Draft Report

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

Unit: COIT20265

Student 1: Ayush Keshar Prasai

Student 2: Jalay shah

Student 3: Ronit Maheshwori

Student 4: Virajsinh jeetendra sinh rahevar

Project Mentor: Mohammad Mohammad

Date: 26/08/2024

CQUniversity Australia

# Introduction

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.

To understand post-quantum cyptography, let us first dive into cyptography 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 civilization which depends upon standards, algorithms and combination of 0’s and 1’s. it is due to this standard which everyones follows, alogirthims that is calculated and 0’s and 1’s that are intrepeted in specific manner from which confidentiality, integrity and availability is provided to the data over the internet or anywhere else.

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 cyptography 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 cyptography 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 risk to post-quantum cryptography.

# System Overview

The system in process of being designed consists of following parts:

1. Technical description of RSA and justification of why RSA has been considered vulnerable.

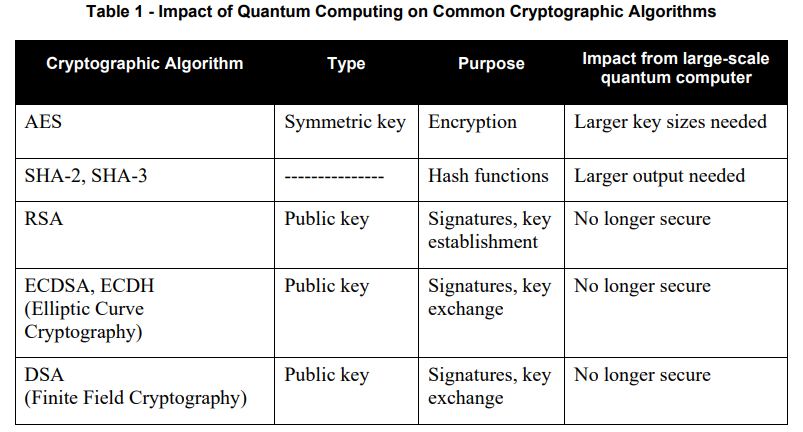


Figure 1: (Chen et al., 2016) impact on algorithms

**Why RSA amongst other asymmetric cyptography algorithms?**

Symetric cryptography:

All symetric cyptography is quantum safe. Symetric algorithms such as SHA and AES do not rely upon mathematical calculations making it quantum safe(Azure, 2024).

Asymetric cryptography

RSA is a 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.

**Vulnerability assessment of RSA:**

vulnerability assessment of RSA is depicted on technical Artefact PG15-Algo-vulnerability.docx where RSA and its vulnerabilities to different post-quantum algorithms are listed in a table.

1. Brute force attack simulating a post-quantum calculation through fermat’s factorization algorithm on RSA to find the private-key from public key.

PG15-fermat-Algo.docx has detailed description along with codes using python and sagemath to showcase the breaking of RSA to figureout the private-key. Post-quantum computational ability is depicted here on a classical computer by limiting the calculation done by choosing the prime number n and e specifically.

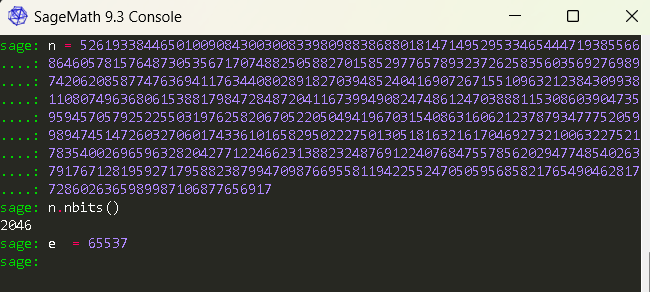


Figure 2: public key pair (e,n)

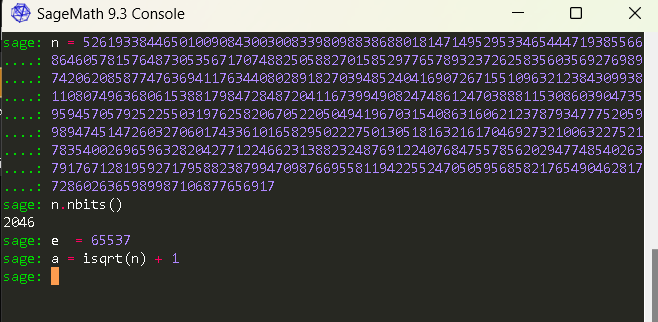


Figure 3: ceiling function a

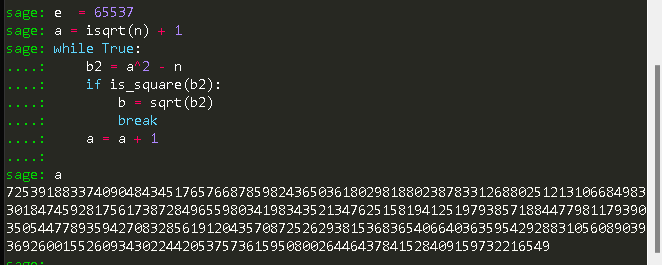


Figure 4: loop that rounds the value of a to the nearest interger until b is found

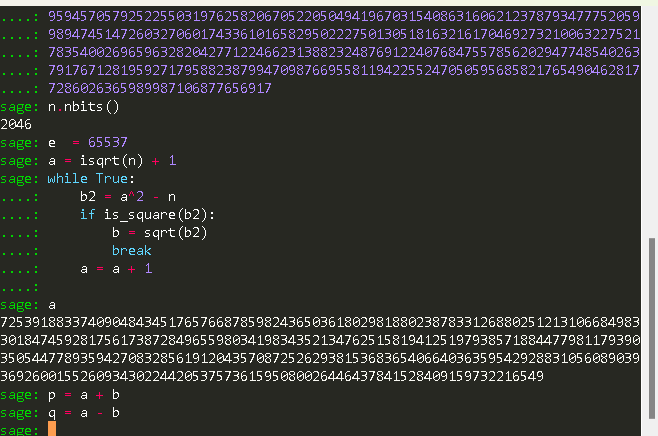


Figure 5: value of p and q where n = p \* q

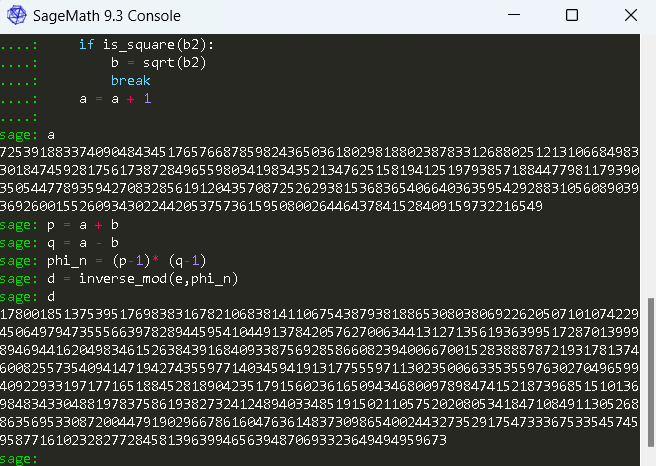
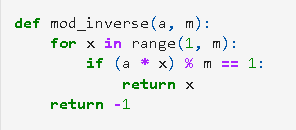


Figure 6: euiler totient to find n and n to find d(private key)

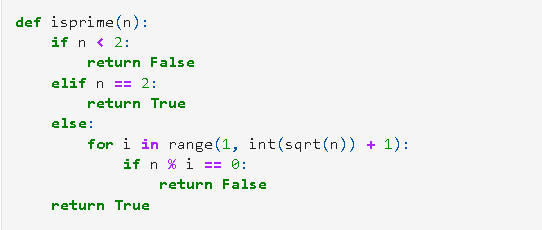
1. Demonstration of use of Qunatum Computing to leverage Shor’s algorithm for cracking RSA Encryption.

Here we demonstrate that RSA encrypted message can be decrypted using Shor’s algorithm (Smaranjitghose, 2020).

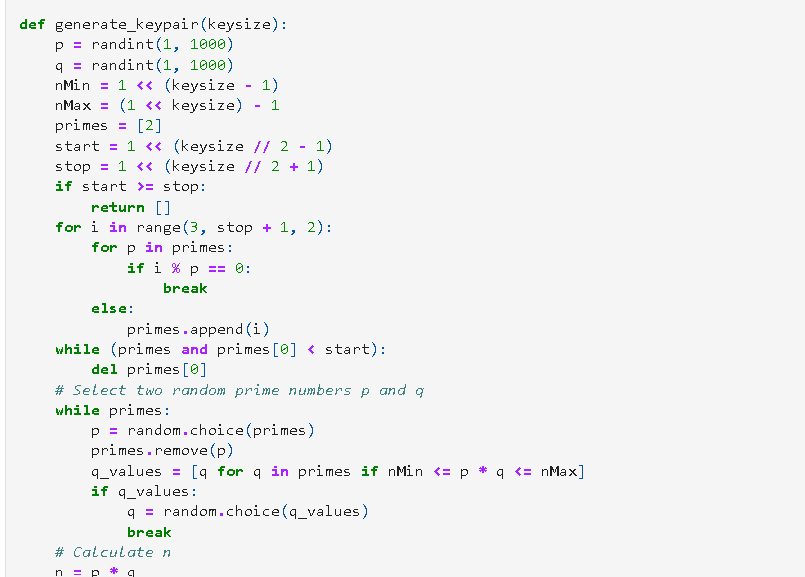
Calculating Modular Inverse:

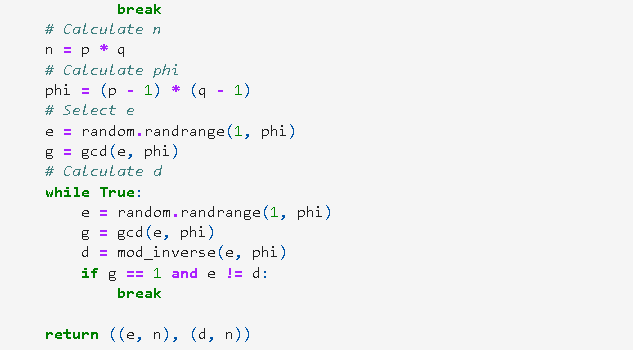


Checking for primality:



Generating Key value pairs:





Encryption step:

A screenshot of a computer

Description automatically generated

Decyption step:

A computer code with text

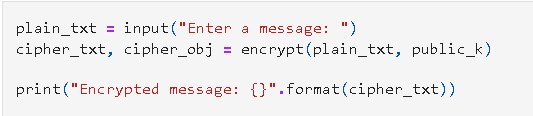
Description automatically generated with medium confidence

**Testing the sample message:**

Generating keys:



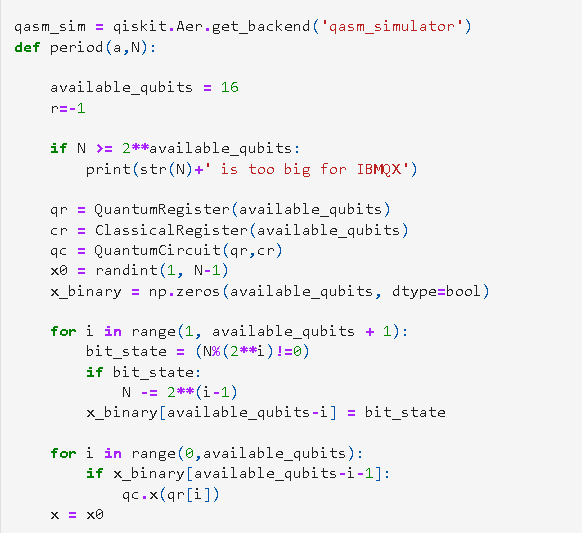
Encryption:

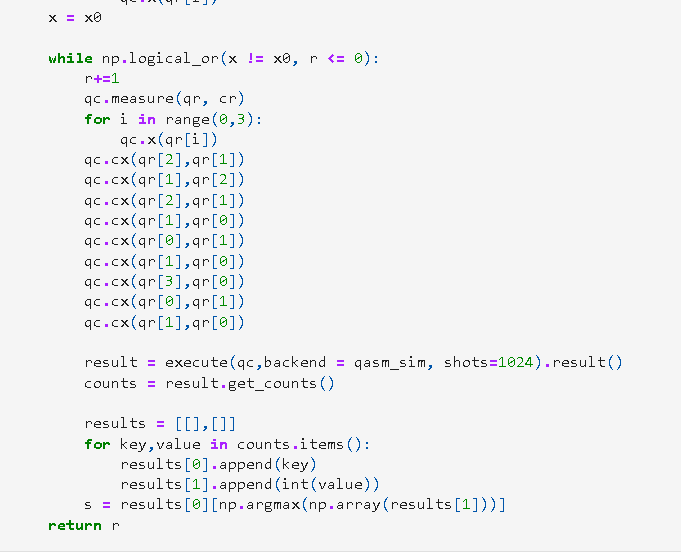


Decryption:



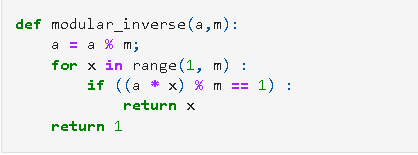
Framing shor’s algorithm:

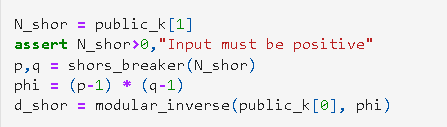




A screenshot of a computer program

Description automatically generated







# Delivered Technical Artefacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **File** | **Description** | **PDF?** |
| Algorithm vulnerability assessment | PG15-Algo-vulnerability.docx | Vulnerability Assessment of cyptography algorithms in use today due to mathematical dependency. | Yes |
| Sagemath Setup Instructions | PG15-Sagemath-setup-instructions.docx | Instructions for deploying the webserver in Azure | No |
| Fermat’s factorization algorithm implemetation | PG15-Fermat-Algo.docx | Detailed description of how a private key can be decrypted from public key pair and factorization of prime number. | Yes |
| Shor’s algorithm decrypting RSA encryption | PG15-Shors-ALGO.ipynb | Source code that demonstartes the shor’s algorith and decryption of RSA encrypted message. | No |

# Contributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Percent** | **Summary of Contributions** | **Technical Lead on Artefacts** |
| Ayush Keshar Prasai | 25% | Researched, tested and documented vulnerabilities from the shor’s and fermat’s algorithm on RSA. | * Fermat’s factorization algorithm implemetation * Shor’s algorithm decrypting RSA encryption * Algorithm vulnerability assessment |
| Jalay Shah | 25% | Researched and proposed algorithms vulnerable to quantum computing. | * Algorithm vulnerability assessment |
| Ronit Maheshwori | 25% | Set up environment to test shor’s and grovers algorithm | * Sagemath setup * Proposed python |
| Virajsinh Jeetendra Sinh Rahevar | 25% | Tested libraries, tools and methodologies essential for building the system. | * Qiskit , numpy, jupiter, python. * Sagemath setup |

# Next steps

Not yet completed

# References

Azure, quantum (2024) *Microsoft*, *Azure Quantum*. Available at: https://quantum.microsoft.com/en-us/our-story/quantum-cryptography-overview#:~:text=Most%20commonly%20used%20public%20key,and%20other%20standardization%20bodies%20globally. (Accessed: 26 August 2024).

Chen, L. *et al.* (2016) *Report on post-quantum cryptography*, *CSRC*. Available at: https://csrc.nist.gov/pubs/ir/8105/final (Accessed: 26 August 2024).

Smaranjitghose (2020) *QUANTUM\_BURGLARY/REQUIREMENTS.TXT at master · smaranjitghose/quantum\_burglary*, *GitHub*. Available at: https://github.com/smaranjitghose/quantum\_burglary/blob/master/requirements.txt (Accessed: 26 August 2024).