Quantum Computing Project Report 2

PL3001 - Quantum Computing

Eshwar SK, Tanya Sravan

Project Report 2

23^{rd} April 2024

Contents

1 What is Post Quantum Cryptography

Post quantum cryptography(PQC) is a response to the potential threat posed by the emergence of a large fault free quantum computer with the ability to run algorithms like Shor's. The main aim of PQC is to protect classical systems from attacks using mathematically hard problems to create classical encryption keys.

2 Why is PQC Relevant

Traditional cryptography relies on the difficulty of problems like integer factorization (IF) and unstructured search. However, quantum algorithms exploit quantum mechanics to achieve significant speedups, posing a threat to existing cryptosystems. Two such algorithms are the Shor's algorithm and the Grover's algorithm.

Grover's algorithm tackles unstructured search, aiming to find a specific element ("marked state") within an unsorted database. Classically, this requires checking each element (bit string) with a time complexity of $O(N)$.

Grover's power lies in utilizing superposition. It prepares a quantum register of n qubits in a uniform superposition state:

Grover's Algorithm

State	Representation $\|\rangle$
Superposition	$(\|0\rangle + \|1\rangle) \otimes n$

Here, n represents the number of qubits in the quantum register, and $\otimes$ denotes the tensor product representing the superposition across all qubits.

This state represents all possible bit strings simultaneously. Subsequently, Grover applies a sequence of rotations (Oracle Operator and Grover Diffusion Operator) that amplifies the probability of the marked state while suppressing others.

Oracle Operator (O): This operator differentiates the marked state. Mathematically, for an input state $|x\rangle$:

$$O|x\rangle = \begin{cases} |x\rangle & \text{if } f(x) = 1 \text{ (marked)} \\ -|x\rangle & \text{otherwise} \end{cases}$$

where $f(x)$ is the function used to identify the marked element.

Grover Diffusion Operator (D): This operator reflects the state around the average, amplifying the marked state. It involves the Hadamard transform (H) and a reflection about the uniform superposition:

$$D = 2|\psi\rangle\langle\psi| - I$$

where:

- $|\psi\rangle$ is the current state of the quantum register.
- I is the identity operator.

Shor's algorithm tackles integer factorization, a critical problem for public-key cryptography (RSA). Factoring large numbers classically is computationally expensive, but Shor's algorithm achieves polynomial time complexity (roughly $O(log^3 N)$).

Shor's algorithm leverages superposition to create a period finding subroutine. It starts with an integer N and a smaller integer a coprime to N. It then puts a superposition of all possible values (0 to $N-1$) in a quantum register and applies a modular exponentiation operation involving a and N.

The resulting state encodes the period (p) with which a^p repeats modulo N. This period finding is achieved using the Quantum Fourier Transform (QFT), which exploits the relationship between addition and multiplication modulo N.

This algorithm utilizes the QFT to find the period in the modular exponentiation. The QFT operates on a superposition of states and relates addition and multiplication modulo N. Here's a simplified representation (the actual QFT involves complex number manipulations):

$$\text{QFT}\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\right) = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle \exp\left(\frac{2\pi i j k}{N}\right)$$

Here:

- N is the integer to be factored.
- k represents an index iterating through possible values (0 to $N-1$).

By finding the period (p), Shor's algorithm can efficiently compute the factors of N. This poses a significant threat to RSA encryption, as breaking the public key (based on large factorized numbers) allows decryption of messages.

More details on Shor's and Grover's are available in the notes in the git hub repository

3 Lattice Based Cryptography

A lattice can be defined as the linear combination of n independent vectors, $L = z_1 b_1 + z_2 b_2$. What makes lattice problems hard is that it is modeled after the closest vector problem. Suppose we have a lattice and a point. The hard problem is finding a point on the lattice that is the closest point to the given point.

When the vectors b_1 and b_2 have an angle of around 90^o in between them then finding the closest point is not hard however when the angle between the vectors are close to 0 then finding the closest point is hard. In lattice based cryptography the primary method of encryption is learning with errors (LWE). There are 3 different systems of LWE, the base model LWE, then the one with further optimization Ring-LWE and the final one optimized for increasing security without changing the dimension size of the matrices, Module LWE.

The explanation of why LWE is hard along with a more descriptive explanation of the 3 systems along with their complexities is uploaded in the git under LWE further explanation. The ploynomial foundations and operations associated with LBC's will also be found in the git repo under polynomial arithmetic

4 Crystal-Kyber

Kyber is a lattice-based cryptosystem, specifically inspired by the Learning With Errors (LWE) problem in module lattices (MLWE). One of its primary design purposes is to establish a shared secret between two communicating parties (which then continue using symmetric cryptography) in a secure way without an attacker being able to decrypt it (and thus future communications between the parties).

Kyber offers three sets of parameters, named Kyber512, Kyber768, and Kyber1024, with the aim of achieving 128, 192, and 256 bit security levels, respectively. An important distinction between Kyber's design and other Ring-LWE encryption schemes is that these schemes need to change the lattice dimensions (n), and modulus (q), to increase the security levels, however Kyber fixes these values and achieves a higher security by increasing a scaling parameter, k, to in-turn attain a higher lattice dimension. This means operations within Kyber, such as multiplication, can all be done using the same fundamental operations, using the same parameters, except that more repetitions of these same operations are required to achieve a higher level of security.

The hardness of this algorithm comes from the fact that t is not simply a matrix product but also offset with a random noise vector e. figuring out the value of the secret key s by just looking at (A, t) is not trivial. (If there were no noise offset, one can use row reduction and Gaussian elimination tricks to solve for s.).

More specifically, solving for s from (A, t) can be reduced to solving an instance of the problem of Module Learning With Errors (M-LWE), which is believed to be hard even for quantum computers (similar proof as why is LWE hard shown above). Kyber gains quantum-resistance - by linking its core underlying public-key scheme with a mathematically hard problem (M-LWE) that cannot be efficiently solved by a quantum computer.

The Encryption and Decryption methodology is uploaded on the git repo under LWE encryption method. A code implementation is also provided with analysis of the code on the git repo under Kyber implementation.

5 Code Based Cryptography

Code based cryptography is heavily based on the concept of error correction codes. Digital media is often exposed to memory corruption and so we use error correction methods where we store k bits in n bits, meaning the redundancy is $n - k$. Checksums are a basic illustration of an error-detection code. The objective is to maximize the likelihood of data transmission accuracy while decreasing the volume of additional information added. If good codes are used then one can correct multiple errors.

The key concept of Code based cryptography is the use of linear codes, where we have a binary code, c, with length n and dimension k. The core concepts of Code based cryptography revolves around the generator matrix. When we apply the generator matrix to the code we are able to increase the distance between the valid codes making it easier to correct in incase there is an error. The parity check matrix then tells us on the receiving end whether or not there is an error and tells us what errors to fix.

Further notes available on the git repo under CBC notes

6 Classic McEliece Cryptography

The McEliece cryptosystem is based on the concept of error-correcting codes. The McEliece system offers a unique approach to encryption by leveraging the difficulty of decoding in linear binary codes (further proof to this is provided in the git repo under CBC efficiency proof). McEliece offers a robust and practical approach to encryption. Its security is based on well-established mathematical principles, and it has withstood decades of cryptanalysis without significant vulnerabilities being discovered. This system is also versatile and adaptable to various settings and security requirements. It can be used for secure communications in a wide range of applications, from secure messaging to protecting critical infrastructure.

The Encryption and Decryption methodology is uploaded on the git repo under McEliece encryption method.

7 Future Scope

First we need to further improve on the Kyber code. This involves:

- Scale up the parameters such that it achieves post-quantum security.
- Try to make the polynomial operations more efficient
- Understand potential vulnerabilities

Then we need to explore algorithms in the multivariate cryptography field

- Final goal is to compare the different fields of cryptography and analyze the benefits to each of them
- We seek to propose and explain the various PQC solutions, suggest some improvements to them and demonstrate a limited practical implementation for various use cases.