

Final Report

Quantum-Safe Cryptography: Mitigating Vulnerabilities in Post-Quantum era

Unit: COIT20265

Student 1: AYUSH KESHAR PRASAI (12198371)

Student 2: JALAY SHAH (12232969)

Student 3: RONIT MAHESHWORI (12179419)

Student 4: VIRAJ SINH RAHEVAR (12198387)

Group : PG-15

Project Mentor: MOHAMMED MOHAMMED

Date: 2024/10/07

CQUniversity Australia

Introduction

The topic of this project is Quantum-Safe cryptography: Mitigating Vulnerabilities in Post-Quantum era. The topic relates traditional methods of encryption and how advancement and invention of large-scale quantum computers which is inevitable will affect encryption methods being used today with all the vulnerabilities outlined and ways we can mitigate such vulnerabilities in post-quantum era.

This report conceptualizes advancement from modern era when classical computers are considered rudimentary and a thing of the past. Such era in this report will often be signified as post-quantum era. Post-quantum era refers to the technological advancement in computational ability that any form of security relied upon in modern times through calculation from classical computers are now vulnerable.

The aim of this project is to:

- Identify vulnerabilities and risks related to traditional cryptography with advancement of quantum computers.
- Analyse various encryption algorithms to find vulnerabilities in cryptographic techniques.
- Compare such encryption algorithms to find the safest cryptographic standards.
- Provide general probability of when such vulnerabilities might be a risk factor.
- Advance encryption policies to post-quantum cryptography standards.
- Analyse ways of mitigation of such vulnerabilities.
- Document how post-quantum cryptography might affect traditional cryptography in all industries that relate to business, government and general user.
- Research and document new cryptographic algorithms safe for post-quantum era.
- Provide how transition might occur or is necessary from traditional cryptography techniques.

The problem this project aims to solve will be of analysing various risks associated with traditional cryptography that includes study of vulnerabilities of traditional algorithms from quantum computers. The problem is largely based on how some of the algorithms that are widely in use might not be able to cope with quantum computing when large-scale quantum computers can be used by threat actors to implement present day theoretical algorithms such as Shor's algorithms.

Up until now, cryptography is practised when algorithms are enforced, and standards are maintained. Post-quantum era now imagines an edge in this constant battle between the attacker and the protector through means of exceptionally powerful hardware. Thus, quantum-safe cryptography refers to post-quantum era where standards, algorithms and interpretation of data is practised with depiction of ability of quantum computers. This report focuses primarily on finding vulnerabilities in modern day cryptography from post-quantum era and mitigating such vulnerabilities for quantum-safe cryptography. Furthermore, this report studies vulnerabilities of algorithms used for cryptography today and assesses its risks to post-quantum cryptography.

System Overview

The system in design for this project in essence boils down to two implementation parts i.e.

- a. Showing vulnerabilities in RSA through **Fermat's algorithm**.
- b. Decryption of encrypted message through **Shor's algorithm**.

It is noted that unlike other system implementations, the system overview for this project requires understanding of why the implementation is needed for this project. This can be further simplified when each of the system implementation is broken down into further sub-headings based upon the tasks completed for this project such that each implementation gives accurate portrayal of mission objective for this task. Thus, to design such implementation,

1. Showing vulnerabilities in RSA through Fermat's algorithm ; consists of:

- 1.1 Comparison between present day computational ability to post-quantum computational ability for cryptography.
- 1.2 Listing of present-day algorithms and assessing vulnerabilities.
- 1.3 Technical description of RSA (vulnerable) and its justification.
- 1.4 System implementation to show vulnerability in RSA from quantum computers through Brute-force attack.

Similarly,

2. Decryption of encrypted message through Shor's algorithm; will provide opportunity to:

- 2.1 Derive risk associated with modern day cryptography from quantum computers.
- 2.2 Prove the need for transition to post-quantum cryptographic techniques.
- 2.3 Proposes such transition.

1. Showing vulnerabilities in RSA THROUGH Fermat's algorithm.

To understand post-quantum cryptography, let us first dive into cryptography today. Since the birth of internet, it can be argued that there are equal number of exceptional practitioners that have used the combination of hardware and software to protect and attack the transmitted data in various ways. There is this thin thread of security that has revolutionized cryptography which depends upon standards, algorithms and combination of 0's and 1's. It is due to this standard which everyone follows, algorithms that is calculated, and 0's and 1's that are interpreted in specific manner from which confidentiality, integrity and availability is provided to the data over the internet or anywhere else.

Quantum computers now pose threat of exceptional hardware from which calculations done will surpass the ability of classical computers. Due to such ability, cryptography in practise now will be

vulnerable to post-quantum era. Such vulnerability can be assessed from the start by understanding the computational ability of quantum computers first. Thus,

1.1 Comparison between present day computational ability to post-quantum computational ability for cryptography.

This is the first instance in this project where a technical artefact is referred. The reference is **PG15-Comparision-of-abilities-1.1.docx**. From this artefact, we can summarize some key aspects that are essential **to implement Fermat's algorithm**. The key aspects taken into consideration are:

- **Algorithm efficiency**
Algorithm efficiency in the context of this project and specially to the system implementation to show vulnerability in RSA refers to the ability to do factorization of large numbers. **It is concluded that post-quantum computer's ability to factorize large numbers efficiently will initiate the vulnerability of cryptographic techniques.**
- **Speed**
Speed in which the post-quantum computers operate will be the key to irrelevance of modern cryptographic techniques. Due to speed in which factorization can be performed by post-quantum computers, **RSA is also considered vulnerable.**

1.2 Listing of present-day algorithms and assessing vulnerabilities

This part of the project outlines that out of two types of algorithms in use for cryptography today, symmetric algorithm is not vulnerable to quantum computers whereas all asymmetric algorithms are vulnerable (RSA). The technical artefact it refers to for such conclusion is **PG15-Algorithm-vulnerabilities-1.2.docx**.

- **Symmetric cryptography:**
All symmetric cryptography is quantum safe. Symmetric algorithms such as SHA and AES do not rely upon mathematical calculations making it quantum safe(Azure, 2024).
- **Asymmetric cryptography:**
RSA is an asymmetric cryptography algorithm consisting of pair of keys i.e. public key and private key. The keys are generated with cryptographic algorithms that rely on mathematical calculations based on one-way functions thus making it vulnerable to quantum cryptography. RSA amongst other asymmetric algorithms is chosen in modern day cryptography for its ability to provide key distribution and secrecy along with digital signatures which reduces the use of multiple asymmetric algorithms.

1.3 Technical description of RSA (vulnerable) and its justification

Table 1 - Impact of Quantum Computing on Common Cryptographic Algorithms

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	-----	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

Figure 1 : Impact of quantum computing on Algorithms

From the sub-headings above, RSA is the most relied upon and vulnerable algorithm to quantum computers and is no -longer secure for post-quantum cryptography because of the **Algorithm efficiency** of quantum computers from which prime numbers can be factorized efficiently thus, breaking the RSA which is depicted **through system implementation of Fermat's algorithm below:**

1.4 System Implementation to show vulnerabilities in RSA from quantum computers through brute force attack (Fermat's Factorization Algorithm).

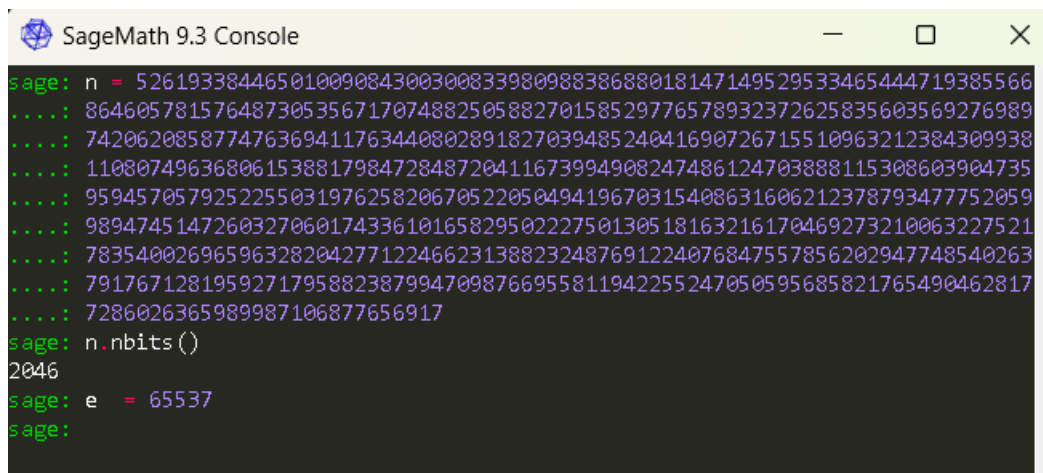
Now that the computational ability of quantum computers to solve complex calculations efficiently is understood, it was realized that RSA, which is the most used cryptographic technique, is vulnerable through factorization of large numbers. All the sub-headings above have led to this system implementation in which we break the RSA using Fermat's factorization algorithm.

To do so, we must first make some assumptions since implementation of this system is done on a classical computer. For implementation purposes we have chosen python and sagemath as our environment. It is referenced in technical artefact **PG15-System-Implementation-Tools.docx**.

Computational ability of quantum computers mainly, **algorithm efficiency** can be assumed in this implementation and is referenced to the technical artefact **PG15-Fermat's-algo-1.4.docx** when:

- Fermat's algorithm can be used in classical computer to find the private key from public key of RSA when keys are chosen very close to each other and not in random **to imitate algorithm efficiency of quantum computer on a classical machine.**
- n is a composite number because it produces two prime numbers when **prime factorization** is conducted.

Therefore, we take the value of n ,

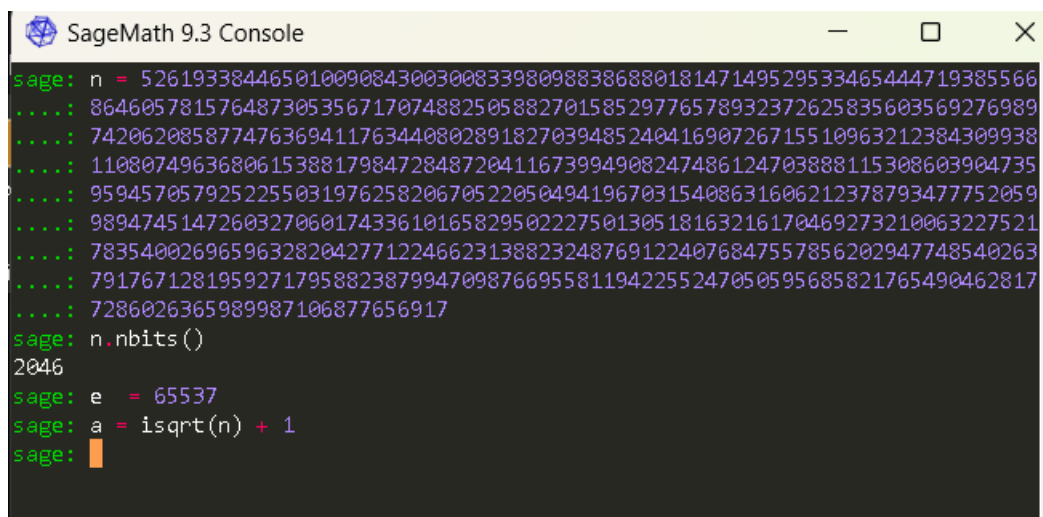


```

SageMath 9.3 Console
sage: n = 5261933844650100908430030083398098838688018147149529533465444719385566
.....: 86460578157648730535671707488250588270158529776578932372625835603569276989
.....: 74206208587747636941176344080289182703948524041690726715510963212384309938
.....: 11080749636806153881798472848720411673994908247486124703888115308603904735
.....: 95945705792522550319762582067052205049419670315408631606212378793477752059
.....: 98947451472603270601743361016582950222750130518163216170469273210063227521
.....: 78354002696596328204277122466231388232487691224076847557856202947748540263
.....: 79176712819592717958823879947098766955811942255247050595685821765490462817
.....: 7286026365989987106877656917
sage: n.nbits()
2046
sage: e = 65537
sage:

```

Here n is a 2046-bit number not taken at a random and exponent (e) is given as 65537, which makes up a private key (n, e) in RSA. d is a private key (d) whose value when determined through factorization from the value of n and e will in theory have broken RSA in post quantum era. To advance,



```

SageMath 9.3 Console
sage: n = 5261933844650100908430030083398098838688018147149529533465444719385566
.....: 86460578157648730535671707488250588270158529776578932372625835603569276989
.....: 74206208587747636941176344080289182703948524041690726715510963212384309938
.....: 11080749636806153881798472848720411673994908247486124703888115308603904735
.....: 95945705792522550319762582067052205049419670315408631606212378793477752059
.....: 98947451472603270601743361016582950222750130518163216170469273210063227521
.....: 78354002696596328204277122466231388232487691224076847557856202947748540263
.....: 79176712819592717958823879947098766955811942255247050595685821765490462817
.....: 7286026365989987106877656917
sage: n.nbits()
2046
sage: e = 65537
sage: a = isqrt(n) + 1
sage:

```

Here we apply a ceiling function a , whose primary objective is to find the value of n to its nearest integer which can be square rooted such that later, the value of p and

q which makes up n i.e. $n = (p,q)$, where p and q are both prime numbers since n is a composite number can be derived using the Euler's totient function.

```
sage: e = 65537
sage: a = isqrt(n) + 1
sage: while True:
.....:     b2 = a^2 - n
.....:     if is_square(b2):
.....:         b = sqrt(b2)
.....:         break
.....:     a = a + 1
.....:
sage: a
72539188337409048434517657668785982436503618029818802387833126880251213106684983
30184745928175617387284965598034198343521347625158194125197938571884477981179390
35054477893594270832856191204357087252629381536836540664036359542928831056089039
36926001552609343022442053757361595080026446437841528409159732216549
sage:
```

Here we introduce a loop to find the value of a into the ceiling function and round it up into the nearest integer until the value of b is found. This is an accurate depiction of a **Brute force attack** where all possible combination of number is tried until the correct value of b is found.

```
.....: 95945705792522550319762582067052205049419670315408631606212378793477752059
.....: 98947451472603270601743361016582950222750130518163216170469273210063227521
.....: 78354002696596328204277122466231388232487691224076847557856202947748540263
.....: 79176712819592717958823879947098766955811942255247050595685821765490462817
.....: 7286026365989987106877656917
sage: n.nbits()
2046
sage: e = 65537
sage: a = isqrt(n) + 1
sage: while True:
.....:     b2 = a^2 - n
.....:     if is_square(b2):
.....:         b = sqrt(b2)
.....:         break
.....:     a = a + 1
.....:
sage: a
72539188337409048434517657668785982436503618029818802387833126880251213106684983
30184745928175617387284965598034198343521347625158194125197938571884477981179390
35054477893594270832856191204357087252629381536836540664036359542928831056089039
36926001552609343022442053757361595080026446437841528409159732216549
sage: p = a + b
sage: q = a - b
sage:
```

We have it given that the value of p and q is as depicted above since $n = (a^2 - b^2)$ as it is a composite number. Since both a and b can be squared and make up the value of n, we can say the value of p and q as such or vice-versa.

Now we have the value of a and b therefore has the value of p and q where $n = (p*q)$.

```

SageMath 9.3 Console
.....: if is_square(b2):
.....:     b = sqrt(b2)
.....:     break
.....:     a = a + 1
.....:
sage: a
72539188337409048434517657668785982436503618029818802387833126880251213106684983
30184745928175617387284965598034198343521347625158194125197938571884477981179390
35054477893594270832856191204357087252629381536836540664036359542928831056089039
36926001552609343022442053757361595080026446437841528409159732216549
sage: p = a + b
sage: q = a - b
sage: phi_n = (p-1)*(q-1)
sage: d = inverse_mod(e,phi_n)
sage: d
17800185137539517698383167821068381411067543879381886530803806922620507101074229
45064979473555663978289445954104491378420576270063441312713561936399517287013999
89469441620498346152638439168409338756928586608239400667001528388878721931781374
60082557354094147194274355977140345941913177555971130235006633535597630270496599
40922933197177165188452818904235179156023616509434680097898474152187396851510136
98483433048819783758619382732412489403348519150211057520208053418471084911305268
86356953308720044791902966786160476361483730986540024432735291754733367533545745
95877161023282772845813963994656394870693323649494959673
sage:

```

Finally, we calculate the Euler's totient function of n i.e. $\phi(n) = (p-1)*(q-1)$. This will give us the value of $\phi(n)$ or (`phi_n`) as in the code picture above. After that the value of d is found using another equation. What this equation basically means is we want to find a number d , which when we multiply by n will give us an intermediate value that can be reduced by mod $\phi(n)$ which will give us 1. Thus, from this equation $(e*d) \equiv 1 \pmod{\phi(n)}$, we calculate the value of **private key d** and prove that RSA is broken through Factorization of prime numbers from public key (e,n) .

2. Decryption of encrypted message through Shor's algorithm:

Shor's algorithm:

Shor's algorithm is the standard bearer for all quantum algorithms. Although it was discovered three decades ago, Shor's algorithm remains the key reason for investment of billions of dollars into quantum technologies as it presents clear and evident danger for national security, financial systems and for all cryptography.

At a high-level Shor's algorithm is easy to understand. It begins with using an integer smaller than the number to be **factored**. The greatest common divisor (GCD) is then calculated classically between these two numbers to determine whether the target number has already been calculated accidentally. This is when a **quantum computer** comes into play. A quantum computer would then be used to determine the results for cryptographically safe number. Only a quantum computer could evaluate the results and determine whether the sought-after integer could have been calculated or not and if so, use a random integer for testing.

To the use of this project, Shor's algorithm is framed such that **modular inverse** is determined to find the value of the private key and therefore **crack the cipher text**.

To initialize this system implementation, we are using Microsoft's Visual studio Code. We then install Qiskit and python to our environment. The codes are referenced from technical artefact **PG15-Shor's_Algo2.ipynb**.

```
%pip install qiskit
%pip install rsa_python
```

Python

These installations make the environment ready to accept the python code and allows the use of Qiskit and python packages to be installed and functions to be used to simulate a quantum computing environment.

```
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: qiskit in c:\users\acer\appdata\local\packages\pytl
Requirement already satisfied: rustworkx<=0.15.0 in c:\users\acer\appdata\local\p
Requirement already satisfied: numpy<3,>=1.17 in c:\users\acer\appdata\local\pack
Requirement already satisfied: scipy<=1.5 in c:\users\acer\appdata\local\packages\
Requirement already satisfied: sympy<=1.3 in c:\users\acer\appdata\local\packages\
Requirement already satisfied: dill<=0.3 in c:\users\acer\appdata\local\packages\j
Requirement already satisfied: python-dateutil<=2.8.0 in c:\users\acer\appdata\lo
Requirement already satisfied: stevedore<=3.0.0 in c:\users\acer\appdata\local\pac
Requirement already satisfied: typing-extensions in c:\users\acer\appdata\local\p
Requirement already satisfied: symengine<=0.11 in c:\users\acer\appdata\local\pac
Requirement already satisfied: six<=1.5 in c:\users\acer\appdata\local\packages\p
Requirement already satisfied: pbr<=2.0.0 in c:\users\acer\appdata\local\packages\
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\acer\appdata\local\j
```

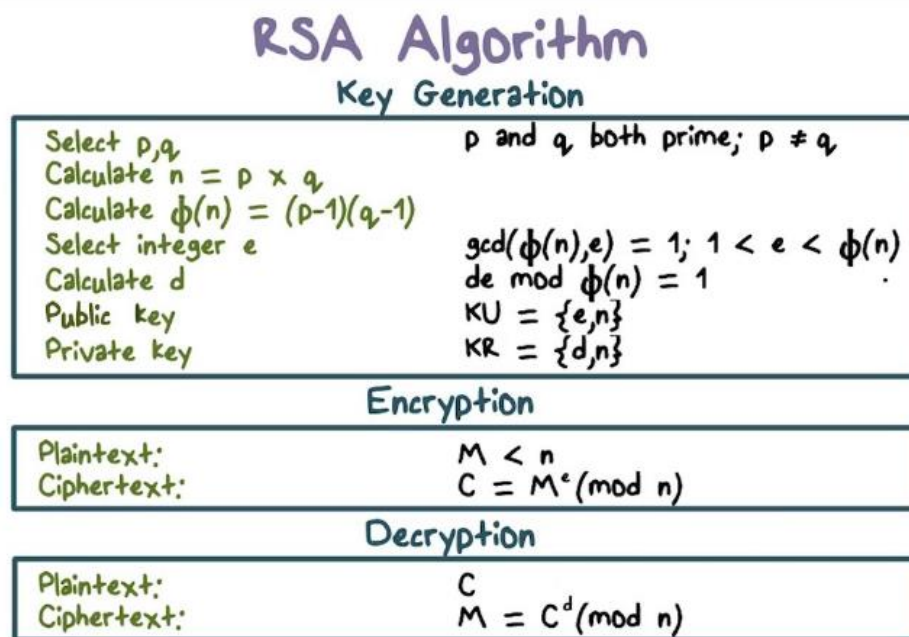
These are the packages successfully installed which are necessary for the system implementation to yield essential outcome.

Now, we import necessary modules to run functions. The modules thus imported are:

```
import numpy as np
from qiskit import *
from math import sqrt, log, gcd
import random
from random import randint
import rsa
```

Python

Making RSA algorithm:



Making of RSA algorithm as depicted in the figure above comprises of three parts in this system implementation. First, we generate the keys i.e. public key and private key as RSA is an asymmetric algorithm. Doing so, we encrypt a plain text using the private key into a cipher text. We then find the value of the private key using Shor's algorithm and then, decrypt the cipher text into the plain text using the private key.

We calculate the modular inverse first to initialize this process which is:

```
def mod_inverse(a, m):  
    for x in range(1, m):  
        if (a * x) % m == 1:  
            return x  
    return -1
```

Python

The modular inverse is calculated at the start of the project as it needs no definitive value, and this function can be called upon later to be used when modular inverse is needed to find the value of d .

Now, we check the primality of the value n . the factorization only works when n comprises of primary numbers which is later essential for this system to calculate the value of p and q which are both prime numbers.

```
def isprime(n):
    if n < 2:
        return False
    elif n == 2:
        return True
    else:
        for i in range(1, int(sqrt(n)) + 1):
            if n % i == 0:
                return False
        return True
```

Python

This returns the value of n to be true or false. In any case the value rounds up until true value is returned thus making sure that n is a prime number.

After that we move to **key generation**:

```
def generate_keypair(keysize):
    p = randint(1, 1000)
    q = randint(1, 1000)
    nMin = 1 << (keysize - 1)
    nMax = (1 << keysize) - 1
    primes = [2]
    start = 1 << (keysize // 2 - 1)
    stop = 1 << (keysize // 2 + 1)
    if start >= stop:
        return []
    for i in range(3, stop + 1, 2):
        for p in primes:
            if i % p == 0:
                break
        else:
            primes.append(i)
    while (primes and primes[0] < start):
        del primes[0]
    # Select two random prime numbers p and q
    while primes:
        p = random.choice(primes)
        primes.remove(p)
        q_values = [q for q in primes if nMin <= p * q <= nMax]
        if q_values:
            q = random.choice(q_values)
            break
    # Calculate n
```

```

# Calculate n
n = p * q
# Calculate phi
phi = (p - 1) * (q - 1)
# Select e
e = random.randrange(1, phi)
g = gcd(e, phi)
# Calculate d
while True:
    e = random.randrange(1, phi)
    g = gcd(e, phi)
    d = mod_inverse(e, phi)
    if g == 1 and e != d:
        break

return ((e, n), (d, n))

```

Python

Key generation refers to the process of generating public key (e,n) and private key (d,n). Two random prime numbers from 1 to 1000 of similar bit size is chosen. Its GCD is calculated and if its similar then the exponent e and private key d is calculated from the random number n such that $n = p * q$ and $p \neq q$.

After generating the value of private and public key we can now encrypt or decrypt a plain text into cipher text and vice versa using the code below.

For encryption:

```

def encrypt(plaintext, package):
    e, n = package
    ciphertext = [pow(ord(c), e, n) for c in plaintext]
    return ''.join(map(lambda x: str(x), ciphertext)), ciphertext

```

Python

For decryption:

```

def decrypt(ciphertext, package):
    d, n = package
    plaintext = [chr(pow(c, d, n)) for c in ciphertext]
    return ''.join(plaintext)

```

Python

Testing the encryption and decryption functions:

Thus, after completion of the above implemented steps, we can now generate keys, encrypt and decrypt using few lines of codes which are:

Generate keypair:

```
import rsa
import rsa_python
bit_length = int(input("Enter bit length: "))

public_k, private_k = generate_keypair(2**bit_length)
```

Python

Public key (Public_k) and private key (private_k) is generated into the value generate_keypair.

Encryption:

```
plain_txt = input("Enter a message: ")
cipher_txt, cipher_obj = encrypt(plain_txt, public_k)

print("Encrypted message: {}".format(cipher_txt))
```

Python

Plain text (plain_txt) is entered as an input and encrypted into a cipher text (cipher_txt) and printed out.

Decryption:

```
print("Decrypted message: {}".format(decrypt(cipher_obj, private_k)))
```

Python

Can now print the decrypted message when private key (private_k) is known. The message cannot be decrypted else.

At this point in system integration, we will start **framing Shor's algorithm**.

```

qasm_sim = qiskit.Aer.get_backend('qasm_simulator')
def period(a,N):

    available_qubits = 16
    r=-1

    if N >= 2**available_qubits:
        print(str(N)+' is too big for IBMQX')

    qr = QuantumRegister(available_qubits)
    cr = ClassicalRegister(available_qubits)
    qc = QuantumCircuit(qr,cr)
    x0 = randint(1, N-1)
    x_binary = np.zeros(available_qubits, dtype=bool)

    for i in range(1, available_qubits + 1):
        bit_state = (N%(2**i)!=0)
        if bit_state:
            N -= 2**(i-1)
            x_binary[available_qubits-i] = bit_state

    for i in range(0,available_qubits):
        if x_binary[available_qubits-i-1]:
            qc.x(qr[i])
    x = x0

```

```

while np.logical_or(x != x0, r <= 0):
    r+=1
    qc.measure(qr, cr)
    for i in range(0,3):
        qc.x(qr[i])
    qc.cx(qr[2],qr[1])
    qc.cx(qr[1],qr[2])
    qc.cx(qr[2],qr[1])
    qc.cx(qr[1],qr[0])
    qc.cx(qr[0],qr[1])
    qc.cx(qr[1],qr[0])
    qc.cx(qr[3],qr[0])
    qc.cx(qr[0],qr[1])
    qc.cx(qr[1],qr[0])

    result = execute(qc,backend = qasm_sim, shots=1024).result()
    counts = result.get_counts()

    results = [[],[ ]]
    for key,value in counts.items():
        results[0].append(key)
        results[1].append(int(value))
    s = results[0][np.argmax(np.array(results[1]))]
    return r

```

Total of 16 qubits are made available from which QuantumRegister (qr), ClassicalRegister (cr) and QuantumCircuit (qc) is operated. The above operation returns the value of r.

We now implement the code to find the value of p and q from the given N and r. we apply the shors_breaker() as:

```
def shors_breaker(N):
    N = int(N)
    while True:
        a=randint(0,N-1)
        g=gcd(a,N)
        if g!=1 or N==1:
            return g,N//g
        else:
            r=period(a,N)
            if r % 2 != 0:
                continue
            elif pow(a,r//2,N)==-1:
                continue
            else:
                p=gcd(pow(a,r//2)+1,N)
                q=gcd(pow(a,r//2)-1,N)
                if p==N or q==N:
                    continue
                return p,q
```

Python

This string of code calculates the gcd to return p and q. the random integer a is used as a ceiling function to round up the value to nearest integer through subtraction of N from 1 i.e. (N-1).

Modular inverse is calculated:

```
def modular_inverse(a,m):
    a = a % m;
    for x in range(1, m) :
        if ((a * x) % m == 1) :
            return x
    return 1
```

Python

```
N_shor = public_k[1]
assert N_shor>0,"Input must be positive"
p,q = shors_breaker(N_shor)
phi = (p-1) * (q-1)
d_shor = modular_inverse(public_k[0], phi)
```

Python

The modular inverse gives us the value of d_shor, shor's breaker gives is the value of p and q, N_shor is the public key made available for all and phi is calculated from p and q.

Finally, the cipher text can now be cracked and **deciphered**:

```
print('Message Cracked using Shors Algorithm: {} '.format(decrypt(cipher_obj, (d_shor,N_shor))))
```

Python

2.1 Derive risk associated with modern day cryptography from quantum computers.

The technical artefact to refer to for this section is **PG15-Risk-Assessment-2.1.docx**. through this document we derive various risks with modern day cryptography from quantum computers amongst which the primary risk mitigating factor is transitioning to post-quantum cryptography.

The risk assessment table derived is depicted below:

Risks identified	who is harmed and how?	What are you doing to control risks?	What further action can be taken	Who carries out the action?	When is the action needed to be carried out?	Process
1. Inadequate preparation	Stakeholders, business, organization, government	Security audits	Transitioning to post-quantum cryptography techniques	organization	Medium term plan	Doing
2. Vulnerabilities in current cryptography techniques.	Government, business	Informed about technical advancement	Training and awareness	Government, educational bodies, organization	immediate	Doing
3. Impact on data security	Business Government stakeholders	Adopt quantum safe cryptography techniques	Inform and train for awareness	Educational bodies, Business organization	Medium term	To do
4. Proliferation of quantum technology	stakeholders	Inform about technological advancement	Transition and train to adapt to post-quantum computers	Organization, Educational bodies, regulatory bodies, government	Long term	To do

2.2 Provide the need for transition to post-quantum cryptographic techniques

The need for transition is primarily realised through implementation of **Shor's algorithm** as it gives significance to the time frame needed for transitioning to post-quantum cryptography. It signifies that although discovered almost 30 years ago, post-quantum era is inevitable and might pose limited timeframe to organizations which poses threat to the security and operational efficiency of the data and standards respectively. This is represented on the technical artefact **PG15-Need-for-transition-2.2.docx**.

3. Delivered Technical Artefacts

Name	File	Description	PDF?
Comparison between present day computational ability to post-quantum computational ability for cryptography	PG15-Comparision-of-abilities-1.1.docx	Detailed outlining of post-Quantum computational ability, comparison from classical computers to post-quantum computers, outlining computational limit of classical computers, assessing vulnerabilities of cryptographic techniques due to the limit.	Yes
Listing present day algorithms and assessing vulnerabilities	PG15-Algorithm-vulnerabilities-1.2.docx	Due to reliability on calculational ability for cryptography, symmetric and asymmetric algorithms are listed and categorized upon vulnerability due to calculational ability of quantum computers.	Yes
Tools and methodology used to implement the system	PG15-System-Implementation-Tools.docx	Use of tools such as sagemath, python and qiskit, visual studio code, GitHub, MS teams etc.	Yes
Fermat's Factorization algorithm using sagemath justification and algorithm	PG15-Fermat's-Facto-Algo-1.4.docx	Public key (e,n) is identified or assumed, and value of private key (d) is factorized using Euler's totient function and ceiling loop from public key to break RSA.	Yes
Shor's Algorithm is framed to use qubits from qiskit in python on visual studio code for factorization of prime numbers to encrypt and decrypt messages.	PG15-Shor's_Algo2.ipynb	First key pair is generated, and modular inverse is calculated. Using packages encryption and decryption is made possible. Shor's algorithm is framed and qiskit is used to generate a quantum computing ability through which both encryption and decryption is made possible.	No
Risk assessment of cryptography with quantum computers.	PG15-Risk-Assessment-2.1.docx	Risk identification, risk analysis and risk mitigation for post-quantum era.	Yes
Need for transition outlines the causes that are triggered from implementation of Shor's	PG15-Need-for-transition-2.2.docx.	Document explains the need for transition to quantum cryptography from modern cryptography and justifies various reasons.	Yes

algorithm that signifies the inevitability of post-quantum era.			
Progression of the kanban board	PG15-Progression-Of-Kanban.docx	Documents the progression of tasks according to the allocation from week 1 to week 13.	Yes
Gantt chart	PG15-Gantt-chart.docx	Gantt chart shows the progression of tasks allocated from start to the end.	Yes
Proposing Transition	PG15-Propose-Transition.docx	Proposing the system implementation of post-quantum cryptography algorithm Kyber.	Yes

4. Contributions

Student Name	Percent	Summary of Contributions	Technical Lead on Artefacts
Ayush Keshar Prasai	25%	<ul style="list-style-type: none"> • Researched and implemented Tools and Methodology needed for system implementation for this project. • Researched Fermat's Factorization Algorithm and its implementation and outcome. • Researched Shor's Algorithm and its deliverables. 	<ul style="list-style-type: none"> • PG15-System-Implementation-Tools.docx • PG15-Fermat's-Facto-Algo-1.4.docx • PG15-Shor's_Algo2.ipynb
Jalay Shah	25%	<ul style="list-style-type: none"> • List the algorithms and assesses vulnerabilities in terms of computational ability of quantum computers. • Assesses risks associated with use of widely used cryptography along with risk assessment table 	<ul style="list-style-type: none"> • PG15-Algorithm-vulnerabilities-1.2.docx • PG15-Risk-Assessment-2.1.docx
Ronit Maheshwari	25%	<ul style="list-style-type: none"> • Compares computational abilities between era to point the hardware superiority of post quantum era. • Depicts the progression of tasks on the kanban board. 	<ul style="list-style-type: none"> • PG15-Comparision-of-abilities-1.1.docx • PG15-Progression-Of-Kanban.docx
Viraj Sinh Rahevar	25%	<ul style="list-style-type: none"> • Created the Gantt chart • Proposed a transition with use of post quantum cryptography algorithm • Researched kyber as post-quant cryptography algorithm 	<ul style="list-style-type: none"> • PG15-Gantt-chart.docx • PG15-Propose-Transition.docx • Need for transition

		and proposed transition to quantum cryptography.	
--	--	--	--

5. Next Steps

2.3 propose transition

The next step to take would be to propose a transition with the use of post-quantum cryptography algorithm as referred to from **PG15-Propose-Transition.docx**. This document proposes such transition by:

- Introducing Kyber.
- Justifying its design and functionality.
- Evaluating Kyber's security.
- Imagining its challenges and future implementation.

1 TABLE OF CONTENTS

2	To do Task 1: Outline Post-Quantum Computational Limit.....	1
3	To do Task 2: Compare Present Day Computational Limit to Post Quantum Computational.....	2
4	To do Task 3: Identify and Outline Vulnerabilities in Present Computational Ability and Cryptographic Techniques	4
4.1	Computational Limits.....	4
4.2	Cryptographic Techniques	5
5	References	6

2 TO DO TASK 1: OUTLINE POST-QUANTUM COMPUTATIONAL LIMIT

In this technological era, with the advent of quantum computing technology, it aims to solve potential complex problems that are not able to be done by random computers. With the introduction of such technologies, it is likely to propose various post-quantum computational limits, security challenges, ethical and social concerns. In simple terms, quantum computers are working on the principles of quantum mechanics in order to perform all calculations in such a way that quantum bits are able to process vast amount of data (Rieffel et al., 2024). The main impact of such technologies is on cryptography where Shor's algorithm is being used to factorize large integers. In the post era, there are likely to be various computational limits due to immense increase of quantum computing technology in today's era. It is undeniable that quantum computers are able to excel specific tasks but could not be able to do various other tasks as of now. For example, it can be said that problems like database searching could be sped up with Grover's algorithm but it is quadratic which means it is not required for such scenarios. Instead of this, classical computers are suited best for such applications rather than focusing on the quantum advantages (Gill & Buyya, 2024).

There are even various challenges that could be faced with quantum computing based on development and scalability. This is because quantum computers are highly sensitive to different environmental disturbances which means there is a dire need for having an advanced error correction system along with

stable qubit architecture. As these metrics or applications are unavailable, it is likely to pose certain limitations on the quantum technology (Desdentado et al., 2024). This also represents that for the foreseeable future, both quantum computing and classical computing will coexist and will complement each other. Furthermore, in the case of research and development also for post-quantum computing, it is vibrant and is burgeoning at an exponential speed. In the coming future, there will be robust post quantum cryptographic standards just like RSA and ECC algorithms which could be used for standardizing the post-quantum cryptographic systems. Furthermore, there will be improved quantum algorithms available in the market which will make sure all potential security issues are patched and it is able to solve different types of problems. Finally, there will also be an appropriate quantum hardware available that would be able to overcome physical and engineering challenges that are being faced in today's world for constructing practical and scalable quantum computers (Stackpole, 2024).

There will be different ethical and social implications available that would have adverse impact on human beings. As quantum computing will be able to perform higher power computational, it means it will be able to break out today's existing cryptographic systems which could lead to the data privacy and security related issues. However, to avoid such things, there would be a need for ensuring a secure transition to the post-quantum cryptographic standards. Moreover, the use of such power can be misused by offenders in the case of surveillance and data analysis which must also be regulated. Henceforth, the post-quantum computing limits is not representing the end point but showing the new beginning where computation can be understood (Rieffel et al., 2024).

3 TO DO TASK 2: COMPARE PRESENT DAY COMPUTATIONAL LIMIT TO POST QUANTUM COMPUTATIONAL

The tabular matrix is attached below that clearly explains present day computational limit with respect to post quantum computational limit.

Aspect	Present Day Computational Limit	Post-Quantum Computational Limit
Cryptographic Security	In the current era, the cryptographic security mainly relies over RSA, ECC and other related algorithm, however, these algorithms are sometimes susceptible to classical attacks.	On the other hand, it mainly requires post-quantum cryptography i.e. lattice-based and code based which helps in resisting all types of quantum related attacks.
Algorithm Efficiency	In the context of classical algorithms, there are various tasks that are computationally infeasible such as factorizing of large numbers (Goldstein, 2023).	In the post quantum computational limit, the use of Shor's algorithm could be done in such a way that it will be able to make factorization feasible.
Speed	When it comes to speed, it is limited by classical transistor based hardware only.	However, in the case of quantum computing, its parallelism mainly offers the speedups for different problems like Grover's search algorithm.
Error Sensitivity	The error sensitivity rate for current limit is robust and even error rates can be manageable easily with current hardware resources.	In the case of post-quantum era, it is highly sensitive to environmental disturbances which means there would be a need for advanced error correction mechanisms.
Hardware Scalability	The hardware scalability is also possible in the current era with Moore's law that is approaching physical limits in the coming years.	The quantum computing hardware is still in early stages which means there would be significant challenges faced when qubits are to be scaled up.
Application Scope	Currently, the wide range of application are supported followed by optimization, machine learning and even data processing.	The quantum computers are perfect fit for higher complex tasks like optimization, natural science simulations and cryptographic operations.
Development Stage	It is well established and continuous increments are being made (Quantum News, 2024).	The field is evolving rapidly but still there are various challenges for quantum

		computing hardware, algorithms and software.
Ethical Considerations	The privacy and security could be maintained easily with current cryptographic standards available in the market.	The privacy and security could not be maintained easily with cryptographic standards as it can break currently available encryptions easily and there is no responsible use.
Integration	The current systems cannot combine with quantum computers as of now.	The hybrid systems can amalgamate with classical computers in an easy manner.
Future Outlook	In the future, the incremental improvements are approaching physical and theoretical limits for the classical computing.	Herein, the paradigm shift is seen for various advancements in the field of science and it can compete with classical computing easily (Quantum News, 2024).

4 TO DO TASK 3: IDENTIFY AND OUTLINE VULNERABILITIES IN PRESENT COMPUTATIONAL ABILITY AND CRYPTOGRAPHIC TECHNIQUES

Under this section, a detailed analysis will be performed on vulnerabilities that are primarily present in computational ability as well as cryptographic techniques.

4.1 COMPUTATIONAL LIMITS

The table attached below lists out all issues that are likely to be faced with quantum computing currently and in the coming years.

Issues	Description
Scalability Related Issues	<ul style="list-style-type: none">• In the coming years, the transistor size is likely to approach the atomic scale which means the Moore’s law will no longer will applicable.• Due to the increased processing power, there is an increased heat and the cooling technologies are likely to suffer.

	<ul style="list-style-type: none">• For higher power consumption, it demands for higher energy that leads to sustainability related issues (S et al., 2024).
Algorithmic Efficiency	<ul style="list-style-type: none">• There are certain problems that could be faced like factoring large integers and computing discrete logarithms which are not able to solve problems easily.• There are multiple problems classified like NP-hard and NP-complete which does not have polynomial time solutions. As a result of this absence, it is likely to limit solvability with classical methods available (Baker, 2023).
Processing Speed	<ul style="list-style-type: none">• In the context of processing speed, it can be said that classical computers is able to process tasks sequentially as there is a limited speed and it cannot perform potential parallelism.• For the latency and bandwidth also, there are communication delays and limited data transfers with current systems.

4.2 CRYPTOGRAPHIC TECHNIQUES

Unlike computational problems, there are various vulnerabilities with cryptographic techniques which have been explained below in a tabular matrix.

Issues	Description
Algorithm Specific Vulnerabilities	<ul style="list-style-type: none">• For the current encryption standard i.e. RSA, it is susceptible to factorization attacks which means it can be compromised.• For the ECC algorithm also, it relies over the discrete logarithm problems and it vulnerable to specific mathematical breakthroughs.• In the context of symmetric encryption, the use of techniques like AES are secure but is vulnerable to various attacks (S et al., 2024).
Key Management	<ul style="list-style-type: none">• Key management is very essential in cryptography and it is challenging now as storage space can be compromised.• In the case of distribution and exchange also, there is a risk of interception.• The length of keys is short and poor which means it is vulnerable to brute forcing.

Implementation Flaws	<ul style="list-style-type: none">• There are various vulnerabilities in cryptographic software and it can be exploited. For example, Heartbleed vulnerability• The risk of side channel attack is very high that results in timing attacks, power consumption and many more (Morstyn & Wang, 2024).
Social Engineering	<ul style="list-style-type: none">• Owing to social engineering attacks, the cryptographic security can be compromised by deceiving all users into divulging sensitive information.• The insider threat is also possible which may allow accessing cryptographic keys easily.
Post Quantum Threats	<ul style="list-style-type: none">• The post quantum threats are also possible where algorithms like Shor and Grover are likely to pose future threats based on current cryptographic standards.• The overall transition period is vulnerable because the quantum is able to compromise everything in the future (Morstyn & Wang, 2024).

5 REFERENCES

Baker, M. A. (2023, June 26). The environmental impact of Quantum Computing - Maher Asaad Baker - Medium. *Medium*. <https://maher-asaad-baker.medium.com/the-environmental-impact-of-quantum-computing-386386990785>

Desdentado, E., Calero, C., Moraga, M. Á., Serrano, M., & García, F. (2024). Exploring the trade-off between computational power and energy efficiency: An analysis of the evolution of quantum computing and its relation to classical computing. *Journal of Systems and Software*, 112165. <https://doi.org/10.1016/j.jss.2024.112165>

Gill, S. S., & Buyya, R. (2024). Transforming Research with Quantum Computing. *Journal of Economy and Technology*. <https://doi.org/10.1016/j.ject.2024.07.001>

Goldstein, P. (2023, October 17). What are the security implications of quantum computing? *Technology Solutions That Drive Business*. <https://biztechmagazine.com/article/2023/10/what-is-quantum-computing-perfcon>

- Morstyn, T., & Wang, X. (2024). Opportunities for quantum computing within net-zero power system optimization. *Joule*. <https://doi.org/10.1016/j.joule.2024.03.020>
- Quantum News. (2024, January 25). Quantum-Resistant FALCON algorithm challenges cryptography, promises secure IoT future. *Quantum Zeitgeist*. <https://quantumzeitgeist.com/quantum-resistant-falcon-algorithm-challenges-cryptography-promises-secure-iot-future/>
- Rieffel, E. G., Asanjan, A. A., Alam, M. S., Anand, N., Neira, D. E. B., Block, S., Brady, L. T., Cotton, S., Izquierdo, Z. G., Grabbe, S., Gustafson, E., Hadfield, S., Lott, P. A., Maciejewski, F. B., Mandrà, S., Marshall, J., Mossi, G., Bauza, H. M., Saied, J., . . . Biswas, R. (2024). Assessing and advancing the potential of quantum computing: A NASA case study. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2024.06.012>
- S, B. K., A, S., M, M., Prasad, Y. J. D. S., & Ahmad, I. (2024). Quantum Computing Basics, applications and future Perspectives. *Journal of Molecular Structure*, 137917. <https://doi.org/10.1016/j.molstruc.2024.137917>
- Stackpole, B. (2024, January 11). *Quantum computing: What leaders need to know now | MIT Sloan*. MIT Sloan. <https://mitsloan.mit.edu/ideas-made-to-matter/quantum-computing-what-leaders-need-to-know-now>

ALGORITHM VULNERABILITY ASSESSMENT

Algorithm type	Algorithms	Vulnerability in post quantum era	Justification
symmetric algorithm	Advanced Encryption Standard(AES)	None	Quantum computer on Grover's algorithm with clock rate of 2THz would take a million years to break AES-128 (JimakosJimakos, 2024).
	Twofish	None	Safer as it supports only 256 bit and not vulnerable to bruteforce.
	Serpent	None	High security margin only second to AES.
	Data Encription Standard (DES)	None, Outdated	Uses 56-bit key (Grabbe, 2022).
	Blowfish	None	Can take upto 448 bits.
	Secure Hashing algorithm (SHA)	None	Uses 256 bits hashing algorithm against collision vulnerabilities and brute force attacks (Harish, 2024).
Asymmetric algorithm	Rivest Shamir Adleman (RSA)	Vulnerable	Succeptable to big enough quantum computer that uses qubits.
	Elliptic Curve Cryptography (ECDSA, ECDH)	Vulnerable	Quantum computers can solve ECDLP efficiently (ExperiMENTAL, 2024).
	Finite Field Cryptography (DSA)	Vulnerable	Quantum computational ability can crack it in days if not hours.

References

JimakosJimakos 75511 gold badge55 silver badges1111 bronze badges, Daniel SDaniel S 24.7k11 gold badge2929 silver badges6767 bronze badges and PrinceofmillerovoPrinceofmillerovo 1522 bronze badges (2024) *Is AES-128 quantum safe?*, *Cryptography Stack Exchange*. Available at: <https://crypto.stackexchange.com/questions/102671/is-aes-128-quantum-safe> (Accessed: 26 August 2024).

Grabbe, J.O. (2022) *The des algorithm illustrated, The DES algorithm Illustrated*. Available at: <https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm> (Accessed: 25 August 2024).

Harish, A. (2024) *2024 complete guide to sha encryption types*, *SecureW2*. Available at: <https://www.securew2.com/blog/what-is-sha-encryption-sha-256-vs-sha-1#:~:text=SHA%2D256%20is%20secure%20due,a%20more%20secure%20hashing%20algorithm.> (Accessed: 26 August 2024).

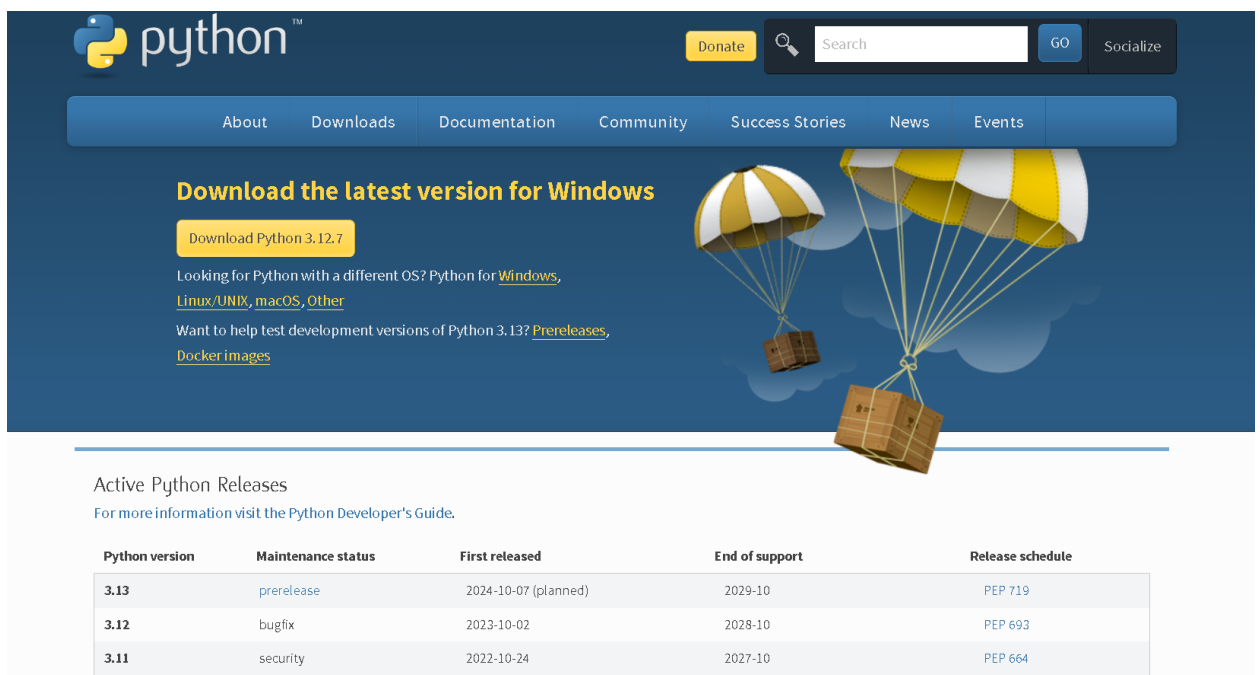
ExperiMENTAL (2024) *Does quantum computing spell the end for elliptic curve cryptography? not quite!*, *Medium*. Available at: <https://medium.com/@jamie.brian.gilchrist/does-quantum-computing-spell-the-end-for-elliptic-curve-cryptography-not-quite-6f22ee202851#:~:text=The%20Quantum%20Threat%20to%20ECC&text=These%20advanced%20machines%2C%20armed%20with,Curve%20isn't%20sufficient%20anymore.> (Accessed: 26 August 2024).

Tools and Methodology of system implementation

Tools in use are:

- Python

Python was used in this project extensively since the system implementations required calculations and python was the best of several languages to integrate the mathematical calculations with due to its easily understandable codes.



The screenshot shows the Python.org website. At the top, there is a navigation bar with links: About, Downloads, Documentation, Community, Success Stories, News, and Events. Below this, a large banner features the text "Download the latest version for Windows" and a button "Download Python 3.12.7". To the right of the banner is an illustration of two parachutes carrying boxes. Below the banner, there is a section titled "Active Python Releases" with a link to the "Python Developer's Guide". A table follows, listing the active Python versions and their support schedules.

Python version	Maintenance status	First released	End of support	Release schedule
3.13	prerelease	2024-10-07 (planned)	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693
3.11	security	2022-10-24	2027-10	PEP 664

Pythons download link: <https://www.python.org/downloads/>

Setting the path:

System Properties



Computer Name Hardware Advanced System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

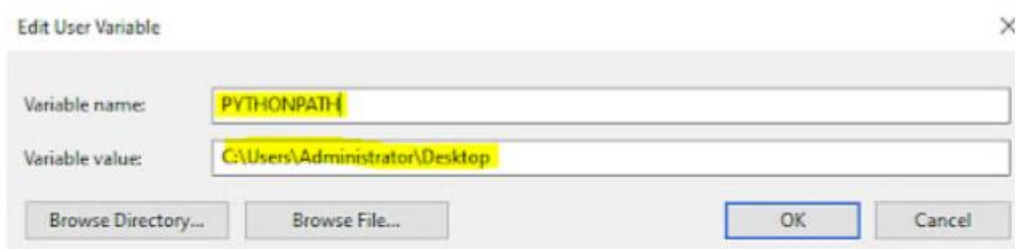
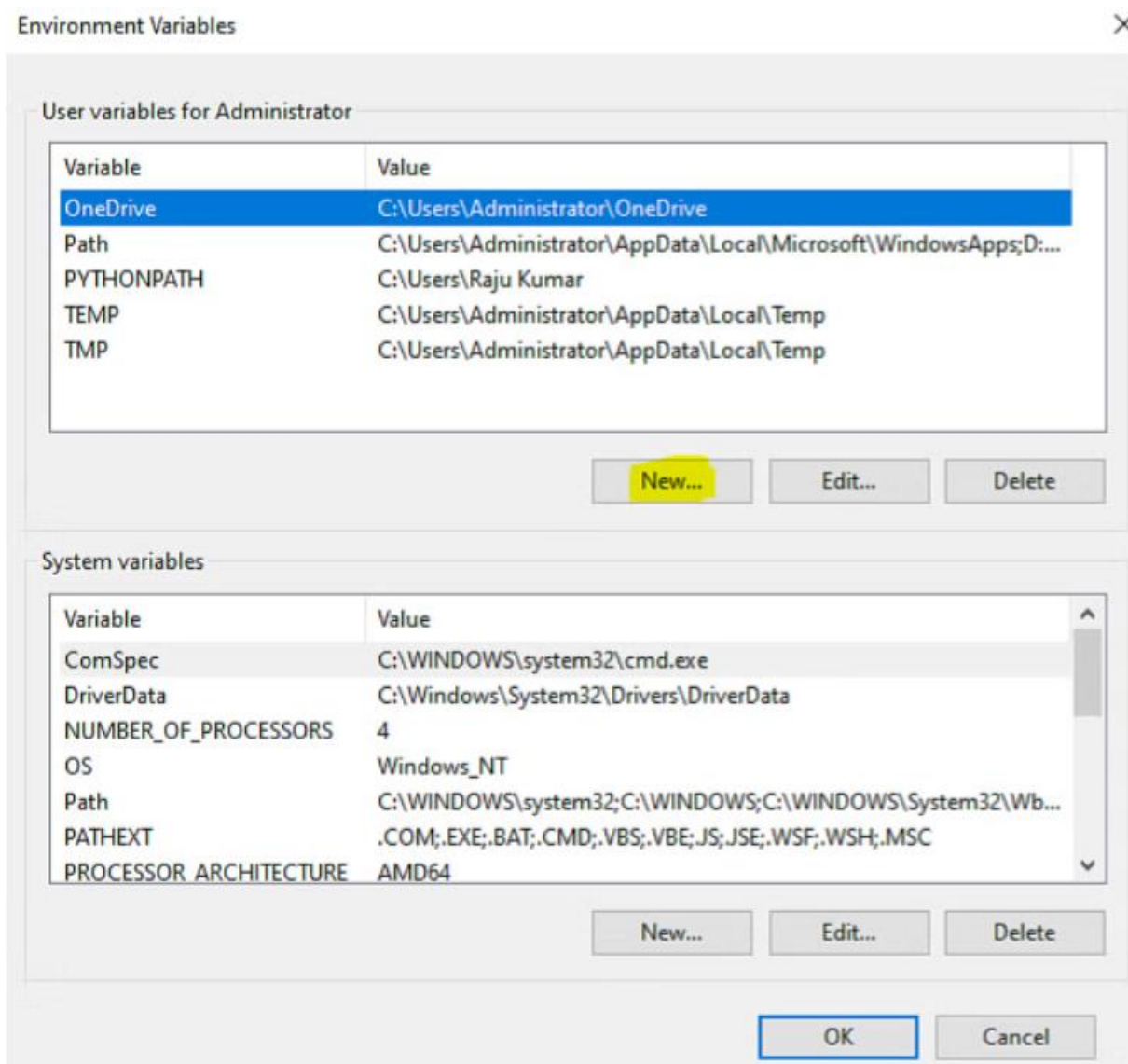
Settings...

Environment Variables...

OK

Cancel

Apply



Python can now be opened in the terminal.

- Sagemath



Sagemath is the environment used to write python code for one of the system implementations.

Other SageMath downloads

Installation guide: [What/how to download](#)

Source (stable) [Download complete source](#)

Source (devel) The [latest development release](#).

To get the source of the latest development release, choose a download mirror and follow relevant instructions on the mirror page.

Apple macOS [Download macOS binaries](#) (3-manifolds)

Docker image [SageMath Docker images](#)

Source (old) [Older Versions of SageMath](#)

Sage math download link: <https://www.sagemath.org/download.html>

- Visual studio code

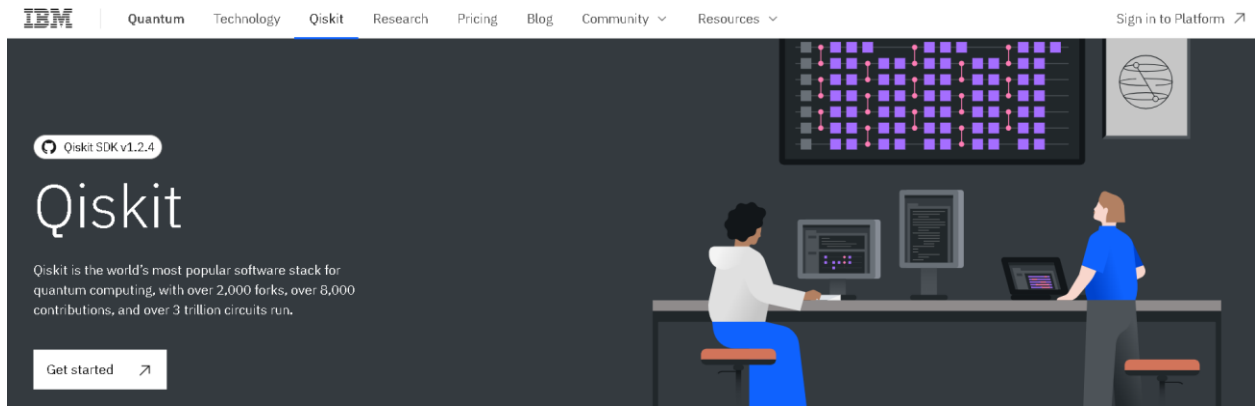


An environment used for installing packages, dependencies and calculating with python and qiskit.

Visual studio code download link: <https://code.visualstudio.com/docs/?dv=win64user>

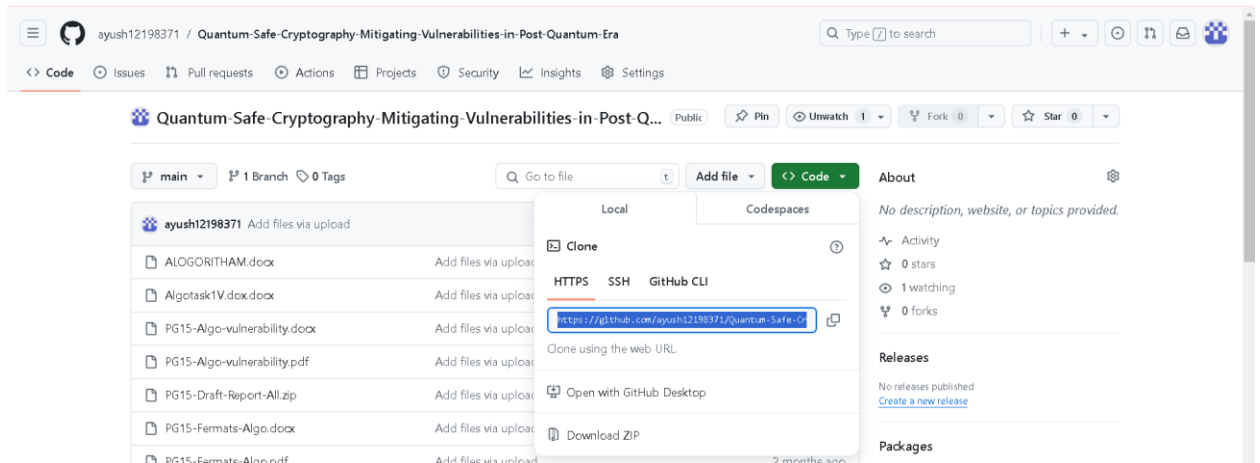
- Qiskit

An IBM development kit for running quantum computers.



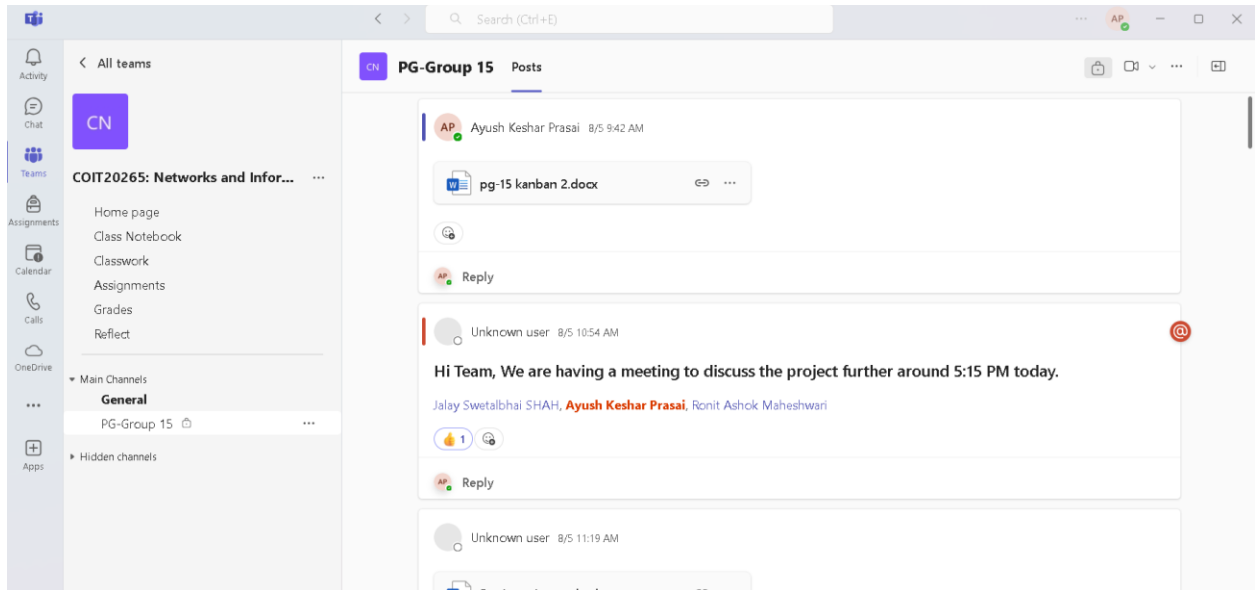
Getting started: <https://docs.quantum.ibm.com/guides/hello-world>

- **Github**
Tool to collaborate and maintain dependency used in recommendation with the institution.



Clone link : <https://github.com/ayush12198371/Quantum-Safe-Cryptography-Mitigating-Vulnerabilities-in-Post-Quantum-Era.git>

- **Microsoft Teams**
Communication tool used for collaboration in real time and holding meetings or exchanging information.



- Agile Methodology

Submissive use of agile ceremonies such as:

Daily standup, Sprint retrospective, daily scrum etc.

Rsa has two keys private key and public key. You sign something using private key and verify the signature using public key. Fermat's algorithm can be used in classical computer to find the private key from public key of RSA when keys are chosen very close to each other and not in random. However, this is relevant to post-quantum cryptography since quantum computers can easily use brute force since it has powerful computational ability.

(e, n) public key

(d) private key

N is a very large number 2000-4000 bits long

e is usually 65537 not a secret (public key)

$N = p \cdot q$, generate two random prime values p and q to multiply it and generate a random composite number n

Breaking rsa:

has given e and n

Factor n into p and q

We use euler totient function to calculate the totient of n

i.e. $\phi(n) = (p-1) \cdot (q-1)$

given, $e \cdot d \equiv 1 \pmod{\phi(n)}$ is congruent to

now we know that n is a composite number because it produces two prime numbers when prime factorization is conducted

ferment's factorization algorithm is effective only when the prime numbers p and q are not very different from each other

$$n = (a^2 - b^2) = (a+b)(a-b)$$

$$b^2 = a^2 - N$$

For this to work we must find the value of b as a squared number to balance the equation above

square root of N is where we want to start and move slowly up through a to find a plausible value for b

for this we use a ceiling function.

Initial guess of a would be $a = \text{square root of } N$ (integer right above it)

In order for this to work, we add 1 to a to find number that are b squared (going to the next integer above it)

what this equation basically means is we want to find a number d, which when we multiply by n will give us an intermediate value that can be reduced by mod $\phi(n)$ which will give is 1.

Code:

n =

```
5261933844650100908430030083398098838688018147149529533465444719385566864605781
5764873053567170748825058827015852977657893237262583560356927698974206208587747
6369411763440802891827039485240416907267155109632123843099381108074963680615388
1798472848720411673994908247486124703888115308603904735959457057925225503197625
8206705220504941967031540863160621237879347775205998947451472603270601743361016
5829502227501305181632161704692732100632275217835400269659632820427712246623138
8232487691224076847557856202947748540263791767128195927179588238799470987669558
119422552470505956858217654904628177286026365989987106877656917
```

random number

n.nbits()

a = isqrt(n) + 1

a

while True:

....: b2 = a^2 - n

....: if is_square(b2):

....: b = sqrt(b2)

....: break

....: a = a + 1

a

b

p = a + b

q = a - b

e = 65537

phi_n = (p-1)* (q-1)

d = inverse_mod(e,phi_n)

Risk assessment for modern day cryptography from quantum computers:

Overview:

Current cryptographic techniques are vulnerable as advancement of quantum computing poses new threats significantly to those that rely upon the use of traditional algorithms. This may include business organizations, private collectors, government bodies or any other imaginable industry that uses modern cryptography. This risk assessment examines such risks associated with quantum computing.

Risk identification:

- Inadequate preparation:
Lack of transition plans from organizations and failure to implement quantum resisting algorithms risks future vulnerabilities.
- Vulnerabilities in current cryptographic algorithms:
Quantum computers and its algorithms such as Shor's algorithm can factor large numbers efficiently, breaking algorithms such as RSA and ECC.
- Impact on Data security:
Data breaches now can be decrypted later which is a serious threat, "store now decrypt later".
- Proliferation of quantum computing:
As quantum technology advances, access to quantum computers results in increase of threat actors.

Risk analysis

Likelihood of quantum threats:

- Short term: low, quantum computers are yet to advance to a point where current cryptographic techniques are at immediate threat.
- Medium term: Moderate, practical quantum attacks are researched to be possible as the quantum technology advances.
- Long term: High, 10+ years after, numerous threat actors and knowledge of quantum computers is likely to pose new risks.

Potential impact:

- Financial costs due to data breaches, regulatory fines and replacement of security systems.

- Loss of data also results to reputation damage.
- Confidential data loss has severe implications for risks.

Risk mitigation strategy

1. Transition to post-quantum cryptography:
Adoption of quantum safe cryptographic techniques, regulations and standards.
2. Security audits:
Evaluation and updating the security protocols and replacing them to meet new standards.
3. Training and awareness:
Stakeholders must be informed of the risks associated and adequate training must be provided to the staff at hand.
4. Adopting encryption practices:
Quantum safe algorithms such as lattice-based, hash-based and polynomial cryptography must be integrated.
5. Informed about the technological advancement:
It is essential to monitor breakthroughs in quantum technology to adapt to the strategies as needed.

Conclusion

Proactive measures are needed to mitigate risks from quantum threats although it does not pose any immediate threats to modern cryptography techniques. Government bodies and business organizations that collect confidential data cannot take risk of a data breach and therefore transitioning to post-quantum cryptographic techniques is a must.

Risk assessment table:

Risks identified	who is harmed and how?	What are you doing to control risks?	What further action can be taken	Who carries out the action?	When is the action needed to be carried out?	Process
1. Inadequate preparation	Stakeholders, business, organization, government	Security audits	Transitioning to post-quantum cryptography techniques	organization	Medium term plan	Doing
2. Vulnerabilities in current cryptography techniques.	Government, business	Informed about technical advancement	Training and awareness	Government, educational bodies, organization	immediate	Doing
3. Impact on data security	Business Government stakeholders	Adopt quantum safe cryptography techniques	Inform and train for awareness	Educational bodies, Business organization	Medium term	To do
4. Proliferation of quantum technology	stakeholders	Inform about technological advancement	Transition and train to adapt to post-quantum computers	Organization, Educational bodies, regulatory bodies, government	Long term	To do

Need for transition from modern cryptographic techniques to post-quantum cryptographic techniques

1. Shor's algorithm:

Potential of quantum computers to solve problems efficiently that underpins the widely used cryptography techniques which would break RSA and ECC proving them insecure.

2. Proliferation of quantum technology

Rapid advancement of quantum computers suggests that with significant investments from both private and governmental bodies, post-quantum era is closer rather than further.

3. National security concerns

Data from government and military that holds sensitive information regarding the country cannot be left vulnerable to quantum threats and should be prioritized first.

4. Security and confidentiality

Transition is necessary to mitigate risks from store now depict later threats. To meet regulatory standards for protection of sensitive information, transition is necessary.

5. Future proofing

Adoption of cryptography techniques now will render the future security measures to be impenetrable to threats from quantum computers.

6. Lack of preparedness of current systems

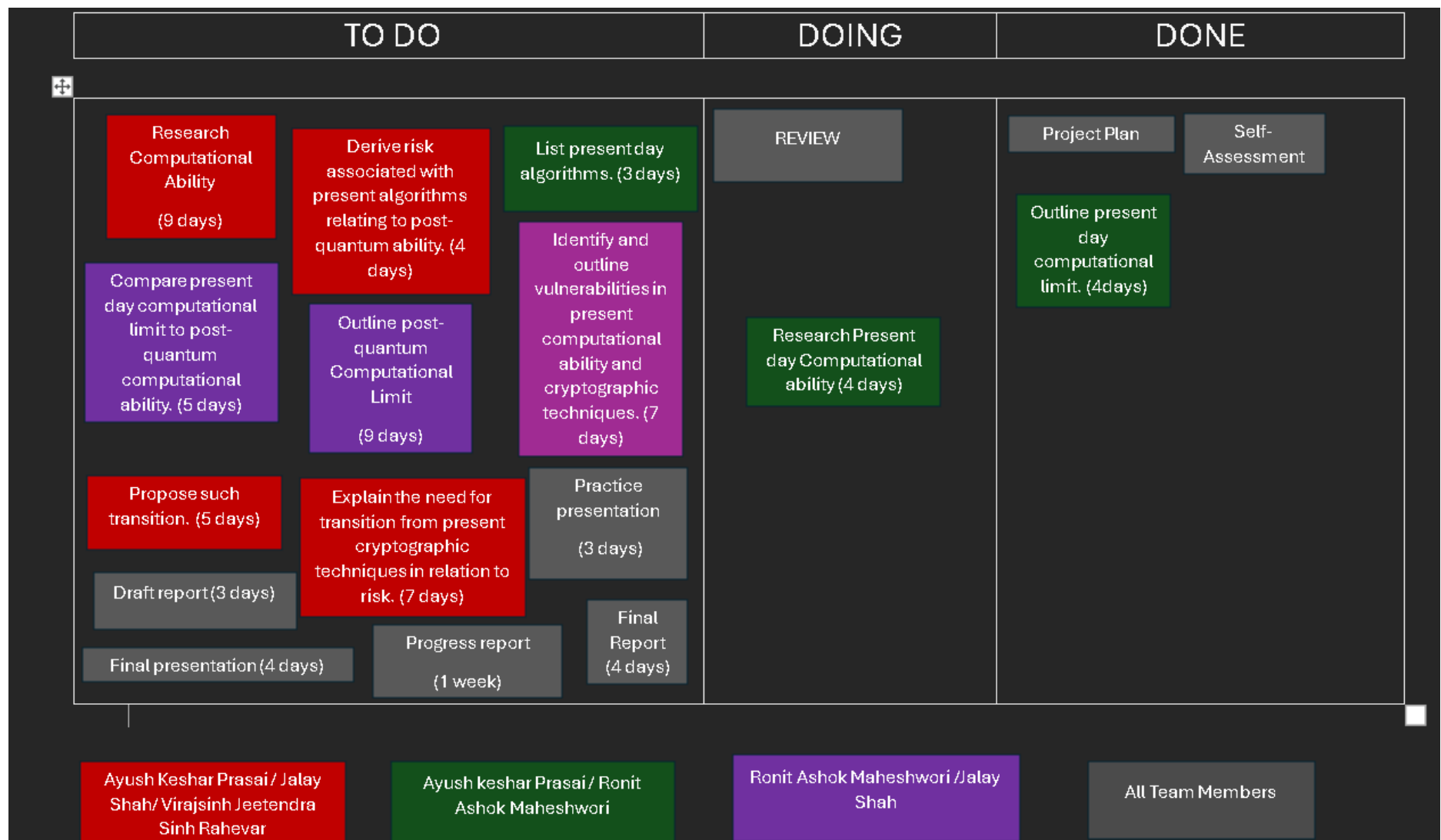
Transition plans are not quite adequate up until now and there is a chance that organizations left behind may have limited time frame to transition that may result in financial stress.

7. Standards development for the community

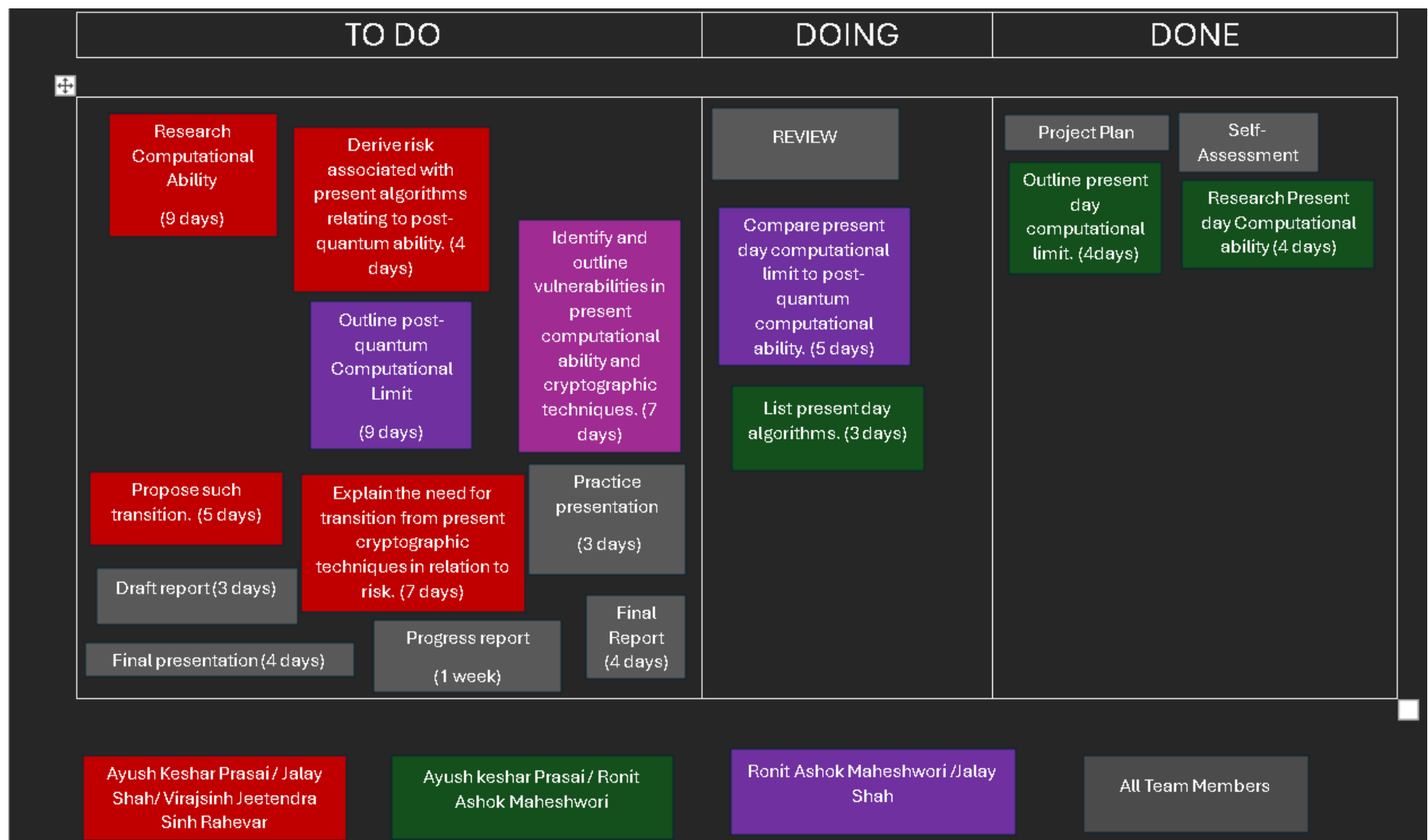
Engaging in such developments as NIST i.e. National Institute of Standards and Technology will ensure that organizations adopt most secure and vetted options.

Progression of the kanban board

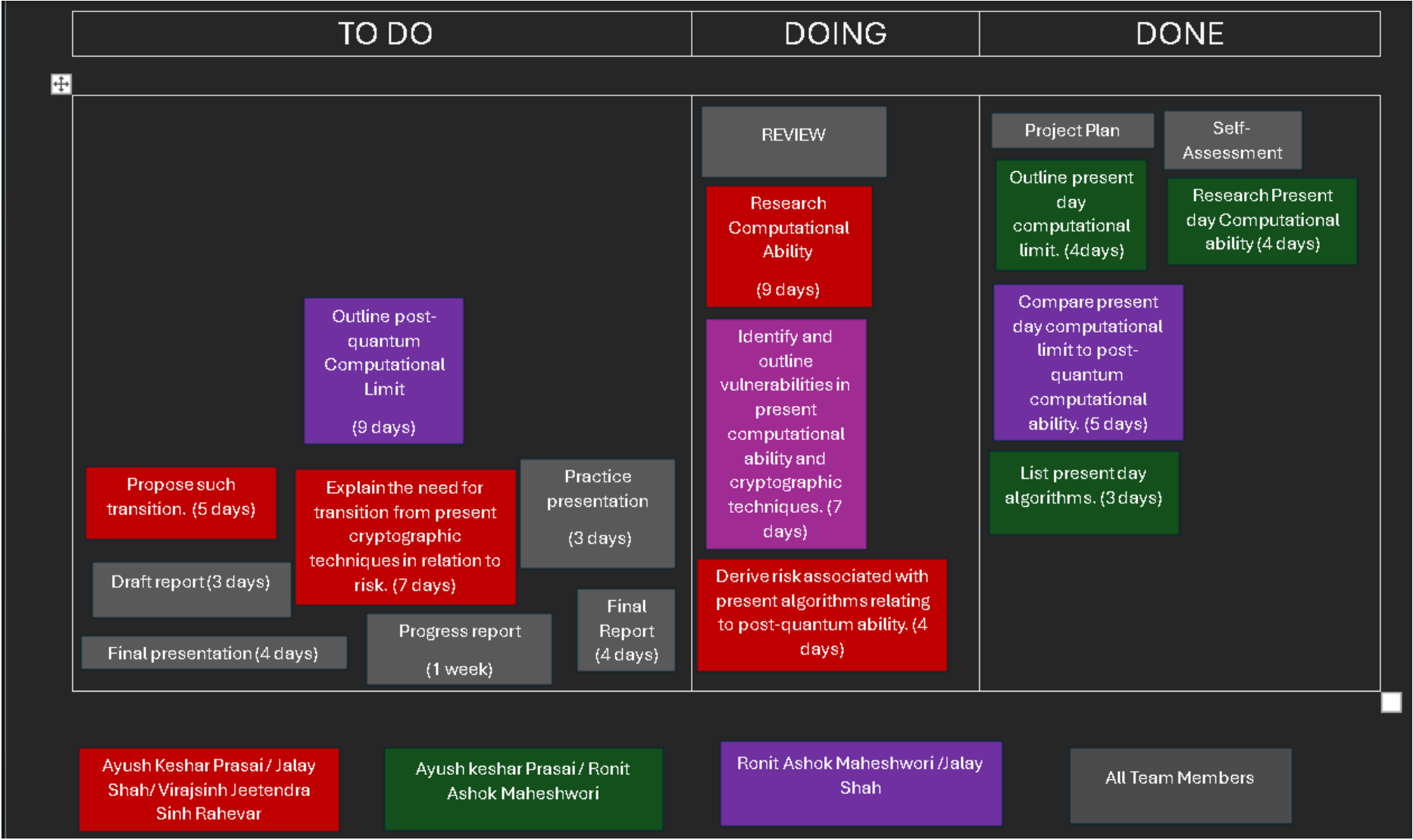
Week 1-3:



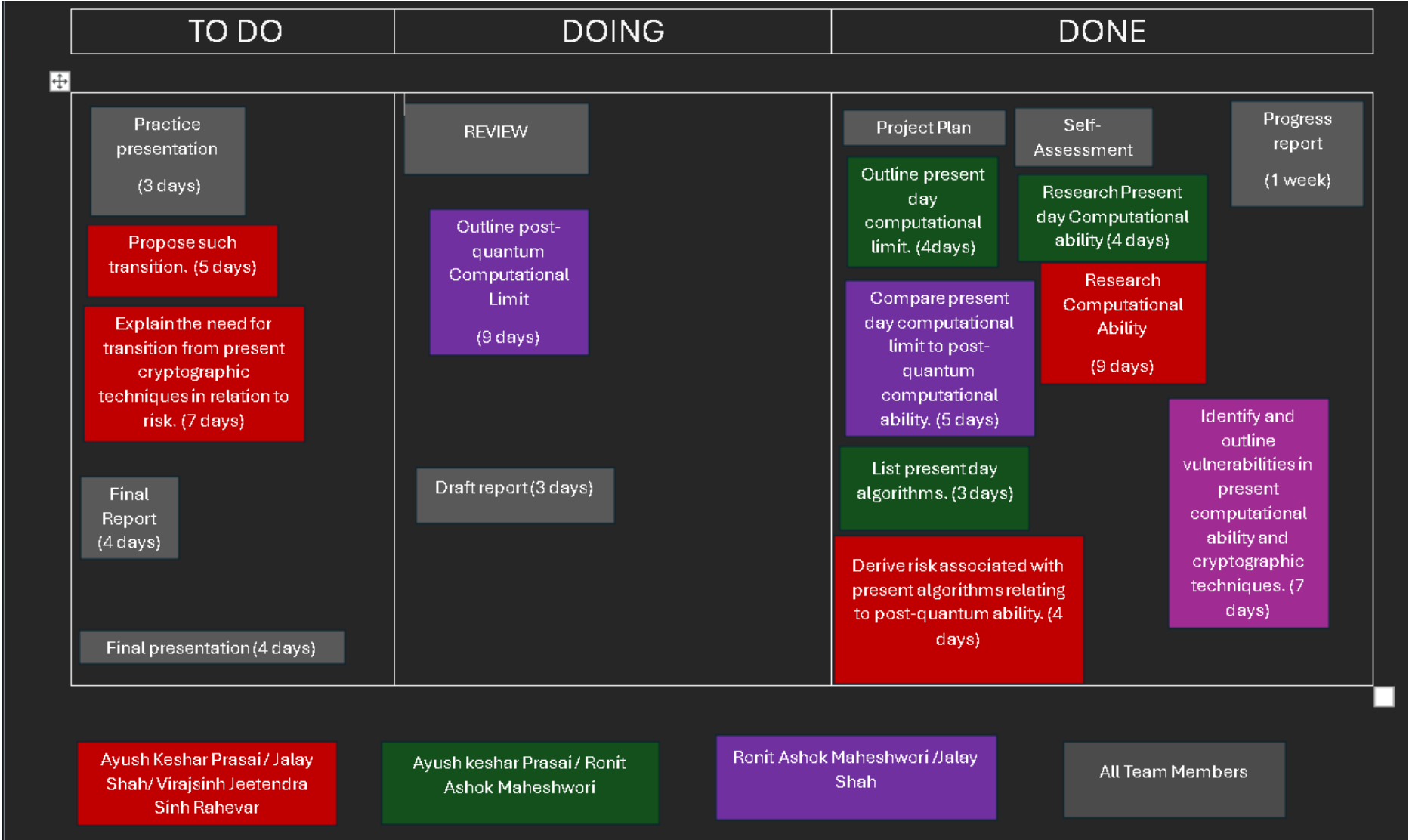
week 4 -5:



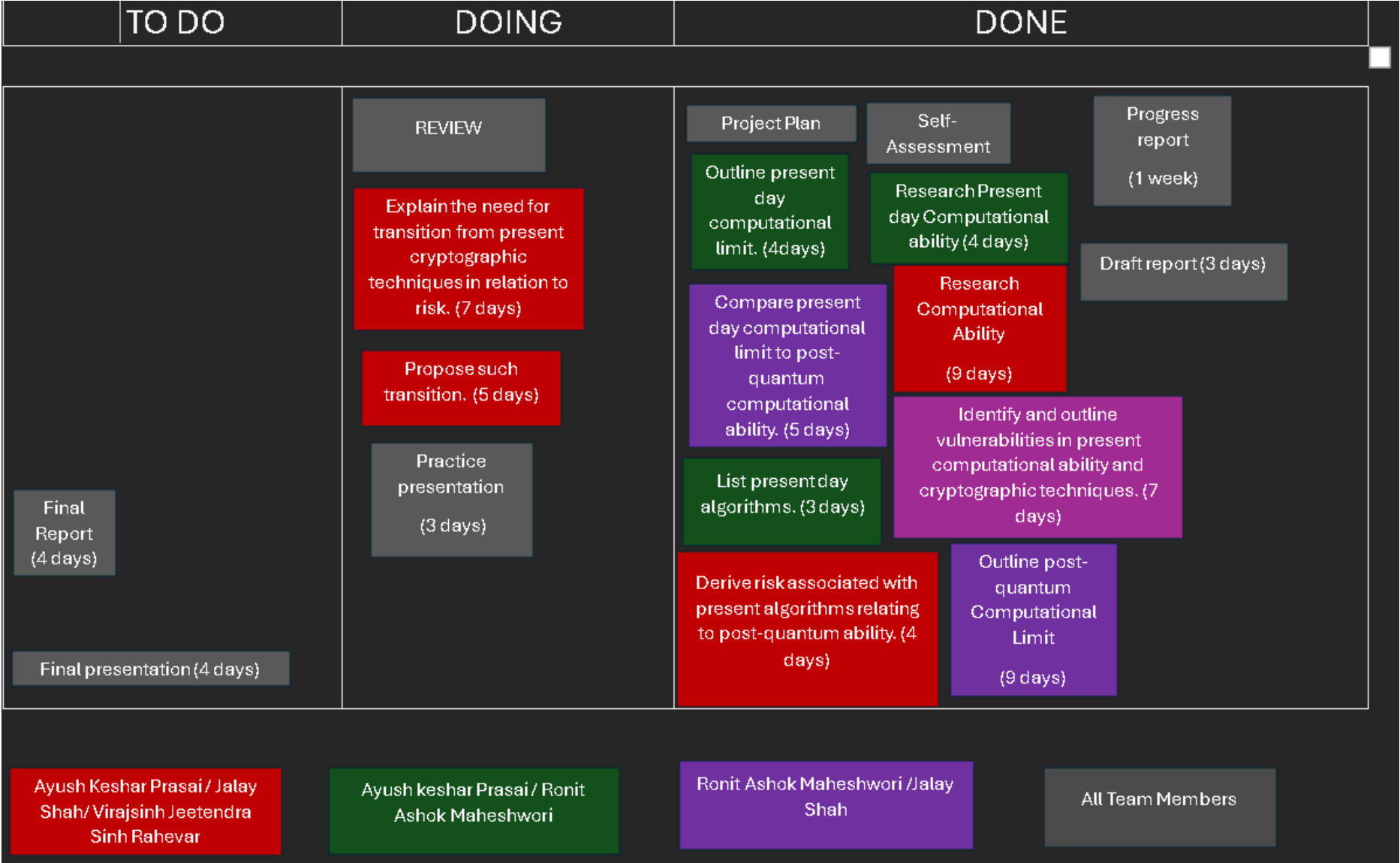
Week 5-6:



Week 7-8:



Week 9-10:



Week 11-12:

TO DO	DOING	DONE	
	<div>REVIEW</div> <div>Final Report (4 days)</div> <div>Final presentation (4 days)</div>	<div>Project Plan</div> <div>Outline present day computational limit. (4days)</div> <div>Compare present day computational limit to post-quantum computational ability. (5 days)</div> <div>List present day algorithms. (3 days)</div> <div>Derive risk associated with present algorithms relating to post-quantum ability. (4 days)</div>	<div>Self-Assessment</div> <div>Research Present day Computational ability (4 days)</div> <div>Research Computational Ability (9 days)</div> <div>Identify and outline vulnerabilities in present computational ability and cryptographic techniques. (7 days)</div> <div>Outline post-quantum Computational Limit (9 days)</div> <div>Progress report (1 week)</div> <div>Draft report(3 days)</div> <div>Propose such transition. (5 days)</div> <div>Practice presentation (3 days)</div> <div>Explain the need for transition from present cryptographic techniques in relation to risk. (7 days)</div>
Ayush Keshar Prasai / Jalay Shah/ Virajsinh Jeetendra Sinh Rahevar	Ayush keshar Prasai/ Ronit Ashok Maheshwori	Ronit Ashok Maheshwori /Jalay Shah	All Team Members

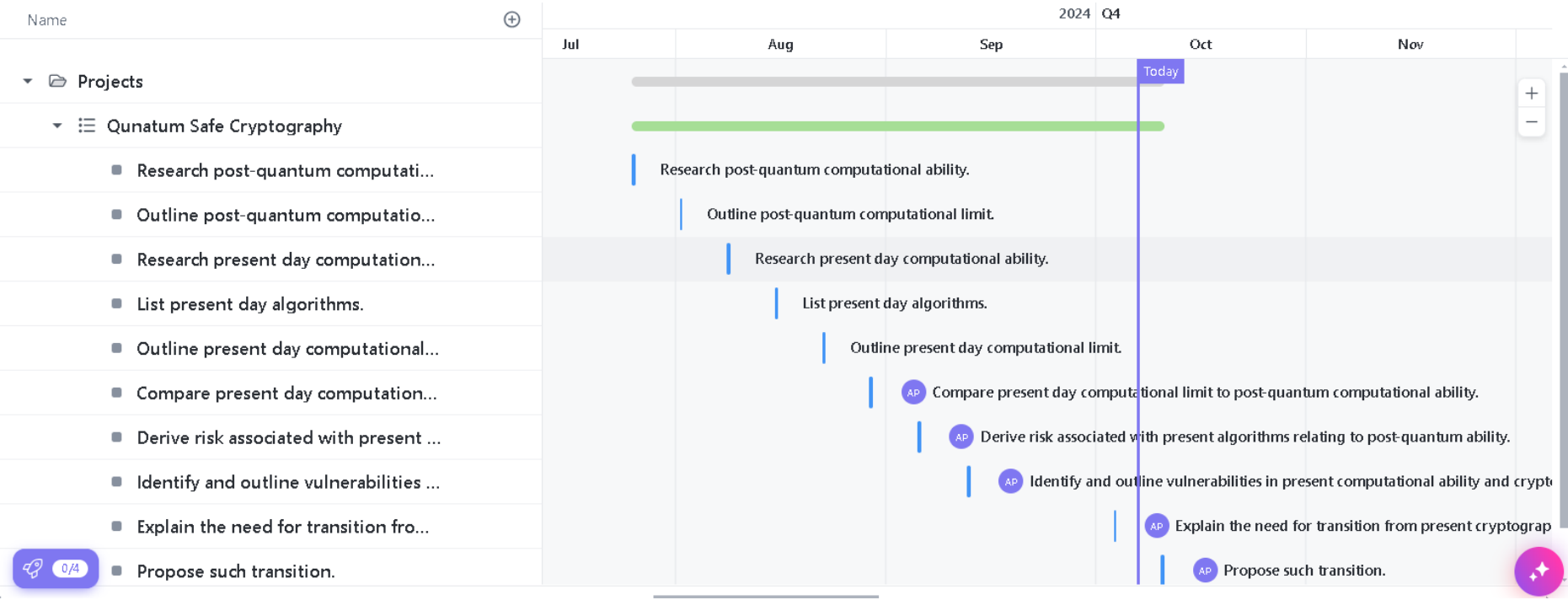
Week 13:

TO DO	DOING	DONE	
		<div>Project Plan</div> <div>Outline present day computational limit. (4days)</div> <div>Compare present day computational limit to post-quantum computational ability. (5 days)</div> <div>List present day algorithms. (3 days)</div> <div>Derive risk associated with present algorithms relating to post-quantum ability. (4 days)</div>	<div>Self-Assessment</div> <div>Research Present day Computational ability (4 days)</div> <div>Research Computational Ability (9 days)</div> <div>Identify and outline vulnerabilities in present computational ability and cryptographic techniques. (7 days)</div> <div>Outline post-quantum Computational Limit (9 days)</div> <div>Progress report (1 week)</div> <div>Draft report(3 days)</div> <div>Propose such transition. (5 days)</div> <div>Practice presentation (3 days)</div> <div>Explain the need for transition from present cryptographic techniques in relation to risk. (7 days)</div> <div>Final presentation (4 days)</div>
Ayush Keshar Prasai / Jalay Shah/ Virajsinh Jeetendra Sinh Rahevar	Ayush keshar Prasai/ Ronit Ashok Maheshwori	Ronit Ashok Maheshwori /Jalay Shah	All Team Members

Gantt chart:

This Gantt chart was made using app clickup which is a free service.

This file cannot be exported and needs an upgrade to a business plan with monthly subscription hence, the photo:



The link to this app is : <https://app.clickup.com/9016558708/v/g/8cpvh3m-176>

Kyber (KEM Algorithm)

1 TABLE OF CONTENTS

Introduction to Kyber.....	1
Designing and Functionality.....	1
Kyber’s Security.....	3
Comparison with other Post-Quantum Algorithms.....	4
Challenges and Future	5
References	5

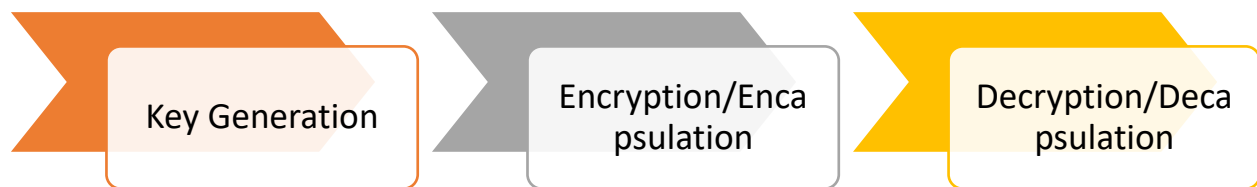
2 INTRODUCTION TO KYBER

Kyber, an algorithm, is basically a post-quantum cryptographic key encapsulation mechanism that has been primarily designed for securing communications when it comes to cyber vulnerabilities in quantum computing (Saoudi et al., 2024). This algorithm has also been defined under NIST framework, including Post-quantum cryptographic (PQC) project in the year 2022 for the purpose of standardizing the algorithm. Basically, the working of this algorithm is based on LWE and RLWE, which are referred to as Learning with Errors and Ring-LWE for resisting attacks from potential computers. The prime reason for using this algorithm for cryptographic key encapsulation mechanism is their efficiency, ease of implementation and even secure data transmission (He et al., 2024). This document delves deeper into Kyber algorithm based on its design, functionality and security related parameters.

3 DESIGNING AND FUNCTIONALITY

The working of Kyber algorithm is primarily based on lattice-based cryptography, which is basically a variant of **Learning with Errors** (LWE) problem. Basically, the hardness of this algorithm is based on mitigating the noisy nature of linear equations and making sure the challenge faced by quantum computers are fixed in an appropriate manner (Saoudi et al., 2024). This algorithm is built on the basis of

three different components such as key generation, encryption/encapsulation and lastly, decryption/de-capsulation. For each component, a detailed discussion have been made below for Kyber algorithm.



1. **Key Generation:** The first component is key generation which is almost same to other algorithms as the key generation is carried out using random polynomials. This generation is primarily done in the finite ring to generate the public and private keys. Once the keys are generated, the use of public key will be done for encryption whilst the private key will be used for decryption, as a normal cryptographic procedure.
2. **Encryption/Encapsulation:** The next component is encryption/encapsulation phase where a random session key is established, which is also said to be the shared key. The use of this key is done for encapsulation and is further transmitted to the recipient's public key. Based on this process, the further encapsulation is performed to the cipher text and finally, it can be shared over insecure channels easily.
3. **Decryption/Decapsulation:** The last one is decryption/decapsulation which is opposite of previous component and is used for decryption. Herein, when the receiver uses the private key, the decapsulation is done on the cipher text so that the original text or a message can be obtained (He et al., 2024).

Also, the working of Kyber algorithm is specified on two types of mathematical foundations such as modular arithmetic and error distribution. In the context of modular arithmetic, the working of this algorithm is on a ring of polynomials by utilizing coefficients under a modular arithmetic. When this has been done, it allows for efficient computation without causing any problem. Furthermore, in the context of error distribution, this algorithm primarily provides a random noise when encryption is being carried out so that when any attack is performed, the original data is not accessible. As a result of this error distribution, it ensures that security is maintained over different types of cyber-attacks. Moreover, the Kyber algorithm is also categorized as Kyber512, Kyber768 and even Kyber1024 levels where the

increasing level will assure higher security (Pathum, 2024). Also, the working of key encapsulation mechanism (KEM) is provided below that shows how the working of Kyber algorithm will be done between a client and a server.

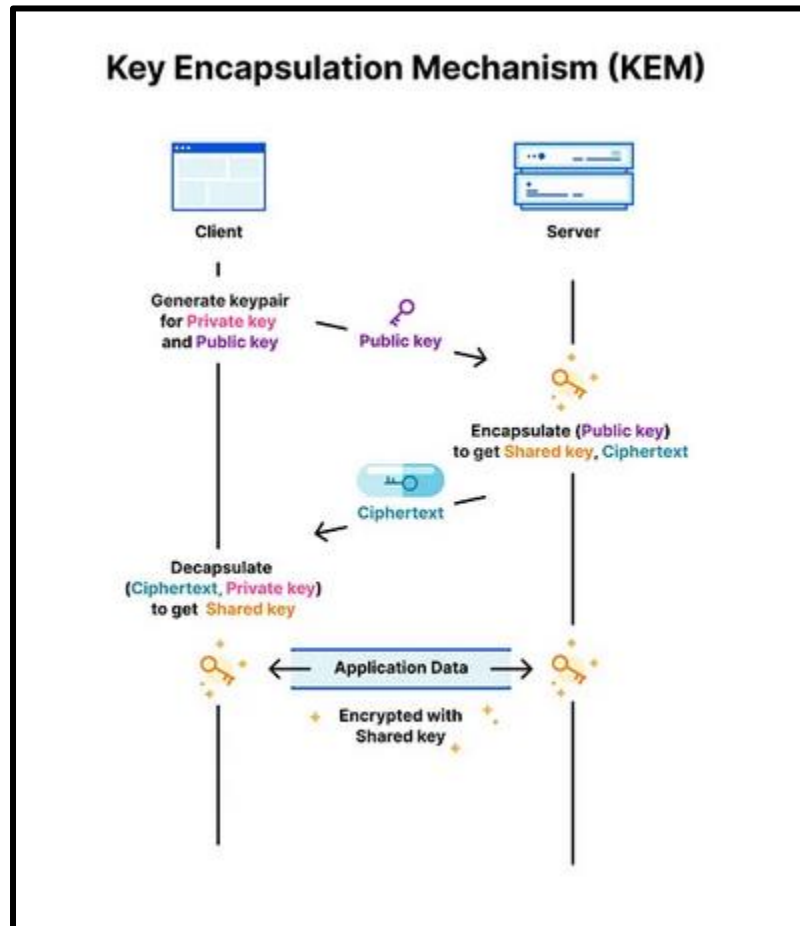


Figure 1: <https://medium.com/@hwupathum/crystals-kyber-the-key-to-post-quantum-encryption-3154b305e7bd>

4 KYBER'S SECURITY

As discussed earlier, the security of Kyber algorithm is based on ring learning with error problem which falls under the lattice cryptographic problem. In terms of security, the Kyber is expected to offer robust working model which will make sure encryption and decryption processes are carried out efficiently. There are various advantages of Kyber when it comes to security, these advantages are explained below.

- The first every advantage is based on post-quantum security where classical key algorithms named as ECC and RSA's reliance on integer factorization and other problems could be solved by

Shor algorithm easily. However, in the case of lattice-based problems, there are no chances for quantum based attacks (Rugo, 2021).

- Kyber is expected to offer provable security as it works under the lattice cryptography which clearly means it is very challenging to break it.
- In the context of chosen cipher text attack resistance (CCA), the selected algorithm offers built-in resistance to different cipher text based attacks. In case a manipulation is performed on the cipher text, there will be no interference to the private key because of message observation (Klas, 2023).
- Finally, for the performance, the designing of the Kyber algorithm is based on balancing a security and efficiency for its optimal working. Once it is done, it will make sure real-world applications such as cloud based services, mobile devices and other areas are protected. In comparison to other post-quantum algorithms, the selected algorithms offers small key sizes with faster key exchange times (Yang et al., 2023).

The tabular matrix provided below shows the three different variants of Kyber algorithm and list its parameters in the context of key sizes, cipher text, and security level.

Variant	Public Key Size	Cipher Text Size	Secret Key Size	Approx./ Security Level
Kyber512	800 bytes	768 bytes	1632 bytes	128-bit, level 1
Kyber768	1184 bytes	1088 bytes	2400 bytes	192-bit, level 3
Kyber1024	1568 bytes	1568 bytes	3168 bytes	256-bit, level 5

5 COMPARISON WITH OTHER POST-QUANTUM ALGORITHMS

In the race for post-quantum cryptography standardization, the Kyber algorithm is not only available algorithm as there are many other present such as NTRU, SABER and FrodoKEM. The table given below discusses why Kyber is better than other algorithms.

Parameters	Kyber	Other Algorithms (NTRU, SABER or FrodoKEM)
Key Sizes	The key sizes for Kyber is small than other algorithms, offering faster speed	The key sizes for these algorithms is larger than Kyber which could utilize more bandwidth

Computation	There is faster computation speed with Kyber as it offers quicker encapsulation and decapsulation times, assuring low latency	In other algorithms, the times are higher and even latency is high (Liang et al., 2024)
Security	The Kyber’s security is underpinned by hardness of lattice based problems (Klas, 2023)	There are no guarantees for security in other algorithms just like Kyber

6 CHALLENGES AND FUTURE

There is no doubt that Kyber is leading in the post-quantum encryption scheme, but it still offers various challenges that have devastating impacts, these challenges are given below.

- The first challenge is based on standardization, even though it has been selected by NIST, but the final finalization is underway.
- Another challenge is related to the adoption as transitioning from the classical cryptographic procedures to the post-quantum cryptographic procedures will take a lot of time, including adoption to the Kyber algorithm (Liang et al., 2024).
- Even there are higher chances that the hybrid approach will be used in future where the classical and post quantum algorithms will be done for proceeding while compatibility issues may be faced.

In terms of future, it can be said that the use of this algorithm will be done in almost entire quantum computing cryptography so that a higher level of security is offered to data. Also, the continuous research is being done which portrays that Kyber is robust against evolving quantum related cyber threats (Liang et al., 2024).

7 REFERENCES

He, S. et al. (2024) 'A lightweight hardware implementation of CRYSTALS-Kyber,' *Journal of Information and Intelligence*, 2(2), pp. 167–176. <https://doi.org/10.1016/j.jiixd.2024.02.004>.

Klas, G. (2023) *CRYSTALS-Kyber for Post Quantum Cryptography demystified*. <https://www.linkedin.com/pulse/crystals-kyber-post-quantum-cryptography-demystified-guenter-klas>.

- Liang, Z. *et al.* (2024) 'Compact and efficient KEMs over NTRU lattices,' *Computer Standards & Interfaces*, 89, p. 103828. <https://doi.org/10.1016/j.csi.2023.103828>.
- Pathum, U. (2024) 'CRYSTALS Kyber : The Key to Post-Quantum Encryption | Medium,' *Medium*, 24 June. <https://medium.com/@hwupathum/crystals-kyber-the-key-to-post-quantum-encryption-3154b305e7bd>.
- Rugo (2021) *Kyber - How does it work? | Approachable Cryptography*. <https://cryptopedia.dev/posts/kyber/>.
- Saoudi, M. *et al.* (2024) 'Low latency FPGA implementation of NTT for Kyber,' *Microprocessors and Microsystems*, 107, p. 105059. <https://doi.org/10.1016/j.micpro.2024.105059>.
- Yang, Y. *et al.* (2023) 'Chosen ciphertext correlation power analysis on Kyber,' *Integration*, 91, pp. 10–22. <https://doi.org/10.1016/j.vlsi.2023.02.012>.