Deutsch-Jozsa Algorithm

→ It is a deterministic quantum algorithm and is exponentially faster than any possible deterministic classical algorithm.

→ We are given a black-box function 'f', which takes as input a string of bits (x), and returns either 0 or 1, that is

$$f(\{x_0, x_1, \ldots\}) \longrightarrow 0 \text{ or } 1$$

where, x_n is 0 or 1.

→ The property of the given Boolean function is that it is guaranteed to either be balanced or constant.

→ A constant function returns all 0's or all 1's for any input, while a balanced function returns 0's for exactly half and 1's for other half.

→ The task is to determine whether the given function is balanced or constant.

→ The Deutsch-Jozsa problem is an n-bit extension of the single bit deutsch problem.

The classical Solution :- In the best case consider the outputs as $f(0,0,0\ldots)$ and $f(1,0,0,0\ldots)$ as 0 and 1 respectively. Here, we can say that f is balanced by looking at only 2 inputs because the output is different.

The worst case follows as to declare f as constant we need to look at least $\frac{2^{n-1}}{2}+1$ i.e $2^{n-1}+1$ inputs out of 2^n inputs.

Therefore, in terms of probability theory we can write as,

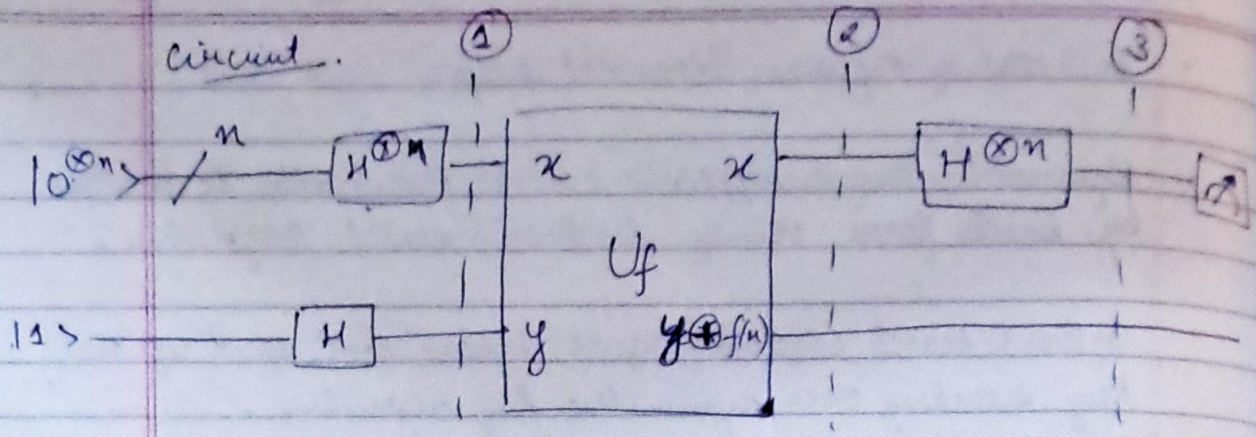
$$P_{constant} = 1 - P_{balanced}$$

$$= 1 - \frac{1}{2^{K-1}} \quad \text{where } 1 < K \leq 2^n$$

Quantum Solution :- We can solve this problem after only one call to the function $f(x)$, provided we have the function f implemented as a quantum oracle which maps $|x\rangle|y\rangle$ as $|x\rangle|y \oplus f(x)\rangle$ where, $\oplus$ is addition modulo 2.

- Circuit for the Deutsch-Jorsa algorithm.

Circuit.



The steps of algorithm are as follows:-

① Prepare two quantum registers. The first is an nn-qubit register initialized to $|0\rangle$ and the second is a one qubit register initialized to $|1\rangle$

② Apply a Hadamard gate to each qubit.

③ Apply the quantum oracle $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$

④ Apply Hadamard gate to each qubit in the first register.

⑤ Measure the first register.

Working example: Consider an example for a two bit balanced function:

Consider a two bit function $f(x_0, x_1) = x_0 \oplus x_1$ (XOR)

such that $f(0,0) = 0$

$f(0,1) = 1$

$f(1,0) = 1$

$f(1,1) = 0$

and the corresponding phase oracle of this two-bit oracle is

$$U_f |x_1, x_0\rangle = (-1)^{f(x_1, x_0)} |x\rangle$$

$$|\psi_0\rangle = |00\rangle_{01} \otimes |1\rangle_2$$

where subscripts 0, 1, 2 are used to index the qubits.

(1) Apply Hadamard gate on all qubits

$$|\psi_1\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)_{01} \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)_2$$

(2) The Oracle function is implemented as $Q_f = CX_{02}\,CX_{12}$

$$|\psi_2| = \frac{1}{2\sqrt{2}}\left[|00\rangle_{01} \otimes \left(|0\oplus 0\oplus 0\rangle - |1\oplus 0\oplus 0\rangle\right)_2\right]$$
$$+ \frac{1}{2\sqrt{2}}\left[|01\rangle_{01} \otimes \left(|0\oplus 0\oplus 1\rangle - |1\oplus 0\oplus 1\rangle\right)_2\right]$$
$$+ \frac{1}{2\sqrt{2}}\left[|10\rangle_{01} \otimes \left(|0\oplus 1\oplus 0\rangle - |1\oplus 1\oplus 0\rangle\right)_2\right]$$
$$+ \frac{1}{2\sqrt{2}}\left[|11\rangle_{01} \otimes \left(|0\oplus 1\oplus 1\rangle - |1\oplus 1\oplus 1\rangle\right)_2\right]$$

$$\Rightarrow |\psi_2| = \frac{1}{2\sqrt{2}}\left[|00\rangle_{01} \otimes \left(|0\rangle - |1\rangle\right)_2 - |01\rangle_{01} \otimes \left(|0\rangle - |1\rangle\right)_2\right.$$
$$\left. - |10\rangle_{01} \otimes \left(|0\rangle - |1\rangle\right)_2 + |11\rangle_{01} \otimes \left(|0\rangle - |1\rangle\right)_2\right]$$

$$\Rightarrow |\psi_2| = \frac{1}{2}\left[|00\rangle - |01\rangle - |10\rangle + |11\rangle\right)_{01} \otimes$$
$$\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)_2$$

$$\Rightarrow |\psi_2| = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)_0 \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)_1 \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)_2$$

Apply Hadamard on first register

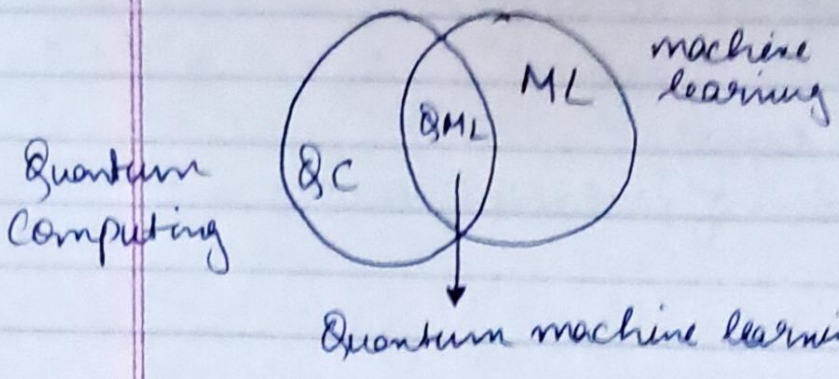
$$|\psi_3| = |1\rangle_0 \otimes |1\rangle_1 \otimes \left(|0\rangle - |1\rangle\right)_2$$

Page No.
Date / /

Measuring first 2 qubits will give the non-zero 11, indicating a balanced function.

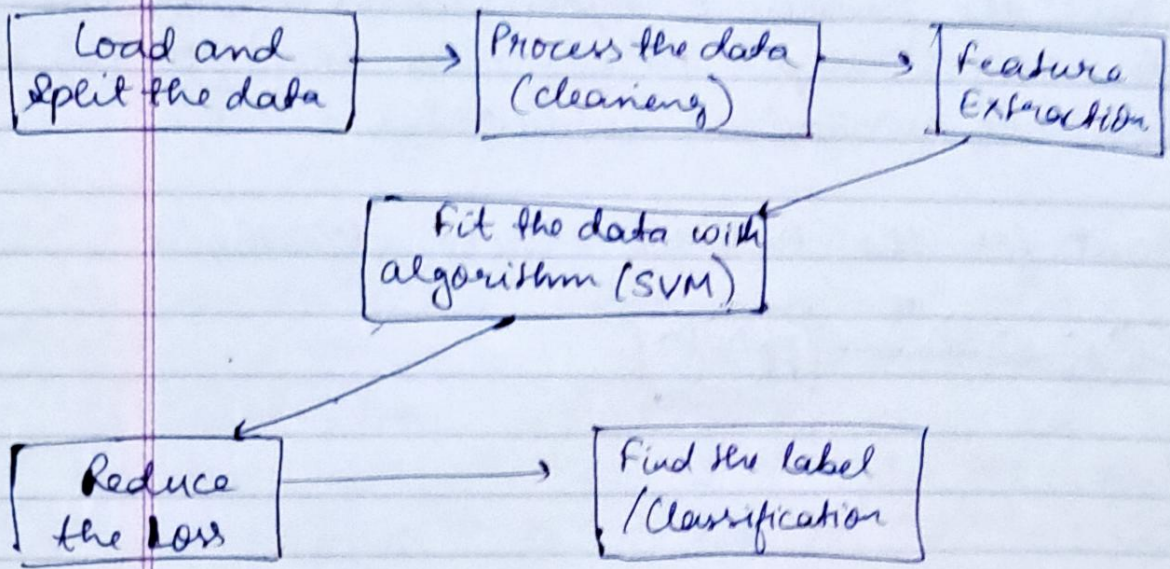
QML (Quantum Machine Learning)

QML is a theoretical field that's just starting to develop. It lies at the intersection of quantum computing and machine learning.



The main goal of QML is to speed things up by applying quantum computing to machine learning.

Normal Machine learning model:-



Four different approaches to combine the discipline of quantum computing and machine learning.

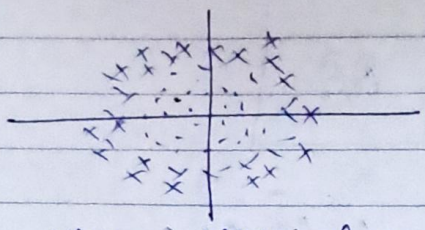
Types of Data ↑ \ Types of algo →	classical	quantum
classical	CC	CQ
quantum	QC	QQ

(1) BLAS – Basic linear algebra subroutine provides standard building block for performing basic vector and matrix operations like
matrix multiplications
Fourier transforms
matrix Inversions
etc...

Since, BLAS is central to machine learning. Quantum computers can speed up the BLAS subroutines exponentially called as qBLAS.

(2) The ML methods convert their input data in a different space to make it easier to work with. eg:- SVM (support vector machines), classify data using a linear hyperplane. A linear hyperplane works well when the data is already linearly separable in the original space however this is unlikely to be true for many data sets. Therefore, we can use a new space where it is linear by way of feature map.

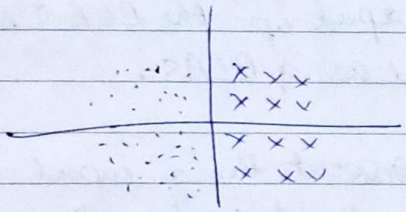
ML :- let X be a set of input data.
A feature map ϕ is a function that acts as
$$\phi : X \longrightarrow F$$
where, F is feature space.
for $x \in X$, $x \longrightarrow \phi(x)$.



data in original space

QML :- let X be a set of input data. A quantum feature map $\phi : X \to F$ is a feature map where the vector space F is a hilbert space and the feature vectors are quantum space

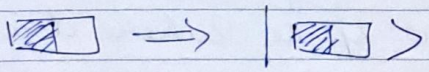
$$x \longrightarrow |\phi(x)\rangle$$



The number of qubits should be minimal.

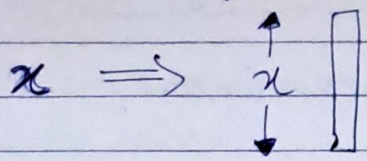
(3) Data encoding for QML

① Basic encoding



Used to encode real numbers into binary numbers and then transform into quantum state. Mainly useful for arithmetic operations.

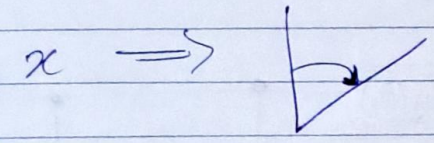
(2) Amplitude encoding



$N = \log_2 N$ qubit

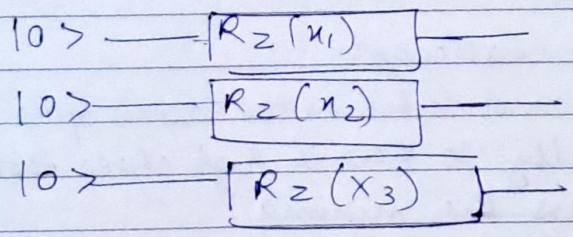
The data is encoded into the amplitudes of a quantum state. The large hilbert space is used in it. This encoding requires $\log_2(n)$ qubits to represent an n-dimensional data point.

(3) Angle Encoding



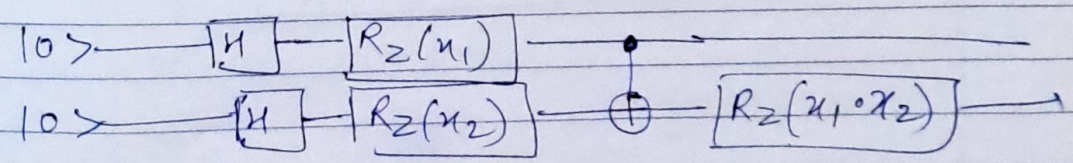
The n classical features are encoded into the rotation angle of the n qubit. Requires one rotation on each qubit. This encoding is directly useful for processing data in quantum neural networks.

$N = N$ qubit



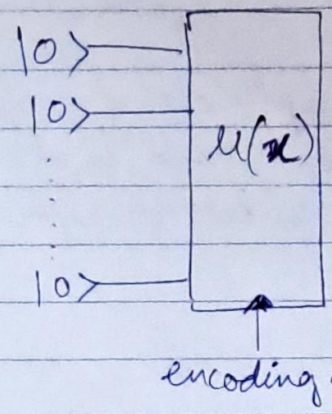
(4) Higher Order Encoding

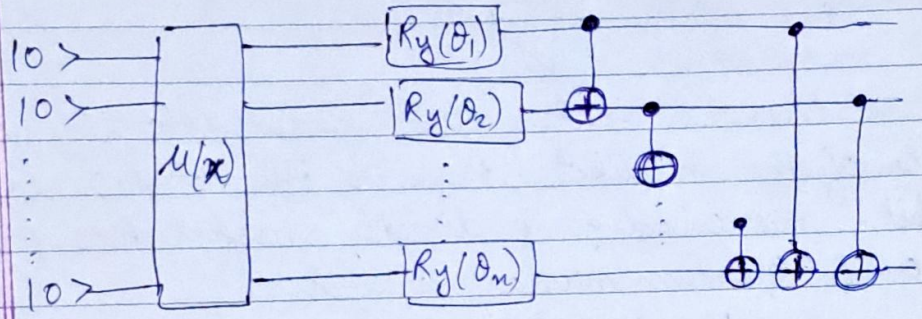
$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



④ **Variational Quantum algorithm.**

Variational Quantum Circuit

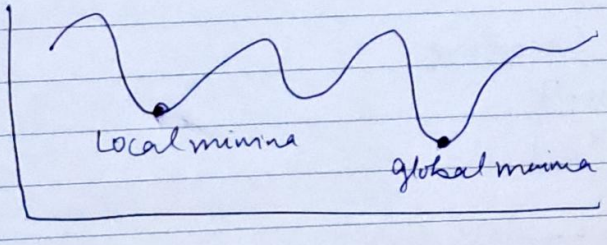




⑤ Optimization

Optimization Challenge :-

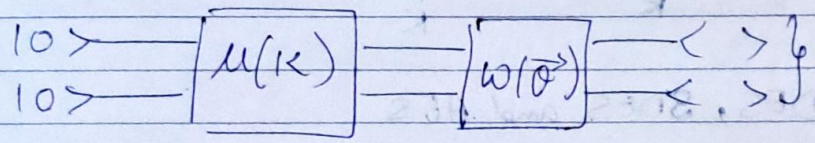
number of candidates in the search space grows exponentially so there is high chance that we can miss the minima



Quantum Approximation Optimization Algorithm (QAOA)

- A quantum computer and a classical computer work together to get the solution.
- A quantum circuit runs on the quantum computer, where the output provides a guess at ~~the~~ the solution. The classical computer then evaluates the quality of that solution, and uses an optimization algorithm to modify the quantum circuit to try to provide a better solution.

⑥ Labeling / Classification

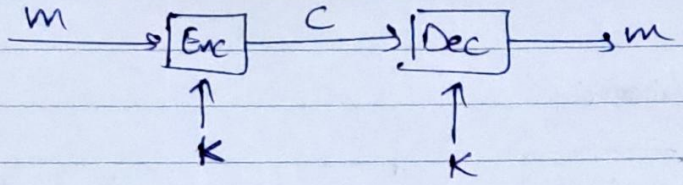


Label Extraction

Types of Cryptography

① Symmetric Key Cryptography

An encryption system in which the sender and receiver of a message share the same key for both encryption and decryption.
Also called Symmetric key ~~encryp~~ cryptography, private key cryptography.



eg:: DES, 3DES and AES

DES Algorithm:-

→ Data Encryption Standard
→ DES uses 56-bit length key, the key length is very should from a perspective of a brute force attack and can easily be breaked.
→ 3DES uses 128-bit key length, the same algorithm is applied three times to produce strong encryption
→ DES is restricted to use a block size of only 64 bits.
→ DES uses a feistel network, which divides the block into two halves before going through the encryption steps.

→ With 64-bit block size, the amount of data that can be transferred with a single encryption key is just 32 GB.

AES (Advanced Encryption Standard)

and stronger

→ It is six times faster than 3DES.

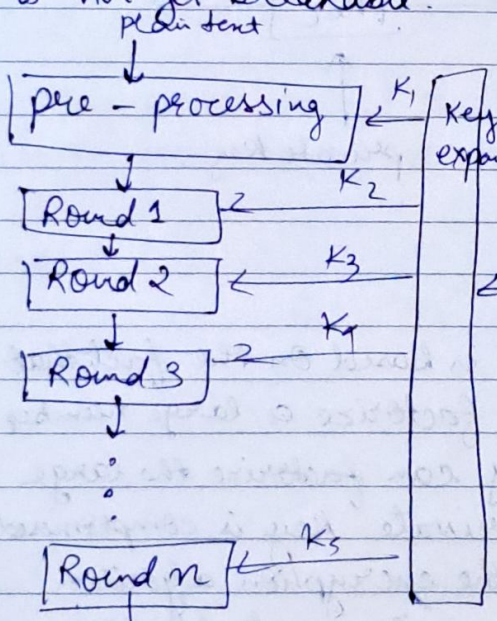
→ Since, Key size of DES was very small so it become vulnerable to bruteforce attack then 3DES was introduced with increased key length but it was found to be slow.

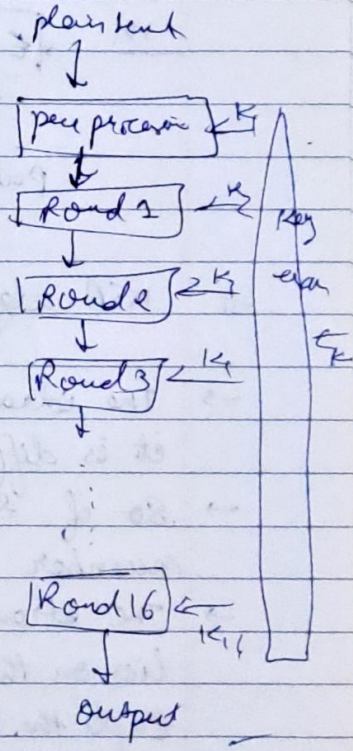
→ AES uses key length of 128 bit, 192 bit, 256 bit.

→ AES uses block size of 128 bits and transfers around 256 billion GB data.

→ AES uses permutation - substitution method.

→ AES is not yet breakable.





n	Key size
10	128
12	192
14	256

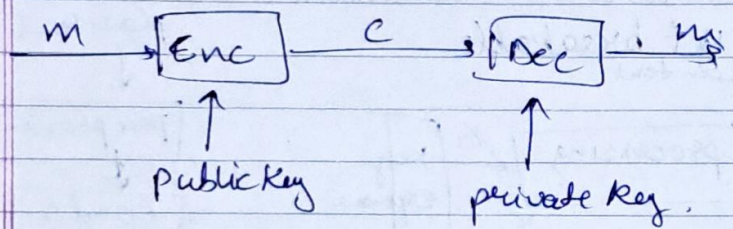
n	Key size
16	56

(2) Asymmetric Key Cryptography.

An encryption system in which the sender uses ~~private~~ public key p to encrypt the data and a receiver uses its private key corresponded to that public key to decrypt the data.

The Public key is publically available and the private key is not shared.

Hence, it is more secure than symmetric key encryption. It is also called public key cryptography.



RSA algorithm :-

→ The idea of RSA is based on the fact that it is difficult to factorize a large number.

→ So if somebody can factorize the large number, the private key is compromised.

→ The strength of the encryption algorithm lies on the key size, if we double or triple the key size, the strength increases exponentially

→ RSA keys are 1024 bit or 2048 bit long

~~Public Key parameters~~

Large composite number n with two prime factors

$$n = pq$$

$$\phi(n) = (p-1)(q-1)$$

Choose e such that $1 < e < \phi(n)$

and $\gcd(\phi(n), e) = 1$

This states that inverse of e exists and according to ~~euclid~~ extended euclidean theorem the multiplicative inverse d will be

$$ed \equiv 1 \bmod \phi(n)$$

or

$$d \equiv e^{-1} \bmod \phi(n)$$

public key parameter $\{e, n\}$

private key parameter $\{d, n\}$

Encryption ( ) :-

$$c = m^e \pmod{\oplus n}$$

decryption ( ) :-

$$m = c^d \pmod{n}$$

ECC (Elliptic curve Cryptography)

→ It is a asymmetric key encryption scheme.

→ Used for digital signatures, pseudo random generators and key agreement.

→ It is based on hardness of elliptic curve problem

→ uses small keys of 163 bits which is very small than 1024 bit key length of RSA

→ For current cryptographic purposes, an elliptic curve is a plane curve over a finite field which consists of the points satisfying the equation:

$$y^2 = x^3 + Ax + B$$

→ The coordinates are to be chosen from the fixed finite field.

→ In this with ECC we take a random number (n), and a point on the elliptic curve (G), and then multiply them together to produce P.

$$P = nG$$

→ G will be an (x, y) point on the curve that both bob and alice will agree to.

→ n will then be Bob's private key and P will be his public key.

→ If n is a 256-bit random value, it will be extremly difficult to find the value even though we know G and P.

Diffie Hellman Key Exchange Algorithm

→ This algorithm is based on the concept that there should be no need to exchange the key for an encryption scheme rather it should be in a way that both parties can compute the key at their respective end without actually transmitting it during communication.

→ It uses the concept where a prime number can be used to generate a list of numbers in a seemingly random sequence. The seemingly random sequence is what makes this algorithm secure.

Steps follows this way:-

① Alice and Bob agree on a prime number P.
② Alice and Bob agree on a primitive root g of the prime number P.
③ Alice chooses a positive whole number 'a' as her secret key and compute its public key.

$$A_{public} = (g)^a \bmod p$$

Alice then sends this Apublic to Bob.

④ Bob on the other hand chooses a positive whole number b as its secret key and compute its public key.

$$B_{public} = (g)^b \bmod p.$$

Bob then sends this B public to alice.

⑤ Alice and Bob at their own end computes the private key and do not share it.

$$A_{private} = (B_{public})^{a} \bmod p$$

$$B_{private} = (A_{public})^{b} \bmod p$$

If $A_{private} = B_{private}$ a secure communication takes place.

Known Threats of Quantum Algorithms.

After Quantum Cryptanalysis we can conclude :-

① Traditional public-key cryptosystems (RSA, ECC) are breakable by Shor's Factoring Algorithm.

- Shor's Algorithm efficiently solves integer factorizations which lead to breaking asymmetric cryptographic scheme.
- Any adversary who sniffs and records a public-key encrypted communication would be able to easily decrypt using a quantum computer. (Communication Harvesting attack)

- Increasing key size does not help.

(2) Symmetric key encryption can be potentially be broken by brute force using Grover's search algorithm.

Grover's search algorithm gives square root speed up on key searching over symmetric key algorithms.

For eg:- a 128 AES can be compromised by 264 operations on a quantum computer.

Doubling the key size can help mitigate the attack.

Post Quantum Cryptography (PQC)

PQC, also called as quantum encryption is the developmat of cryptographic systems for classical computers that are able to prevent attacks launched by quantum computers.