****

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**A Survey On**

**Post Quantum Security of Classical Cryptographic Algorithms**

**Prepared By**

**Ritik Rohit Bavdekar 2017B4A70349P**

**Eashan Jayant Chopde 2017A3PS1161P**

**Under**

**Dr Ashutosh Bhatia**

**Instructor in Charge of Cryptography**

**Computer Science Department**

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

# **Acknowledgements**

We would like to thank Dr Ashutosh Bhatia for giving us the opportunity to work in this new and exciting field. We would also like to thank the CS department for giving opportunities .We also want to express gratitude towards our parents who have been providing for internet, electricity backup and mental support in these tough lockdown times. We would also like to thank the National Institute of Standards and Technology (NIST) for keeping well documented records on all the algorithms presently in the selection process for Post quantum cryptography.

# Table of Contents

[**Acknowledgements**](#_c158c94oqpmi) **2**

[**Table of Contents**](#_nwdlso88hzyu) **3**

[**Abstract**](#_86oxopcbidej) **5**

[**Introduction**](#_cag1p4ip2zu8) **6**

[Quantum Computing](#_w5hjtfafp2rb) 6

[Quantum Key Distribution(bb84 protocol)](#_uv1f9hicbjw8) 7

[Shor’s Factoring Algorithm](#_w4et9jnd1wjm) 7

[Grover’s Algorithm](#_oo6nwj3pzqq2) 8

[Simon’s Algorithm](#_htcrp3yi3ngw) 9

[**Symmetric Cryptographic Algorithms**](#_5vsg2mixumam) **9**

[AES](#_cu1jppmbcls5) 10

[Basics](#_niy3i7j6gayx) 10

[Classical Exploitations](#_9cebihdn6c1) 11

[Quantum Security](#_og17gks3r49c) 11

[DES](#_a811ahiyg21s) 12

[Basics](#_6q1p65fmhv7s) 12

[Classical Exploitations](#_430qdi57anhf) 13

[Quantum Security](#_fy4npxbto26a) 14

[**Asymmetric Cryptographic Algorithms**](#_24ismreplu07) **15**

[RSA](#_qa91ba6ikquu) 15

[Basics](#_d25iaj40nlpx) 15

[Classical Exploitations](#_ahpwvmlx1keg) 16

[Post Quantum Security](#_tfzgmmpr1i0t) 16

[Diffie Hellman Key Exchange Protocol](#_ecspa4whnjqf) 17

[Basics](#_i93gcpsvzuue) 17

[Post Quantum Security](#_r5e71qb0fbai) 18

[Elliptic Curve Cryptography](#_8w1uaeg0hue3) 18

[Basics](#_wwxbpzm7y9gu) 18

[Post Quantum Security](#_2f7qzug0c3xk) 19

[**Post Quantum Cryptographic Algorithms**](#_q0xz20o9njw0) **21**

[Lattice Based Cryptography](#_gaerp94oi9ze) 21

[Hard Lattice Problems](#_hccj25brw5zd) 22

[GGH Encryption Algorithm](#_qkcwzosn0oo9) 22

[Learning With Errors](#_r99nkif0u08g) 22

[Hash based Digital Signature Schemes](#_vk6dlpchtzr) 23

[Code based Cryptography](#_33nibs557ddc) 23

[Non Commutative Cryptography](#_u7ja6qfagkb8) 24

[Stickel Key Exchange Protocol](#_2h0mydxtan1v) 24

[Other Families](#_rzbzk24n8fhc) 25

[**NIST - Post Quantum Cryptography**](#_w5ecyhdi8lir) **26**

[Lattice Based Algorithms –](#_42luy2j71j2g) 27

[CRYSTALS-KYBER (Finalist)](#_vgdiv7trzp9r) 27

[SABER (Finalist)](#_z1cgx92n2m6l) 28

[NTRU (Finalist)](#_dhi2wqzhgou7) 28

[CRYSTALS-DILITHIUM (Finalist)](#_u6w9kyayq2sj) 29

[Falcon(Finalist)](#_brzw1vug2djr) 29

[Code Based Algorithms:](#_a96201252pit) 30

[Classic McEliece (Finalist)](#_w3t7elm185ba) 30

[BIKE(Alternative)](#_sra0srquxmgx) 30

[HQC(Alternative)](#_bjw6qi2r0gwp) 31

[Multivariate Based Algorithms:](#_mjrqiscfpo6i) 32

[RAINBOW (Finalist)](#_e7e58sutyt72) 32

[GeMSS(alternative)](#_4qal12rkdbmn) 32

[Isogeny Based:](#_4sxkivj34lkw) 33

[SIKE](#_jpqgpcy9v6wf) 33

[Hash Based Algorithms:](#_uu4tdr8h9ttw) 33

[SPHINCS+ (Alternative)](#_uhncwg72ixr9) 33

[Others](#_lhlwyqt2fh6h) 34

[PICNIC(alternative)](#_punkqhj83gec) 34

[**Conclusions**](#_i7ienqere0ym) **35**

[**References**](#_spq1su5x5kvr) **37**

# 

# Abstract

The development of large quantum computers will have dire consequences for cryptography. The present classical cryptographic algorithms are implemented on a large scale in all the major IT industries. Most of the present day symmetric and asymmetric cryptographic algorithms are exploitable by quantum algorithms.

Most of the present day symmetric cipher schemes like the Feistel scheme and Even-Mansour scheme can be cracked using quantum computers. Grover’s search algorithm gives a O(n½) time boost for the search of the key. This is not very helpful as the key size can always be doubled making it computationally secure.

Asymmetric algorithms like RSA[1] are based on the mathematics hard problem of prime factorisation. The best solution available on classical computers is the Number Field Sieve Method, which takes exponential time. Shor’s factoring algorithm can solve the prime factorisation problem in O(logN \*k) making it a polynomial time solution.This problem can be solved using only local quantum computational resources. Unlike most classical symmetric algorithms, the adversary is independent of the platform(classical or quantum) in which the RSA algorithm is implemented. Quantum Computing models this prime factorisation problem as an optimisation problem. It is but a matter of a few years before major breakthroughs occur in Quantum Computing. Soon enough the software and hardware optimisation will converge and render 2048 bit RSA useless.

Our study aims to provide a critical view on classical cryptographic methods under the microscope of quantum attacks. It is the need of the hour to change or make new algorithms that are robust in the face of a quantum adversary. Various algorithms and their vulnerabilities shall be analysed to provide a survey of their viability and security in a post quantum world.

# Introduction

Quantum cryptography is the field of cryptography that uses quantum effects to perform cryptographic tasks.These tasks could vary from cryptographic protocols to quantum algorithm cracks. Two of the most well known applications of quantum information are in cryptography: Quantum Key Distribution and Shor’s Factoring Algorithm. Quantum information opens up a variety of new functionalities in cryptography that are not available in classical information theory. However quantum cryptography does have limitations and challenges.

## Quantum Computing

Quantum mechanics is characterized by its absurdity. It defies intuition. This is due to the concepts like superposition, entanglement and quantum uncertainty. Quantum superposition allows a particle to be at multiple places at the same time. Quantum entanglement describes correlations between particles that are not possible in the classical world. Quantum uncertainty states that if we observe one property of a particle some other property’s information is lost.

The fundamental unit of information in classical computers is a bit. This bit can take only two discrete values 0 and 1. In quantum information the unit is a qubit.[8] This is a unit vector in a two-dimensional complex vector space. A qubit is represented using the ket notation. ∣ψ> = α∣0> + β ∣1> is the notation for a qubit. Here ∣0> and ∣1> are basis vectors in the complex 2d space.α and β are complex numbers whose sum of squares adds up to 1. For representation of 2 or more qubits tensor product is used.

A collection of quantum gates to form a circuit is called a quantum algorithm. Common quantum gates include Controlled NOT Gate, Hadamard gate, negation and phase.

In quantum computing measurement is an operation that takes in a qubit and returns a classical bit( 0 or 1). If a qubit ∣ψ> = α∣0> + β ∣1> is measured the probability of 0 or 1 being returned is |α|2 and |β|2 . Also right after a measurement all information about the qubit is lost and it becomes classical. Post measurement the superposition is lost and depending on the value of the classical bit returned the qubit is transformed into ∣0> state or ∣1> state.

Quantum no-cloning property states that there is no possible way to clone the state of a quantum system. In simple words there is no circuit that can be built which takes in as input a qubit and returns a copy along with the original qubit. Quantum entanglement is a phenomenon in which 2 or more qubits behave in a way that their quantum states can only be defined collectively. They cannot be defined separately.eg (∣00> +∣11>)/√ 2 is state with 2 qubits which cannot be defined individually. This kind of state is called a Bell State.If one of the qubits is measured the other qubits quantum state is also automatically changed.

There are a variety of different physical systems that can be used as quantum devices. These include superconduction qubits , photonic qubits(polarisation of light) and others. Photonic qubits are good for long distance communication while supercomputing qubits are better for quantum interactions.

## Quantum Key Distribution(bb84 protocol)

The protocol is a secure method of sharing a common key between two systems. The algorithm makes the assumptions that there is a classic channel and quantum channel for information transfer. The classic channel is two way while the quantum channel can be one way(from only sender to receiver and no return link). The bit and basis selected for encoding and sending a qubit are both randomly selected.

The protocol considers 2 different orthogonal basis vector sets.[6](eg - 0,90 horizontal-vertical and 45,-45 diagonal polarisation of photons). Alice and Bob decide encodings 0 and 1 in both basis sets. Note- measuring information in one basis set makes us lose information in the other basis set. Alice generates a random key and encodes the bit using a random basis. These qubits are sent to Bob from the quantum channel. Bob uses random basis filters (diagonal or vertical/horizontal) to measure the bits. After that both use a classical method to announce the filters they used for each bit. If they are different filters the bit is discarded. A final key is transmitted which is only known by Alice and Bob. Since Eve can’t clone the qubits and if she measures them information may be lost Bob will know when he announces a few random bits to Alice to ensure no tampering. Also if Eve measured all the qubits using random filters she would have selected a few different filters and lost information of the qubits to measure them again.

## Shor’s Algorithm

Shor designed two different quantum algorithms which can crack the two major hard problems on which the security of present day cryptosystems is based: Prime Factorisation Problem and the Discrete Logarithm Problem. Classical solutions to these problems have exponential solutions. With the help of period finding using Quantum Fourier Transformation Shor designed these probabilistic algorithms in polynomial time. Shor’s algorithm can crack the factoring number problem in polynomial time while before this the best classical solution(General Number Field Sieve) takes exponential time[7].

Both the algorithms consist of 2 major parts. The first part is a classical part that involves converting the problem to a period-finding problem. The second part is the quantum part that takes advantage of quantum parallelism for the period-finding problem.It uses the quantum fourier transformation algorithm to find the period.

Shor’s algorithm is a probabilistic algorithm. It doesn’t always factor the number in the first attempt. With increase in the number of times the algorithm is run the probability of getting a factor increases.

## Grover’s Algorithm

It’s a quantum search algorithm that searches in an unsorted database in O(n1/2).[4] In classical computers, search over an unordered database is an O(n) problem. Grover’s algorithm is also probabilistic and can be repeated several times to ensure the search is complete. To search a Grover iteration is performed. We can repeat iterations the increase the probability of finding the element in a database.

Grover’s algorithm can be used to find the mean and median of data , for finding the inverse value of a function ,etc. These tools can be used by quantum cryptanalysts for cracking algorithms. It can also be used to search for keys in symmetric cryptosystems.

## Simon’s Algorithm

Simon’s algorithm is a quantum circuit that finds out if a function is 1 to 1 or 2 to 1. The property of the function states that if two input values map to the same output their XOR results in a constant b. If b is all 0s then it becomes a 1-1 function else it is a 2-1 function.

The classical solution requires O(2n-1+1) to find a collision. The quantum algorithm provides an exponential speedup.[5]

Simon’s algorithm inspired Shor’s algorithm. Simon’s algorithm itself has applications in cryptography provided correct query access model is available.

# Symmetric Cryptographic Algorithms

Symmetric Cryptography is a branch of cryptography wherein the encryption and decryption algorithms are both executed with a single key. Given key lengths of sufficient size and randomness, it is beyond the scope of current computational resources to perform brute force attacks on such schemes. However due to the heavy reliance on a single key, a major factor towards implementing symmetric cryptography schemes lies in the communication of the secure key from sender to receiver. Such schemes are mostly employed in Data encryption for storage applications.

## AES

### Basics

AES is a widely used symmetric cryptographic algorithm. It relies on the randomness of ciphertext generated from multiple rounds of a well-defined, standardized symmetric permutation network. The AES schemes in use currently employ 128-bit, 192-bit or 256-bit keys and generally consist of 10, 12 or 14 rounds of 4 stages each. A simple key schedule is used to generate subkeys for each round. Sufficient confusion and diffusion is achieved in the ciphertext through multiple rounds of mixing and substitution.

Each Round of the AES algorithm consists of 4 stages –

1. SubByte : In this stage each byte is substituted with a value given by the relation

b’i = bi ⊕ b(i+4)mod8 ⊕b(i+5)mod8 ⊕ b(i+6)mod8 ⊕ b(i+7)mod8 ⊕ ki

Where *ki* can be taken from the NIST standard as 01100011. These substitutions can be carried out using predetermined S-boxes consisting of 256 values.

2. ShiftRows : In this stage the elements obtained from the S-boxes are then shifted right by guidelines specified in the NIST standard, or chosen independently on a need-by-need basis.

3. MixColumns : Each column is then read as polynomial () coefficients which are then multiplied by an encoding polynomial given by –

The resultant polynomial is then reduced –

The coefficients of represent the modified columns of the state array.

4. AddRoundKey : A round key generated by a simple key schedule is added to the state array.

All calculations are done over a Galois Field (256).

The Decryption Algorithm is the same algorithm with the same number of rounds, in a reverse order.

### Classical Exploitations

Vulnerabilities pertaining to repeated sequences of characters in the message, improper key selection have been found and proven to be exploitable. These attacks however do not decrease the computation time required by enough an amount to make it viable for brute force searches. Bounded by current computational resources, AES can be considered ‘secure’.

### Quantum Security

AES has proven to be one of the most robust cryptographic schemes currently in use, proving resilient to the level of exhaustive brute force attacks that are currently computationally viable.

The recent advancements in development of Quantum Computers have however led people to reopen the question of AES’s security against Quantum Computer Brute Force Exhaustive Search attacks, Asymmetric Public-Key schemes such as RSA, ECC, etc have been known to be broken completely by Quantum Exhaustive Searches due to the immense parallel processing possible through superposition of qubits.[9]

Various algorithms that exploit the principle of superposition have been put forward such as Simons Promise, Grovers Search, and Shors Algorithm.

Grovers Search algorithm reduces the exhaustive key search from to trials, where M can be reduced to 1 by choosing functions which give single solutions while implementing AES in a quantum circuit.

One of the major drawbacks of current Quantum Exhaustive Search Algorithms is that for symmetric schemes, they require the scheme to be implemented by a quantum circuit before they can be applied to the cipher. Many implementations of AES as a quantum circuit exist, however these circuits have been created based on the requirement that they must be able to be analyzed by Grover’s Search Algorithm, limiting the applications and scope of true AES implementation in Quantum Circuits. Furthermore, as mentioned above, current algorithms are only able to reduce the trials by , where N is the length of the key, one can simply increase the key length to preserve security, for example, moving from AES-128 to AES-256 can still be considered to be unbreakable in the face of Brute Force Searches.

## DES

### Basics

The Data Encryption Standard is a widely used cryptographic algorithm that has already been proven to be vulnerable to Brute Force Searches in a pre-quantum world. However, like AES, by increasing the complexity of the key this vulnerability may be overcome, leading to the use of modified schemes such as the Triple-DES (3DES), which may be used with 2 or 3 keys, exponentially increasing the number of operations required in an exhaustive search.

DES is based on a Feistel-Round structure, where the message is split into two halves, which undergo alternate operations in each round. Each message block is of 64 bits leading to 32 bits being processed per round. The first 32 bits for each round are XOR’ed with the output of a Feistel Function which takes input as the remaining 32 bits in the round. Similarly, in the next round, the unaffected bits are swapped with the processed bits, and the round is carried out again. A DES structure consists of 16 such rounds, started off by an Initial Permutation (IP) and ended with a Final Permutation (FP) that is the inverse of the IP.

The Feistel Function is given below –

1. Expansion: The 32-bit input is converted to 48-bits by duplicating and appending the first and last bits of the next and previous 4-bit block to the beginning and end of the current 4-bit block.

2. Key Mixing: Each round corresponds to a 48-bit subkey that is obtained from the original key through a simple key schedule. The output of the Expansion stage is XOR’ed with the subkey.

3. Substitution: In this stage, the output from the key mixing stage is split into 8 6-bit blocks which are passed through non-linear S-box transformations, the S-boxes have been predetermined by an NIST standard and can be used as a lookup table. Each S-box takes 6-bits and outputs 4-bits, the resultant output is thus converted back to 32-bits so it may be passed onto the next round.

4. Permutation: Finally the S-box outputs are rearranged in a P-box in a manner which ensures that output of each S-box is distributed across 4 different S-boxes in the next round.

DES is able to provide sufficient confusion and diffusion through the S-boxes and P-boxes to ensure the output remains random enough so the only way to break the cipher would be a Brute Search. DES has already been proven to be broken and is not recommended for use.

### Classical Exploitations

As of today DES has been considered as not secure. It was first broken by the DESCHALL Project in 1997, and with the increasing rise in computational power each year, the time required has been reduced to being virtually unusable as a cipher. As of 2017, the fastest DES break (for chosen plaintext attack ) was observed with the use of Rainbow Tables, obtaining the key in under 25 seconds.

To overcome the ease of breaking DES an effort was made to increase computation trials required through increasing effective key length. This has been implemented through the 3DES Algorithm.

3DES

Triple DES provides a solution to the vulnerability of DES due to its small (56-bit key size) by ‘expanding’ the key size using 3 different keys K1, K2, K3 each of 56-bits.

Encryption of the plaintext is carried out by an EDE sequence –

Where the ciphertext is first DES encrypted using K1, Decrypted using K2 and finally Encrypted using K3.

Based on how we choose the keys, the time complexity of Brute Force can be increased, by using 3 different values of K1, K2, K3 we can achieve the maximum security 3DES has to offer, with an effective key length of 168-bits.

This attempt however was also shown to be vulnerable due to the small block size increasing the probability for collisions, this was exploited by the sweet32 attack wherein collisions can be found in trials.

### Quantum Security

Like AES, DES tries to achieve sufficient randomness through multiple rounds, however, due to the comparatively smaller key size of DES (56-bits), it can be broken easily in view of current computational resources. 3-DES increases the key length, albeit at the loss of ease of encryption, to 168-bits. Increasing the key space to 2168 raises the security to be resistant to modern brute force searches. Like AES, Grover’s Algorithm can be used for Quantum Searches on the key, this reduces the number of operations required to 284 which is not very secure. 3DES is also prone to collisions and therefore is increasingly susceptible to Simon's Algorithm. 3DES is also very slow and computationally taxing and cannot be considered for widespread use.

Models of SDES[10] are currently being implemented to understand the Feistel Round structure in Quantum Circuits. The implementation of the S-Box lookup tables requires many complicated gate arrangements and cannot be compromised due to its core function in providing non-linearity to the Algorithm. The complexity of implementing 3DES in Quantum Circuits may suggest some viability for use however due to the ease in breaking the core DES algorithm, it is avoided by most.

Symmetric Cryptographic Algorithms are currently considered as non-implementable in a Quantum Model and are a lot more resistant to Quantum Searches compared to public-key schemes.

# Asymmetric Cryptographic Algorithms

Asymmetric cryptography is the branch of cryptography where the key is divided into 2 different parts: public key and private key. The public key is given by the receiver to the sender who uses it to encrypt the data. The private key is then used by the receiver to decrypt the message. This branch of cryptography ensures the authenticity and confidentiality of the message.

## RSA

### Basics

RSA is a public key cryptosystem that is widely used for secure data transmission. It was invented by Ron Rivest, Adi Shamir and Leonard Adlema in 1977.

The algorithm is based on the prime factorisation of a number’s problem. The best known solution for prime factorisation in classical computers is of exponential complexity(General Number Field Sieve). The algorithm takes advantage of the above fact to encrypt data. This makes the algorithm computationally secure.

The algorithm takes advantage of one way functions. Such functions are easy to compute but inverting them is hard(Multiplying two prime numbers is easy but factoring the number is difficult). Prime Number Theorem of deep number theory and the Chebhysev’s bound ensures that when randomly traversing numbers of 200 digits or more at least 1/1000 of them are prime. This makes it easy for the users of RSA to find big prime numbers of 200 digits and more. To test if a number is prime Fermat’s primality theorem is used. This gives a probabilistic result of if a number is prime or not.

Prime Number Theorem:

π(n) ::= |primes ≤ n| ~ n/ln n

Fermat’s Primality Test

an−1 = 1 (Zn )

When Fermat’s primality test is applied with more and more different *a*’s(*a* is in *[1,n)*) and the output is 1 the probability of *n* being a prime number increases. If a number is not prime Fermat’s Primality Test fails for at least half the a’s in the range. Hence with every a giving 1 as output the probability of the number being prime decreases by *0.5*.

For secure communication a receiver generates a private and public key. For this the receiver searches for 2 big prime numbers at random *p,q.* The receiver then multiplies the two prime numbers to get *n*  which becomes part of the public key n=p\*q. The receiver then finds a number *L=(p-1)\* (q-1)*. Then a number *e* relatively prime to *L* is found*.*  The receiver makes *(e,n)*  pair as the public key which is used by senders to encrypt data. On the other hand receiver computes the private key as *d= e-1* .This inverse is taken in the space Z\*(p-1)\*(q-1)

The message is encrypted by the sender using *m’= me in the space (Z n).*  It is decrypted by the receiver using m*e*=(m’)d in the space (*Z n*).

### Classical Exploitations

For large RSA key size(1024 bits and above) no efficient method for solving the problem has been found. All of the RSA cracking solutions revolve around prime factorisation of big numbers.[3] As the complexity is exponential it is impossible to crack RSA with 2048 bits key size in realistic timelines with present day computational power. RSA with 1024 bits key size was cracked in over a month using huge amounts of computational power. Hence for the past 35 years cryptanalysts have tried to crack RSA and failed. RSA is used widely in a lot of different systems that require asymmetric cryptography.

### Post Quantum Security

Shor’s factoring algorithm[1] is a quantum circuit that can factorise big numbers in polynomial time. This is a huge speedup compared to classical methods that are dependent on exponential solutions for the same. With the development of quantum computers in the near future factorisation problem will become very fast to solve. This will make RSA cracking very easy. The first practical implementation of RSA for big factorisation numbers required more than a billion qubits. This went down to 20 million qubits in late 2019[2]. This implementation is capable of factoring 2048 RSA integers in as less as 8 hours using 20 million noisy qubits. The most powerful present day quantum computers have 50-100 qubits. In the next 25 years the software improvements and hardware improvements will meet to make factoring prime numbers a reality.

The algorithm uses quantum fourier transformation to take advantage of quantum parallelism. The prime number factoring problem is converted to a period finding problem. Many optimisations are done by the authors of the paper to reduce the number of qubits required for the factorisation. Ekera-Hastad’s[2] derivative of the Shor’s factorisation algorithm is used to decrease the number of multiplication operations. Many other optimisations like windowed arithmetic and oblivious carry runaways are used.

The main idea behind Shor’s Prime Factorisation algorithm is as follows

The first part of the algorithm requires setting an input and output register where input is the superposition of all possible inputs and output is set of 0s. The input and output registers are entangled.

Let N=p\*q such that p and q are prime numbers. To factorise N a random number **a**<N is selected such that gcd(**a**,N)=1. The period of f(x)=**a**xmod N is found(Quantum circuit). If the period r is an odd number the algorithm is repeated from the start. Now if the period is an even number we have f(x)=f(x+r). The factors of N are gcd(ar/2+-1 , n).

The period calculation takes help of quantum fourier transformation. Input and output registers with log2N qubits each are initialized. The input register is initialized to a superposition of all possible values from 0 to N-1. The output register is initialized with 0. *N*− 1/2∑*x*∣*x*⟩∣0⟩

A quantum circuit for the function f(x)=axmod N is made and the input register is passed through the circuit to initialize the output register as a superposition of f(x) for all x in input registers. *N*− 1/2∑*x*∣*x*⟩∣*f*(*x*)⟩

Quantum Fourier Transformation is applied to the input register resulting in *N*− 1∑*x*∑*ye*2*πixy*/*N*∣*y*⟩∣*f*(*x*)⟩

A measurement is then performed in the input register which gives y. Since the input and output registers are entangled this changed the quantum state of the output with *f*(*x*0).

y/N is converted to an irreducible fraction and the value of the denominator is used as a potential value for the period of the function.If it is the period done else multiples of the denominator are tested. If they work the period is found else the registers are initialized again and the procedure is repeated(measurement value is random and may change resulting in different output of y).

Shor’s factoring algorithm taps at the root idea behind RSA algorithm. A replacement for the same is required in the post quantum world. NIST is looking for new post quantum computer era encryption algorithms.

## Diffie Hellman Key Exchange Protocol

### Basics

Diffie Hellman Key Exchange is based on the discrete logarithm problem. The discrete logarithm problem is based on another one way function that is easy to compute but hard to invert. The discrete logarithm requires a big prime number *N*. A generator **g** is found for the group ZN that uniformly sweeps through all possible numbers in the group. The generator at some power **p** in group ZN is easy to compute while finding the power **p** for given a *N*, generator **g** and output **o** is hard. Ie In the equation **o**=**gp mod N** given **g, p and N** it is easy to compute **o.** Given **o,g and N** it is hard to find **p=dloggo** in group ZN

The discrete logarithm problem solution has exponential complexity. As the proof of security of Diffie Hellman Key Exchange is based on the assumption that discrete logarithm problem is hard cracking the Diffie Hellman Key Exchange protocol also has exponential complexity.

The Diffie Hellman Key Exchange protocol requires two shared values that are public knowledge. These include a big prime number *p* and a generator in the space *Zp* , *q.* Both the parties involved in the key exchange first generate a secret random number each *a and b*.Then they compute

*A = qa mod p*

*B = qb mod p*

*A* and *B* are shared as public keys . Then both the parties can use their private keys *a* and *b*  to generate a common key.

*X = Ba mod p*

*X = Ab mod p*

Both the parties can use the common key *X* for further communications using symmetric algorithms.

Classical Exploitations

In Diffie Hellman key exchange protocol there is no special authentication. A ‘Man in the Middle Attack’ can be used. The main algorithm of Diffie Hellman is based on the discrete log problem which is hard to solve. Hence the eavesdropper can act as the other party for each party successfully pulling a man in the middle attack. It can create a separate common key with both Alice and Bob. It can then act as Alice for Bob and Bob for Alice sending messages from one party to another while successfully decoding and modifying all of them.

If the parameters aren’t selected carefully a small subgroup confinement attack can be done. Cracking algorithms also take advantage of ideas like the most common prime numbers. As long as the infrastructure and supporting algorithms for Diffie Hellman are modified and made stronger it will remain as strong as the discrete logarithm problem.

### Post Quantum Security

Diffie Hellman key exchange protocol is based on the discrete logarithm problem. The idea behind Shor’s algorithm can be used to give an exponential speedup to solve the discrete logarithm problem.

The discrete logarithm problem can be converted to a bivariate function’s period finding problem. The period finding problem can be solved using quantum fourier transformation which takes advantage of quantum parallelism to speed up the computations.

Let  *p* and *q* be a big prime number and a generator in *Zp* group respectively. *y= qk. k* has to be found to solve the problem. A bivariate function *f(x1,x2)=qx1yx2*  is defined. The period of the function is found using the period finding algorithm. *f(x1+w1,x2+w2)=f(x1,x2).* After finding the period *k=-w1/w2 mod p.[11]*

The above algorithm is a polynomial solution to crack the Diffie Hellman Key Exchange Protocol. The protocol is based on the discrete logarithm problem. As the discrete logarithm problem will be computable with the development of quantum computers the protocol will become unsafe.

## Elliptic Curve Cryptography

### Basics

Elliptic Curve Cryptograhy is based on the elliptic curve. Elliptic curve has the property that a line can at most intersect the elliptic curve at 3 points. This idea is used for a one way function. An operation A dot A is defined . This is performed n times to give a result. Y is easy to compute however computing number of times the operation was performed is very hard. Basically A dot A dot.. p times= Ap =o . **p = dlogAo**. This is a hard mathematics problem in the group.

ECC has a max value which limits the number of points in the group. It helps in wrapping around of the lines making it similar to modular arithmetic. This max value is dependent on the key size.

ECC has gained a lot of interest in recent years as the key sizes required for security are much smaller than their RSA counter part. A 256 bit elliptic curve provides the same security as a 3072 bits RSA key. It is currently used by famous platforms like Bitcoin, Tor and Whatsapp.[12]

ECC hardness is based on the hardness of solving the discrete logarithm problem in elliptic curve groups.There are many different elliptic curves selected by NIST which are recommended for ECC systems.

Classical Exploitations

With ciphertext only attack the only technique to crack ECC is to find an algorithm to solve the discrete logarithm problem. No solution has yet been found to solve the problem in realistic computation times.

Side-channel attacks can be used to crack ECC. These involve making measurements on the physical implementation. They include simple timing attacks,simple power attacks, differential power attacks and fault analysis. The variance ,amplitude and shape of voltage peaks can be analyzed by the attacker.

Twist-security attacks are another category of attacks on ECC. These attacks are successful when several conditions are met. These can lead to victims private key. In a twist attack the attacker carefully selects a public key and shares it with the victim to compute a shared key. They then use the shared key to reverse engineer the private key of the victim. However they are very easy to avoid with the correct choices of curves and parameters.

Standard curves suggested by NIST have been recently found out to have backdoors. Cryptographers suspect that the curves were intentionally selected and promoted as they had mathematical weaknesses known to the NSA.

### Post Quantum Security

ECC is based on the discrete logarithm problem in elliptic groups. Shor published two algorithms in his famous paper. One of them was for factoring big numbers and the other was a solution to the discrete logarithm problem in any group.

All possible attacks to ECC require attacking the discrete logarithm problem. There are only exponential algorithms for the same. Shor’s algorithm is a polynomial time algorithm which takes advantage of quantum parallelism and Quantum Fourier Transformation for period finding. This can be used to solve the discrete logarithm problem and hence attack ECC in polynomial time.

As ECC requires smaller key sizes it will be comparatively easier to crack than RSA when quantum computers are a reality. The number of qubits required to crack ECC will be significantly lesser than qubits required to crack RSA making it susceptible to attacks sooner than RSA in the near future.

# Post Quantum Cryptographic Algorithms

Post Quantum cryptography is the field of cryptography where encryption algorithms are developed which are secure from a adversary with quantum computers. All the present day asymmetric algorithms(RSA , ECC , DH , DSA) are crackable using quantum computers. They are based on the Prime Factoring problem or the Discrete Logarithm Problem which are easy to solve on quantum computers using Shor’s Algorithm. Mathematicians and cryptographers used these number theory problems to base the securit of the asymmetric algorithms. Now they have to search for new mathematical problems which can’t be solved by quantum computers easily.

Symmetric Algorithms and hash functions are comparatively secure in a post quantum world. Grover’s Algorithm can speed up the attacks by squareroot complexity. However most of the algorithms can be made secure again by doubling the key size.

All the current asymmetric algorithms are based on mathematical problems for which people have searched solutions to for centuries. However the weakness they have is that quantum computers are good at parallel tasks that require one result in the end. Since the algorithms require only one result in the end a superposition of qubits can be used to parallelize all the computations and then the result can be measured. To avoid taking advantage of parallelism of quantum computers algorithms which require several results can be used. This way parallelism of quantum computers can’t be used to its full extent.

Presently most of the post quantum algorithms are part of 6 different families. All of them are a different family of mathematical problems which are difficult to solve even for quantum computers. These mathematical problems will form the base for the next generation of asymmetric algorithms. NIST started the Post Quantum Cryptography standardization process in 2016. It is searching for new digital signature schemes and Public Key Encryption schemes.

## Lattice Based Cryptography

Lattice based cryptography algorithms are based on hard mathematics lattice problems.The family of lattice problems are used in this type of cryptography.[13]A key feature of lattice based cryptography is that it involves security based on the worst case problems. Most of the other cryptosystems have security based on average case.

Lattice is a regularly spaced grid of points stretching out to infinity. A vector is a point in this lattice ie a tuple representing the coordinates of a point. The origin is a tuple of all 0s. A vector is called a far vector if it is far away from the origin and a short vector if it is close to the origin. A basis is a small set of vectors that can be used to represent the entire lattice space. These vectors can be linearly combined to represent any vector in the lattice space. For a n dimensional lattice n vectors are selected for the basis with the condition that when a line is made with the origin and a point no other point in the set lies on the line. This way a basis can be formed.

There are many bases for a lattice. A basis is short if it consists only of short vectors and a basis is long if it consists of long vectors.

### Hard Lattice Problems

Short Vector Problem: A long basis for a lattice *L* is given. Find a grid point in *L* as close to the origin as possible.

Short Basis Problem: A long basis for a lattice *L* is given. Find a short basis for the same lattice space *L.*

Closest Vector Problem: A long basis for a lattice *L* is given. Also a challenge point *P* in the lattice space is given. Find the closest point to *P* in the lattice space *L.*

Short Integer Solution Problem: **m** vectors of **n** dimensions are given such that the vectors **vi** **Zqn**. Find a coefficient vector **y Zqm** such that the linear combination of y and **vi** gives 0. Here y are small integers like y {0,1}m.

All these problems seem elementary in a small space. However in cryptography the lattice space has huge dimensions. The problems are also simple if a short basis is given but for long basis it’s a hard mathematics problem. Mathematicians as far back as the 1800s have worked on lattices. These serve as deep insights on what can and cannot be done with lattices. This gives confidence in using the lattice problems as the base for asymmetric algorithms. A major attack against them involves using lattice reduction algorithms like LLL. LLL algorithm is a polynomial algorithm used to find a relatively small basis(not smallest) for a lattice space in which a long basis has been specified.

### GGH Encryption Algorithm

A receiver’s private key is a short base and the public key is a long base. The number of dimensions in the lattice is equal to the bit size of the message. The sender uses the bad base to find a challenge point that is close to a lattice point which is the message. This message is easily decrypted by the receiver as the receiver has a short base available. For adversaries it is difficult to crack as they have only long bases. It is believed to be quantum proof but was broken by a classical computer due to certain vulnerabilities.After the collection of many encrypted messages the key can be cracked. Similar ideas with reduced vulnerabilities are used in many of the post quantum cryptography candidates.

### Learning With Errors

Learning With Errors is a subset of lattice cryptography.[13] It takes advantage of a new trapdoor function that is easy to compute but hard to invert. AX = B. If A and B are given computing X is easy using Gaussian Elimination. Now if random errors e are added to AX and the only information available is B and A finding X becomes a hard mathematics problem. AX+e = B.

This is used as the base for cryptographic algorithms. Rings Learning with Error and Module Learning with Errors are modifications of LWE that do not require big key sizes like LWE.

The equation AX=B can have more equations than variables. However equations are written so that the system is solvable. The private key is the value of all the variables. The public key is the matrices A and B. Now if a sender wants to encrypt a message the sender randomly selects a subset of the equations and adds them. In this new equation the sender adds a big error to encode 1 and a small error to encode 0. The receiver can use the private key(value of variables) to check if error added is big or small to decode to 1 or 0 accordingly. This problem is a hard mathematics problem making it computationally secure.

27 of the 69 algorithms submitted to NIST are lattice based algorithms. Google implemented Learning With Errors to its google chrome browser recently.

## Hash based Digital Signature Schemes

Hash based digital signatures schemes are an alternative to present day digital signature schemes which use asymmetric algorithms like RSA.[13] They depend on 2 properties of hash function’s collision resistance and preimage resistance.

Preimage resistance of a hash function H implies that given an output y of the hash function it is difficult to find any input x such that y=H(x). Weak collision resistance of a hash function H implies that given arbitrary message m1 , it is difficult to find another message m2 such that H(m1)=H(m2). Strong collision resistance of a hash function H implies finding message m1 and m2 such that H(m1)=H(m2). Note: Strong collision resistance is easier to exploit because of the birthday paradox. Unlike weak collision resistance and preimage resistance which require a O(2n-1)(where n is the number of bits of output of hash function) search, strong collision resistance requires O(2n/2) search as a consequence of the birthday paradox.

Finding collisions and preimages is a difficult problem if the underlying hash function is good. Finding quantum algorithms to perform these tasks will be hard if not impossible. Hence these hash based digital signature schemes can be used for authentication in the post quantum world. However they suffer from a serious disadvantage that each digital signature can be used only once.

In these schemes Alice generates two random numbers and hashes them using a good hash function and sends them to Bob as her digital signature. The 2 random numbers correspond to 0 and 1 bit. The random numbers are Alice’s private keys and the hash values are her public key. After every signature Alice has to publish the random number corresponding to the bit. Hence it can be used only once for signature.

2 of the 69 schemes in the NIST competition are hash based cryptosystems.

## Code based Cryptography

Code based cryptography is based on error correcting codes.[13] Computer scientists have been working on these for over 40 years. Error correction codes are codes used widely in communications to correct transmission errors. To send a message the text is sent into an error correction code. Then to the output a few errors are randomly introduced and sent.

An example of a well known code based cryptosystem is McEliece algorithm. It takes the help of linear error correction codes(matrix multiplication). The receiver has a good error correction code as the private key. This is multiplied by 2 blinding matrices to produce a bad error correction code , the public key. The public key is shared with everyone. A sender sends the plaintext through the bad error correction code. Then according to the overhead the sender adds errors. This is the final ciphertext that is transmitted. The receiver uses her good error correction code to decrypt the ciphertext.

McEliece algorithm has been around for more than 40 years and nobody has found a weakness in it. The major reason why it is not practically implemented is the size of the public key. It is much larger than its asymmetric counterparts like RSA.

In the NIST competition 21 of the 69 algorithms are code based cryptosystems.

## Non Commutative Cryptography

Non Commutative cryptography takes help of non-commutative groups( A+B != B+A). A simple example is a rubik’s cube where a move sequence is a set of moves applied on the rubik’s cube. Addition is the concatenation of move sequences.[13] Addition of move sequences is non commutative. Move elimination is a move that does the opposite move. This way the previous move is eliminated and the move sequence can be written without the two moves. Move sequence negation of a move sequence basically consists of all the opposite moves in reverse order. This way an entire move sequence is reversed. The negation sequence of a move A is represented as -A.

Conjugacy Problem: 2 move sequences A and B are given. Find X such that

*X+A-X=B*

Since it is non commutative it is very hard to solve. The one way function is easy to set up but difficult to reverse.

### Stickel Key Exchange Protocol

As Diffie-Hellman key exchange protocol is not quantum proof an alternative called Stickel Key Exchange is used. This is quantum proof and based on non-commutative cryptography.[14]

In Stickel key exchange 2 sequences, *A and B* are defined. Also both Alice and Bob have 2 natural numbers each which serve as their private key. Let the numbers be *n* and *m* for Alice and *r* and *s* for Bob. Alice generates a public key PKa=mA+ nB and sends it to Bob. Bob generates a public key PKb=rA+sB and sends it to Alice.Then Alice computes

*Ka = mA + PKb+ nB = (m+r)A + (s+n)B*

and Bob computes

*Kb= rA + PKa+ sB = (r+m)A + (n+s)B*

The 2 keys are the same and cannot be computed by an adversary due to non commutativity of the group. For an eavesdropper the conjugacy problem has to be solved

In the NIST competition only 1 algorithm out of 69 is a non-commutative cryptosystem. This was broken hence no non-commutative cryptosystem will be standardised in the competition.

## Multivariate Cryptography

Multivariate cryptography is based on the hard mathematics problem of solving a system of multivariate polynomials.Multivariate cryptosystems are all based on the multivariate quadratic map. The quadratic map takes a sequence x = (x1, ..., xn) ∈ Fnq and returns an output y = (p1(x), ..., pm(x)) ∈ Fmq where pi(x) are multivariate quadratic polynomials for i = 1, .., m and the coefficients of the polynomials ∈ Fq . The map is called a multivariate quadratic map P with m components and n variables.

MQ Problem: Given P: Fnq → Fmq a multivariate quadratic map and a target t∈ Fmq find a value s such that P(s)=t. Here s is not unique as the map P is not an injective map.

This problem is considered a hard problem even for quantum computers. There are methods like the Grobner basis that help solve the problem. Recently many Grobner basis-like algorithms such as F4/F5 and XL are used for solving the MQ Problem.

Mainly digital signature schemes are designed on top of the MQ problem. The most widely known digital signature algorithm in multivariate cryptography is the Oil and Vinegar Scheme. Rainbow is a digital signature scheme based on the Unbalanced Oil and Vinegar scheme and a finalist in the NIST competition.

### Oil and Vinegar Schemes

The digital signature scheme is based on the hard mathematics MQ problem.It has a combination of Oil and Vinegar variables in the polynomials. A random quadratic map P: Fnq → Fmq  is selected. The number of variables in this map are **n**(variables) and number of polynomials are **m**(components)**.** The number of variables are greater than the number of polynomials. There are **m** oil variables and **n-m** vinegar variables.

The idea behind the scheme is that all the polynomials have quadratic terms which combined vinegar-vinegar variables and oil-vinegar variables. There are no oil-oil quadratic terms. These polynomials are dispersed using an invertible linear map to avoid differentiation between oil and vinegar variables.

If Alice wants to digitally sign a document she first makes her complete quadratic map R=P ◦ T(combination of linear maps and quadratic map) public and keeps P and T as her private key. To digitally sign a document **d** she computes hash of d using any secure hash function **H**

**y=H(d).** Then she computes **s’=T-1(y)** as T is invertible linear map**.** Then shefinds **s** such that **s=P-1(s’)** and shares it. This is her signature. Anyone who receives the document can use her public key **R**  to compute **y =R(s)** and match it with the hash of the document.

For Alice computation of **s=P-1(s’)** is easy as she knows the vinegar and oil variables. She randomly selects values for vinegar variables and inserts them. As only the vinegar variables have quadratic terms she is left with a linear system of oil variables. As there are m oil variables and m equations it is easily solved using Gaussian elimination. However due to linear map it becomes difficult for the adversary to differentiate between oil and vinegar variables. He is stuck with an NP hard MQ problem to solve.

Oil and Vinegar scheme has equal number of oil and vinegar variables. It was easily cracked by Kipnis and Shamir. A new scheme called Unbalanced Oil and Vinegar scheme was proposed where the number of vinegar variables are not equal to the number of oil variables. After a lot of deliberation correct parameter values for n and m have been decided which are less vulnerable to attacks.

The Rainbow digital signature scheme is a scheme which has multiple UOV layers built into it. This was done in order to make it less vulnerable to the MQ Problem attack. However these complications gave rise to a new attack called the MinRank Problem attack.

MinRank Problem: **k** Matrices **Mi {i=1,2,..,k}** are given with **n** rows and **m** columns and a target rank **r** is given. The problem requires finding coefficient vector **y** such that the linear combination of the matrices with coefficient vector **y** give rank atmost **r**

The increase in layers increase its vulnerability against the MinRank Problem. An optimized size 2 layers is used in the NIST Rainbow Scheme.

## Isogeny-based Cryptosystem

The isogeny-based cryptosystems are based on elliptic curves. Curves are defined by the solution of a polynomial equation in 2 variables. Elliptic curves are of the form y2 = x3 + ax + b, where 4a3 + 27b2≠0. They have a property that when any two points on the elliptic curve are joined a line is formed that intersects the curve at a third point.This third point is reflected over the X axis to find a new point that is called addition in the elliptic curve. The condition 4a3 + 27b2≠0 ensures no singular points are there. To double a point a tangent is drawn on the elliptic curve with that point. The tangent intersects at another point whose reflection over X axis is taken. This gives its double. Also scalar multiplication with **n** of a point **P** involves adding P n times.ie [n].P=P+P+..+P(n times).

Elliptic Curve Diffie-Hellman(ECDH) was the first elliptic curve key exchange algorithm introduced. It involved standardizing an elliptic curve in a field over a prime number **p** and a point **P** with special properties for communication. Then both Alice and Bob would select random numbers na and nb which functioned as their private key. They calculated Pa=[na].P and Pb=[nb].P and shared it with the other party. They then multiplied their private key to the shared point to arrive at a shared secret ie Alice calculated [na].Pb= [nanb].P and Bob calculated [nb].Pa= [nanb].P a shared secret.

The security of this algorithm is based on the discrete logarithm problem over elliptic curves. This was theoretically cracked by Shor’s algorithm and hence is no longer safe in a world with quantum computers.After this people further studied elliptic curves and came up with isogeny based systems.

### Supersingular Isogeny-Based Diffie Hellman(SIDH)

An isogeny is a homomorphic rational map between two elliptic curves φ : E0 → E1. In SIDH Alice and Bob generate to private isogenies as their private key. They apply their isogenies to a common elliptic curve **E** and then share their results to the other party. The other party uses the new elliptic curve and finds another elliptic curve with the help of their private isogenies. Unlike ECDH , the final curves are not necessarily the same curve, however they are structurally identical ie the curves are isomorphic. For elliptic curves a number called the j-invariant can be calculated. This is the same for isomorphic curves. Hence the j-invariant is calculated individually by Alice and Bob and serves as the shared secret. A n-torsion group of an elliptic curve E[n] is a set of all points P in E that satisfy [n].P=0.(where 0 is the additive identity of E) .

The SIDH was introduced in 2011. It involves fixing a supersingular elliptic curve E that is defined on the field Fq where q=p2 and p=2a.3b - 1. Alice and Bob share the supersingular elliptic curve E along with a basis Pa ,Qa for E[2a] and Pb ,Qb for E[3b](A basis is a set of points whose linear combinations can generate the entire set of points in the group).

Alice and Bob select random numbers ra and rb between 0 and 2a-1 and 0 and 3b-1 respectively. Alice calculates her isogney φA using the kernel generated by RA = PA + [rA].QA. Alice then shares EA=φA(E),φA(Pb) and φA(Qb) as her public key. This helps Bob compute his second isogeny ψB . Bob calculates ψB using the kernel generated by φA(PB) + [rB] .φA(QB). He then finds the j-invariant of ψB(EA). Alice uses a similar procedure to compute her j-invariant which the becomes the shared key.

The security of the scheme is based on the le-isogeny hard problem. The le isogeny problem is given two elliptic curves E1 and E2 and an isogeny φ exists between the two with kernel size le, find the kernel of φ. For SIDH it involves the 2a-isogeny and 3b-isogeny hard problems. Mathematicians have worked for over 20 years but have not found any significant attacks against it for a big power e and any natural number l. A key encapsulation mechanism SIKE(Supersingular Isogeny-Based Key Encapsulation) is based on the SIDH and is part of the 3rd round alternate candidates for Key Encapsulation Mechanisms in the NIST PQC Standardization process.

# 

# NIST - Post Quantum Cryptography

As highlighted in the previous sections, current public key algorithms and digital signatures prove to be not secure in the face of an appropriately powerful quantum computer. The latest advancements in Quantum Computing have led to the construction of a 53-qubit Quantum Computer by Google, although this is nowhere near powerful enough to expose public key cryptography, it has changed the question of ‘if’ they can be broken to ‘when’ they can be broken. Google and IBM are in a race to build noise robust high qubits quantum computers.

As a result, many algorithms are being developed that move away from the prime factorization problem, or problems vulnerable to parallel computing. These algorithms are mainly based on Lattices and Error Correcting Codes. Most algorithms are tending towards using hard-to-solve lattice problems due to their security in both worst case as well as average case scenarios.

In light of the growing need for Post-Quantum Security, NIST gave out a worldwide call for submissions of problems that could be used to substitute current Public Key Encryption Schemes and Digital Signature Schemes.

69 algorithms were submitted for the NIST- Post Quantum cryptography competition. The best ones will be selected by 2023. These algorithms will be standardized and will be used for many years to come. This 69 consists of 20 digital signature schemes and 49 public key encryption schemes. 26 algorithms were selected after round 1.

The algorithms in NIST round 1 were evaluated on 3 major aspects : security,cost and performance and algorithm and implementation characteristics.[15]

Security was the most important aspect.The algorithms were encouraged to be semantically secure against adaptive chosen ciphertext attack. NIST also considered algorithms which are semantically secure against chosen plaintext attack. Also digital signature schemes were required to generate unforgeable signatures with respect to adaptive chosen message attack.NIST defined 5 separate security categories and included a list of other desirable security properties like side channel resistance and resistance to multi key attack ,etc.

Cost and performance was the second most important aspect.This cost includes computational efficiency(speed of algorithm) and memory requirements(code size and RAM requirements).

Algorithm and implementation characteristics include features like simple and elegant designs, flexibility (runs on different platforms and parallelism).

After round 2 NIST has selected 15 algorithms. These algorithms can be listed out based on the type of Hard Problem used to ensure the security of such algorithms.

## Lattice Based Algorithms –

### CRYSTALS-KYBER (Finalist)

CRYSTALS-KYBER is a finalist in the NIST PQC standardization process. It belongs to a family of primitives from the Cryptographic Suite for Algebraic Lattices (CRYSTAL). The same suite has been used for generating the DILITHIUM Digital Signature Algorithm. Kyber is a key encapsulation mechanism (KEM) which conforms to IND-CCA2-security. This algorithm relies on the hardness of solving the module learning-with-errors (MLWE) problem.[16]

KYBER has been proposed as a public key algorithm which provides levels of security comparable to AES schemes. NIST uses AES-128,192,256 as standards for security of level 1,3,5 respectively. Kyber flavors such as kyber-512,768,1024 are able to provide mildly comparable levels of security (approx. within 230 of AES-256). It is one of the most competitive proposals in place due to its performance in comparison with other proposed cryptographic schemes. The change in levels of security can be implemented by simply changing the order of the block matrices used in the algorithm.

Due to the relatively faster performance of lattice systems, and smaller public key sizes, such systems make an attractive option for implementation,[17] KYBER enforces this by displaying a key size of 800 bytes to 1.5 kilobytes, compared to the 2 kilobyte key required for present day RSA.

Due to its dependence on MLWE, many algebraic attacks applicable to the LWE problem must be defended against, due to public key compression, questions were raised on its effect on the cryptanalysis of the ciphertext, however no effects have so far been found. In later versions, the compression was removed and parameters were shifted to symmetric primitives based on SHAKE256, rather than the previous SHA3-256 protocol. CRYSTALS-KYBER also shares its framework with CRYTSALS-DILITHIUM , a digital signature scheme,which makes it more attractive to implement as a complete suite.

CRYSTALS-KYBER along with other post quantum algorithms have been implemented by CloudFare in its Reusable Cryptographic Library. Amazon has introduced a hybrid KYBER mode for its AWS Key Management Service.Also IBM has used the KRYSTAL Suite, KYBER and DILITHIUM in what it claims to be the world’s first quantum computing safe tape drive.

### SABER (Finalist)

SABER is another lattice based KEM algorithm which is a finalist in the NIST PQC Standardization process.[16] It has 3 flavors namely, LightSABER, SABER and FireSABER, offering level 1,3,5 security by the NIST requirements.The security of LightSABER, SABER, and FireSABER are equivalent to AES 128, AES 192 and AES 256 respectively. It is dependent upon the Module Learning with Rounding problem(MLWR) which differs from the MLWE problem in a sense where the errors are introduced by rounding off the values. Like CRYSTALS-KYBER, the implementation of different flavors simply requires a change in the dimensions of the block matrices used in the algorithm. Unlike KYBER, SABER uses a non-NTT method of multiplication, which is unique to this algorithm out of all the NIST Round 3 submissions.

SABER uses learning with rounding which does not require sampling from an error distribution. This decreases the pseudo randomness of the algorithm and makes the implementation easier. Also its security rests on only one core element. This makes it easy to implement it with applications with different security requirements.

SABER is seen to offer similar core-SVP values as CRYSTALS-KYBER, and is characterized by the simple operations used in the algorithm, making it a lot easier to implement. SABER is constant-time by design, therefore no timing faults affect the security of this scheme. It is widely projected to be used in anonymous communication.

All integer powers are modulo 2 thus reducing the algorithm bandwidth, this in conjunction with LWR, make PKE compression provably secure. At the end of Round 3 of NIST Standardization, SABER is proposed to be improved by studying side-channel attack resistance and misuse resistance. It is one of the most promising candidates for standardization.

### NTRU (Finalist)

NTRU is a lattice based finalist cryptosystem that does not derive its security from the hardness of the Ring Learning With Error(RLWE) problem or the Module Learning with Error problem. This makes it different from the other lattice based cryptosystems. Along with Classic McEliece it is one of the oldest cryptosystems of all the submissions. Due to its age, there is extensive research and literature on attacks and problem difficulty, giving us better confidence in the algorithm. The current Round 3 submission of NTRU is a merger of Round 2 submissions NTRUEncrypt and NTRU HRSS-KEM.[16]

It lacks a formal worst-case-to-average-case reduction. It has two different cost models for security analysis. The non-local model has a metric similar to the coreSVP metric which is used by all the other lattice based schemes. It also has a local model which assigns higher security to the parameters. The non-local models offer a security of category 4 at best.

It is faster and more compact than widely used RSA. There also exist certain redundancies in the algorithm which may be removed at the cost of perfect correctness. Research into proper parameterization of NTRU has been going on since the 1990’s, therefore being one of the better documented and trusted cryptosystems in Round 3. It has the advantage of not being subject to intellectual property claims, as all or most patents regarding the structures used in the cryptosystem have expired.

### CRYSTALS-DILITHIUM (Finalist)

CRYSTALS-DILITHIUM is a digital signature algorithm that belongs to the Cryptographic Suite for Algebraic Lattices.[16] The difficulty of this algorithm relies on the Short Integer Solutions Problem and the MLWE problem.It security against chosen message attacks is based on the hardness of lattice problems over module lattices. Its design is based on the Fiat-Shamir with Aborts technique, with the exception of sampling from a uniform distribution rather than a gaussian distribution. This makes it much easier to implement computationally giving it an edge over its main competitor, FALCON. DILITHIUM has the smallest combination of public key and signature size compared to any lattice-based signature scheme that only uses uniform sampling.[18]

DILITHIUM has the lowest core-SVP performance of all the Round 3 lattice-based cryptosystems, and does not qualify for NIST Level 5 security as of now. However, DILITHIUM performs well in real-world experiments and is available for use with different parameter sets.

### Falcon(Finalist)

Fast-Fourier Lattice-based Compact Signature over NTRU is a lattice based cryptosystem. Its security is based on the hardness of Shortest Integer Solution problem over NTRU lattices.The framework is based on NTRU lattices, with a trapdoor sampler “fast fourier sampling”.It is a combination of NTRU lattices, Fast Fourier Sampling and the GPV framework( hash and sign lattice-based signature schemes)

There are security proofs of FALCON in both the Random Oracle Model and the QROM. FALCON requires a complex implementation with many floating point operations, tree data structures and random sampling from discrete Gaussian distributions.

FALCON requires the smallest bandwidth amongst all the digital signature schemes. FALCON is fast in signing and verifying. It is slow in key generation. FALCON can easily be replaced into all existing protocols and applications with good overall performances. It also has a constant time implementation after round 2.

NIST selected FALCON as a finalist for the digital signature schemes.[16] It plans to standardize either FALCON or DILITHIUM. More analysis on floating point operations causing errors and side-channel attacks has to be done.Also more study on its sampler has to be done.FALCON’s key generation algorithm uses less than 30KB of RAM.[19]

## Code Based Algorithms:

### Classic McEliece (Finalist)

Classic McEliece is the oldest cryptosystem present in the Round 3 of submission for the NIST PQC standard.[16] It is based on the 1979 McEliece cryptosystem which used hidden Goppa Codes. The original cryptosystem was not based on the necessity for conforming to public use computation restrictions, and was able to offer one way chosen plaintext attack security meaning that an attacker cannot efficiently find the message from a ciphertext and public key when the message is chosen randomly.[20] The current submission modifies the original cryptosystem in order to provide efficient implementation and CCA security which has somewhat compromised the OW-CPA security. The current system is a merger of the NTS-KEM and Classic McEliece submissions of Round 1.

Classic McEliece is remarkable in a sense that there has been 0% improvement in the attacks on the cryptosystem with the increase in computational resources. It utilises the tight conversion methods of PKE to IND-CCA2 secure KEM to provide security against ROM, and on proper parameterization, QROM attacks. In order to protect against attacks on hashing functions, well studied, highly unstructured hashing functions are used.

Classic McEliece is vulnerable to Information Set Decoding attacks, these can be mitigated by choosing appropriate amounts of error so as to ensure no unmodified text appears in the ciphertext. Side-channel resistance is obtained by ensuring constant time implementation, i.e. giving away no information regarding runtime. Classic McEliece is resistant to CCA attacks, as it implements hashing of errors that are introduced into the ciphertext, and assuming security of hash functions, this method is comparatively slower than other attacks.

Classic McEliece produces remarkably small ciphertexts, of around 256 bytes, enabling ease of integration into network packets for transmission of data. However one of the biggest drawbacks of this cryptosystem is the large size of the public key of around 1.5 MB.

### BIKE(Alternative)

BIKE is a code-based algorithm that is used for Key encapsulation. It’s the short form for Bit Flipping Key Encapsulation. It is based on Quasi-Cyclic Moderate Density Parity Check code. It takes inspiration from the MCEliece scheme.The submitters of the algorithm changed the scheme before final submissions to round 2. They improved the bandwidth and introduced a new decoder(Black Gray Flip).[21]

The security of BIKE is based on a hard problem of coding theory, the distinguishing problem. BIKE provided parameters only for category 1- 4 and did not provide for category 5. BIKE takes the help of assumptions on the hardness of Quasi Cyclic Syndrome Decoding and QCCF. With these assumptions it is IND-CPA secure. It also assumes the correctness of the decoder to prove it is IND-CCA secure. Its security from attacks is based on information-set decoding and complexity estimates.However there are side-channel attacks and CCA security concerns. This is due to their lack of confidence in their decoder which they changed recently.

As BIKE recently changed its implementation, commenting about its security on a variety of different fronts is not possible.It is however viewed by NIST as a promising candidate and has been kept in the alternate category for round 3. [16]Once all the security concerns are addressed and worked upon it can be considered for standardization. It will serve as a trustable standard in case a fallback is found in lattice based schemes.

### HQC(Alternative)

Hamming Quasi Cyclic is a code-based public key encryption scheme. It is based on the hardness of decisional quasi-cyclic syndrome decoding with parity problem. HQC is IND-CPA secure and the submitters also claim that it is CCA2 secure.[16]

The HQC submitters presented a new decoder which uses Reed-Muller and Reed-Solomon codes. This helped in reducing key size by a lot. Despite the reduction in key size the public key and ciphertext are 1.6-2 time and 4-5 times the size of BIKE. The bandwidth of HQC exceeds BIKE but its key generation and decapsulation mechanisms are much faster. It uses hard-coded lookup tables for faster GF arithmetic. It also has a constant time decoding algorithm. It was susceptible to side-channel chosen ciphertext attack. This was countered using the constant time decoding algorithm.

The major disadvantages of HQC include non-zero decryption failure rate, large ciphertexts compared to BIKE and larger public keys. Attacks on Hamming metrics have been studied for over 50 years and provides an assurance to security.[22] It is an efficient constant time decoding algorithm.It was selected as an alternate candidate in round 3 due to its thoroughness in security compared to other code based algorithms. It was not selected as a finalist due to its performance characteristics.

## Multivariate Based Algorithms:

### RAINBOW (Finalist)

Rainow is a digital signature algorithm based on a variant of the Unbalanced Oil and Vinegar (UOV) multivariate scheme. The algorithm has barely been changed since its introduction in 2005. It is characterised by its use of small signatures and an extremely fast signing and verification process. Due to the layered UOV structure, it is naturally resistant to sidechain attacks, while offering security against traditional UOV attacks such as the Kipnis-Shamir attack. However the increase in the complexity of the structure has led to further exploitations. Since 2008 there have been no new attacks on this scheme, and it is sufficient to say even the current attacks can be protected against by choosing appropriate parameters.

Currently Rainbow DSA is at par with NIST Level 1,3,5, and is considered to be NP hard. [16]However after the Round 3 submission 2 attacks have been proposed by Ward Buellens very recently (October 26). The first reduces the security of the Rainbow 1,3,5 schemes by 7 bits, 4 bits, and 19 bits respectively and is based on the original Kipnis-Shamir attack, this attack also exposes the UOV signature scheme. The second attack reduces the security of Rainbow 1,3,5 by 20 bits, 40 bits and 55 bits respectively. In the light of such attacks it is highly unlikely for NIST to consider current parameter models, and the developers must now work on establishing new parameters to maintain the level of security NIST requires. Rainbow DSA is also limited by its large public and private key sizes, reaching up to 1.8 MB.

### GeMSS(alternative)

A Great Multivariate Short Signature algorithm is a signature scheme based on the ‘big field’ paradigm. GeMSS claims EUF-CMA security with the help of universal unforgeability of the HFEv(Hidden Field Equations) primitive. GeMSS was inspired by another algorithm QUARTZ incorporating latest results in multivariate cryptography to improve efficiency and security.

GeMSS is based on a well-studied mathematical problem. It supports a fast verification algorithm and the smallest signatures among all the candidates. However it requires large public keys, and long signing times. It is also difficult to implement on low-end devices. The large public keys make it difficult to implement in TLS and SSH protocols.

GeMSS submitters are trying to improve the performance of the algorithm.[24] It is competing with another promising algorithm Rainbow. Rainbow can perform well on low end devices , has good security and has smaller public keys. In case a major fallback occurs in rainbow GeMSS will be the primary candidate for standardization.[16] NIST selected it as an alternative for the same reasons.

## Isogeny Based:

### SIKE

SIKE is the only algorithm based on Elliptic Curves. Although Elliptic Curve based algorithms have been shown to be easily broken by Shor’s Algorithm, SIKE overcomes this by using pseudorandom supersingular isogenous curves that provide one to one mapping of points rather than point multiplication. Due to its closeness to ECC and Diffie-Hellman protocols, if standardised it will be one of the easiest algorithms to integrate into widespread usage.

SIKE boasts of the smallest key size of all Round 3 submissions (750 bytes) even for the highest security specifications.[23] Due to the extent of research on Elliptic curves, parameters can be chosen and understood a lot more easier in terms of defense against sidechain attacks. However a major drawback of SIKE is the comparatively less amount of research into finding isogenies for Elliptic Curves, therefore we can not be as confident as compared to other algorithms with regards to its security. SIKE is also approximately an order of magnitude slower than the other algorithms.

SIKE is currently under debate as attacks have been found against simpler versions of the algorithms, leading to speculation with regards to the overall security of the proposed algorithm. [16]The developers however have shown that with the current submitted parameter sets, SIKE still remains competitive in its security levels.

## Hash Based Algorithms:

### SPHINCS+ (Alternative)

SPHINCS+ is a hash based digital signature scheme. It is an improvement over SPHINCS ensuring multi-target attack protection.[25] Its security is proved completely on the basis of the underlying hash function.

SPHINCS+ is based on hash-based signature schemes. These were studied before the breakthrough in public cryptosystem algorithms like RSA. It is the least likely algorithm in all the round-3 candidates to be cracked that can be broken crypt-analytically in the post quantum world. It is a clean design with lucid specifications.

SPHINCS+ is vulnerable to fault attacks and side-channel attacks. The submitters are working on the implementation to reduce these vulnerabilities and to improve its performance. The signature generation scheme is much slower than the other candidates and the signatures much larger in size. SPHINCS+ signatures with the shortest length are 4 times the size of Dilitihium and require 1000 times the computation power.

SPHINCS+ speed and size will require major reengineering in TLS protocols which presently use public key encryption methods. NIST considers SPHINCS+ as a fallback in case the other algorithms fail in round 3.[16] It will also consider standardizing it for systems that require very high security and are willing to compromise on speed and size.

## Others

### PICNIC(alternative)

Picnic is a digital signature algorithm that provides security against both quantum and classical attacks. It is an algorithm that is not dependent on any number theoretic result to proves its security. It is based on zero knowledge proof system and symmetric key primitives like hash functions and block ciphers.[26] The present implementation of picnic’s security is based on its hash function and the LowMC block cipher.

Picnic has large signatures and slow signing and verifying times. It has small public keys.Also its implementation is vulnerable to serious side-channel attacks. Picnic is a modular design. Variants of Picnic with AES have been explored but they require much larger key sizes.

NIST believes picnic is not a mature algorithm yet.[16] Its design is constantly evolving and a new concept. A big advantage is its flexibility. All of its building blocks can be chosen individually. Picnic shows great promise and there’s a potential to increase both performance and security. NIST selected picnic as an alternative.

# Conclusion

Quantum Cryptography is a nascent and rapidly growing field. Many corporations around the world are pouring in resources to further the knowledge and practices we have with regards to Post Quantum Security. Due to the varied interests and studies spread out across the globe, a need to understand the current emphasis on Quantum Security and the current advancements in this field have been presented as a survey.

Symmetric key algorithms are both classical and quantum resistant (AES-256 has been used to characterise the highest level of security for all new algorithms), but they are difficult to implement in quantum circuits especially considering that quantum machinery has been developed only for a very small message size (approx 20 bits). Further advancements in Quantum Mechanics based technology might lead to an expansion of these capabilities, resulting in better and more efficient ways to implement symmetric cryptosystems such as AES.For symmetric cryptosystems, the quantum ways to crack the algorithm require a quantum oracle. As long as symmetric cryptography is not implemented with quantum oracles they are safe against quantum attacks. All our present day classical data is safe. However, the implications of quantum computing on Public Key cryptosystem are much more serious. There is no requirement for a quantum implementation of the algorithms to crack it. An adversary with local quantum resources can exploit and crack the encryption algorithms. This makes all the asymmetric enciphered data unsafe and susceptible to attacks when efficient quantum computers are built.

As of now, Quantum algorithms already exist for all major Public Key Cryptosystems and it is only a matter of time before they are broken completely. Researchers have been trying to find ways to either increase the hardness of the problems that are currently being used (RSA, ECC) or come up with new problems that are sufficiently difficult enough even for a Quantum Computer. Of these main approaches, the latter has been more extensively utilised, with more research being encouraged into the fields of Lattice Problems, Error Correcting Codes, Non commutative cryptosystems and Hash Based Cryptosystems. Many algorithms being proposed however are difficult to implement and their performance must be optimised in order for widespread public use.

In 2017, NIST issued a call for algorithms around the world, in order to be able to determine a standard for Public Key Cryptography in the future. It identified that the need for defining a system was approaching fast, based on the criteria that the time for implementation along with development of the algorithms must not exceed that of the development of systems that may break the cryptosystems currently in use. The NIST Post Quantum Cryptography competition was made up of 3 rounds of submissions, with all submissions being open to publick purview, as a result open source contributions were able to siphon out many vulnerable submissions, with only 15 of the initial 69 making it to the final round. As of now, these 15 algorithms were analysed and an overview has been provided. NIST is hopeful that within the next year they shall have sufficiently scrutinized the remaining submissions and their parameters to declare a standards.

# References

*[1] Bernstein D.J., Heninger N., Lou P., Valenta L. (2017) Post-quantum RSA. In: Lange T., Takagi T. (eds) Post-Quantum Cryptography. PQCrypto 2017. Lecture Notes in Computer Science, vol 10346. Springer, Cham.* [*https://doi.org/10.1007/978-3-319-59879-6\_18*](https://doi.org/10.1007/978-3-319-59879-6_18)

*[2]Gidney, Craig, and Martin Ekerå. “How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits.” ArXiv:1905.09749 [Quant-Ph], Dec. 2019. arXiv.org,* [*http://arxiv.org/abs/1905.09749*](http://arxiv.org/abs/1905.09749)*.*

*[3]SANS Institute: Reading Room - Encryption & VPNs. https://www.sans.org/reading-room/whitepapers/vpns/paper/1006. Accessed 10 Dec. 2020.*

*[4]Grover, Lov K. “A Fast Quantum Mechanical Algorithm for Database Search.” ArXiv:Quant-Ph/9605043, Nov. 1996. arXiv.org,* [*http://arxiv.org/abs/quant-ph/9605043*](http://arxiv.org/abs/quant-ph/9605043)*.*

*[5]Hui, Jonathan. “QC — Simon’s Algorithm.” Medium, 25 Apr. 2019,* [*https://jonathan-hui.medium.com/qc-simons-algorithm-be570a40f6de*](https://jonathan-hui.medium.com/qc-simons-algorithm-be570a40f6de)*.*

*[6]Nurhadi, A. I., and N. R. Syambas. “Quantum Key Distribution (QKD) Protocols: A Survey.” 2018 4th International Conference on Wireless and Telematics (ICWT), 2018, pp. 1–5. IEEE Xplore, doi:10.1109/ICWT.2018.8527822*

*[7] Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” SIAM Journal on Computing, vol. 26, no. 5, Oct. 1997, pp. 1484–509. arXiv.org, doi:10.1137/S0097539795293172.*

*[8] Yanofsky, Noson S. “An Introduction to Quantum Computing.” ArXiv:0708.0261 [Quant-Ph], Aug. 2007. arXiv.org,* [*http://arxiv.org/abs/0708.0261*](http://arxiv.org/abs/0708.0261)*.*

*[9]Bogomolec, Xenia et al. “Towards Post-Quantum Secure Symmetric Cryptography: A Mathematical Perspective.” IACR Cryptol. ePrint Arch. 2019 (2019): 1208.*

*[10]Almazrooie, Mishal, et al. “Quantum Exhaustive Key Search with Simplified-DES as a Case Study.” SpringerPlus, vol. 5, no. 1, Sept. 2016, p. 1494. BioMed Central, doi:10.1186/s40064-016-3159-4*

*[11] Smith, Benjamin. “Pre- and Post-Quantum Diffie-Hellman from Groups, Actions, and Isogenies.” ArXiv:1809.04803 [Cs], Dec. 2019. arXiv.org,* [*http://arxiv.org/abs/1809.04803*](http://arxiv.org/abs/1809.04803)*.*

*[12]Wagner, Lane. “(Very) Basic Intro To Elliptic Curve Cryptography.” Qvault, https://qvault.io/2020/09/17/very-basic-intro-to-elliptic-curve-cryptography/. Accessed 10 Dec. 2020.*

*[13]Bernstein, Daniel J. “Introduction to Post-Quantum Cryptography.” Post-Quantum Cryptography, edited by Daniel J. Bernstein et al., Springer, 2009, pp. 1–14. Springer Link, doi:10.1007/978-3-540-88702-7\_1.*

*[14]Shpilrain, Vladimir. “Cryptanalysis of Stickel’s Key Exchange Scheme.” Computer Science – Theory and Applications, edited by Edward A. Hirsch et al., Springer, 2008, pp. 283–88. Springer Link, doi:10.1007/978-3-540-79709-8\_29.*

*[15]Chen, Lily, et al. Report on Post-Quantum Cryptography. NIST Internal or Interagency Report (NISTIR) 8105, National Institute of Standards and Technology, 28 Apr. 2016. csrc.nist.gov, doi:*[*https://doi.org/10.6028/NIST.IR.8105*](https://doi.org/10.6028/NIST.IR.8105)*.*

*[16]Alagic, Gorjan, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, et al. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NIST Internal or Interagency Report (NISTIR) 8309, National Institute of Standards and Technology, 22 July 2020. csrc.nist.gov, doi:*[*https://doi.org/10.6028/NIST.IR.8309*](https://doi.org/10.6028/NIST.IR.8309)

*[17] Schwabe, Peter. Kyber. https://pq-crystals.org/kyber/index.shtml. Accessed 10 Dec. 2020.*

*[18] Schwabe, Peter. Dilithium. https://pq-crystals.org/dilithium/index.shtml. Accessed 10 Dec. 2020.*

*[19] Falcon. https://falcon-sign.info/. Accessed 10 Dec. 2020.*

*[20] Classic McEliece: Intro. https://classic.mceliece.org/. Accessed 10 Dec. 2020.*

*[21]BIKE - Bit Flipping Key Encapsulation. https://bikesuite.org/#spec. Accessed 10 Dec. 2020*

*[22]HQC. http://pqc-hqc.org/implementation.html. Accessed 10 Dec. 2020.*

*[23]“SIKE – Supersingular Isogeny Key Encapsulation.” SIKE – Supersingular Isogeny Key Encapsulation, http://sike.org/. Accessed 10 Dec. 2020.*

*[24]Casanova, et al. GeMSS- A Great Multivariate Short Signature,* [*https://www-polsys.lip6.fr/Links/NIST/GeMSS.html*](https://www-polsys.lip6.fr/Links/NIST/GeMSS.html)*.*

*[25]Schwabe, Peter. SPHINCS+. https://sphincs.org/. Accessed 10 Dec. 2020.*

*[26]Zaverucha, et al. https://microsoft.github.io/Picnic/. Accessed 10 Dec. 2020.*