dated: March 3, 2017)

8. J. M. Pollard, "Theorems on factorization and primality testing," Mathematical Proceedings of the Cambridge Philosophical Society, vol. 76, pp. 3049–3052, 1974.

9. Jie Fang, Chenglian Liu, 'A Generalize Estimating the φ(n) of Upper/Lower Bound to RSA Public Key Cryptosystem,' International Journal of Network Security, Vol.20, No.2, PP.332-336, Mar. 2018 (DOI: 10.6633/IJNS.201803.20(2).14)

10. Johnston , A. M., 'Shor's Algorithm and Factoring: Don't Throw Away the Odd Orders,' February 6, 2017

11. Leonhard Euler (1736), 'Theorematum quorundam ad numeros primos spectantium demonstratio', Commentarii academiae scientiarum Petropolitanae, 8 : 141–146, (presented: August 2, 1736; published: 1741)

12. L. Euler (published: 1763), 'Theoremata arithmetica nova methodo demonstrata', Novi Commentarii academiae scientiarum Petropolitanae, 8 : 74–104. Euler's theorem appears as "Theorema 11" on page 102

13. Legendre, A. M. (1798). 'Essai sur la théorie des nombres' Paris. p. 186

14. Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. Annals of Mathematics, 160(2):781–793, 2004

15. M. O. Rabin. Probabilistic algorithm for testing primality. J. Number Theory, 12:128–138, 1980.

16. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge U.P., 2000.

17. Maria Sabani, Ilias Galanis, Ilias Savvas, Georgia Garani (2021), 'Implementation of Shor's Algorithm and Reliability of Quantum Computing Devices', PCI '21: Proceedings of the 25th Pan-Hellenic Conference on Informatics November 2021, Pages 392–396. https://doi.org/10.1145/3503823.3503895

18. Pocklington, Henry C. (1914–1916). "The determination of the prime or composite nature of large numbers by Fermat's theorem." Proceedings of the Cambridge Philosophical Society. 18: 29–30. Retrieved 2022-06-22.

19. R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. SIAM Journal on Computing, 6:84–86, 1977.

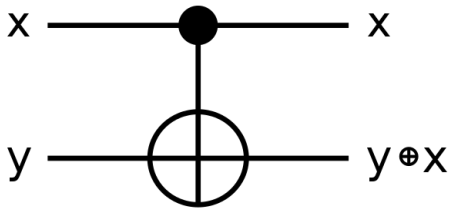
20. R. Laubenbacher and D. Pengelley, "'Voici ce que j'ai trouvé:' Sophie Germain's grand plan to prove Fermat's last theorem," Historia Mathematica, vol. 37, no. 4, pp. 641–692, 2010.

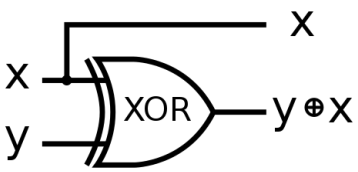
21. Séroul (2000), R. 'The Bézout Theorem' §2.4.1 in Programming for Mathematicians. Berlin: Springer-Verlag, p. 10

22. Wang, Q, 2017, '(2 x prime) + 1 = ? :The 200-year-old story of Sophie Germain and its 21st century legacy', https://sitn.hms.harvard.edu/flash/2017/2-x-prime-1-200-year-old-story-sophie-germain-21st-century-legacy/#:~:text=A%20Sophie%20Germain%20Prime%20is%20a%20prime%20number%20that%20satisfy,way%20are%20called%20Safe%20Primes. Viewed on 16th April, 2023

Annexure-I

Controlled NOT gate:





input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩

input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

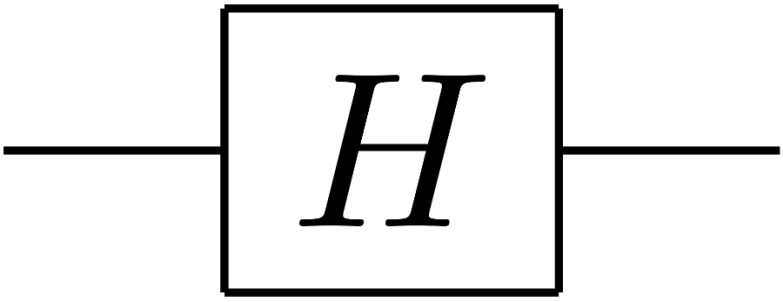
$U_{CNOT} =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Annexure-II

Hadamard gate:

The Hadamard gate acts on a single qubit. It creates an equal superposition state if given a computational basis state. The two states are sometimes written as |+> and |->.



H =

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Annexure-III

Oracle Circuit U_f for Deutsch-Jozsa (DJ) algorithm:

Deutsch's algorithm Statement- Given a function f:0,1⟶0,1 and without knowing anything more than that, determine whether f is a constant or a balanced function with the minimum number of function evaluations.

The function is implemented as a quantum oracle in DJ Algorithm. The oracle maps the state |x⟩|y⟩ to |x⟩| x ® y ⟩ where ® denotes addition modulo 2.

CONSTANT ORACLE

When the oracle is *constant*, it has no effect on the input qubits, and the quantum states before and after querying the oracle are the same. Since the H-gate is its own inverse, we obtain the initial quantum state of in the first register.

$$H^{\otimes n}\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix} = \frac{1}{\sqrt{2^n}}\begin{bmatrix}1\\1\\1\\\vdots\\1\end{bmatrix} \xrightarrow{\text{after } U_f} H^{\otimes n}\frac{1}{\sqrt{2^n}}\begin{bmatrix}1\\1\\1\\\vdots\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\\vdots\\0\end{bmatrix}$$

BALANCED ORACLE

Before applying in oracle, input register is an equal superposition of all the states in the computational basis. When the oracle is *balanced*, phase kickback adds a negative phase to exactly half these states.

The quantum state after querying the oracle is orthogonal to the quantum state before querying the oracle. Thus when applying the H-gates after oracle evaluation, we must end up with a quantum state that is orthogonal to initial state. This means we should never measure the all-zero state.

$$U_f\frac{1}{\sqrt{2^n}}\begin{bmatrix}1\\1\\1\\\vdots\\1\end{bmatrix} = \frac{1}{\sqrt{2^n}}\begin{bmatrix}-1\\1\\-1\\\vdots\\1\end{bmatrix}$$

Annexure-IV

Qubit initialization:

A quantum state is undetermined with respect to its phase. A quantum register of length n is postulated where all the superpositions of the 2^n basis states exist with the equal amplitudes of $1/\sqrt{N}$ $i.e. each\ state\ is\ equally\ probable$. Q-bit initialization is done by first resetting the qubits to $|0\rangle$. Then we apply Hadamard transformation on it.