International Journal of
Mathematics And its Applications
ISSN: 2347-1557

Int. J. Math. And Appl., **11(4)**(2023), 111–125
Available Online: http://ijmaa.in

Group Theory in Lattice-Based Cryptography

Michael N. John[1,*], Otobong G. Udoaka[1], Itoro U. Udoakpan[2]

[1]*Department of Mathematics, Akwa Ibom State University, Mkpat Enin, Nigeria*

[2]*Department of Mathematics, University of Port Harcourt, East/West Road, Choba, Rivers State, Nigeria*

Abstract

Group theory plays a fundamental role in lattice-based cryptography, providing a rich mathematical framework for the design and analysis of cryptographic protocols. This paper explores the application of group theory concepts within lattice-based cryptographic systems, focusing on the algebraic structures formed by lattices and their subgroups. The utilization of group theory in lattice-based cryptography enhances the security and efficiency of key exchange, encryption, and digital signatures. Through a mathematical lens, we investigate the foundational principles, theorems, and cryptographic constructions that leverage group theory, shedding light on the symbiotic relationship between group theory and lattice-based cryptography. The paper also proposes cryptographic scheme based on group theory in lattice-based.

Keywords: Group theory; lattice-based cryptography; algebraic structures; key exchange; encryption; digital signatures; subgroup; lattice problems; Ring-LWE; homomorphism; cryptographic protocols.

1. Introduction

Group theory plays a significant role in lattice-based cryptography, a cryptographic paradigm that relies on the hardness of certain problems associated with mathematical structures called lattices.[20] have worked on algebraic method for cryptography. Lattice-based cryptography, emerging as a promising paradigm for secure communication, relies heavily on the principles of group theory. Read the work of [18] on cube-based lattice cryptography. This introduction lays the groundwork by elucidating the fundamental concepts of group theory and their application in lattice-based cryptographic protocols. See [5] for polycyclic group-based cryptography. We explore how lattice structures, defined by basis vectors, form additive abelian groups, providing a versatile platform for cryptographic operations. Lattice-based cryptography is known for its resistance to attacks by

*Corresponding author (storm4help1@gmail.com)

quantum computers, making it a promising candidate for post-quantum cryptography [12]. Here's how group theory is employed in lattice-based cryptography:

1.1 Lattice as an Additive Abelian Group

Lattices, in the context of lattice-based cryptography, are typically treated as additive abelian groups. The set of points in a lattice, along with the vector addition operation, forms an abelian group.

Proposition 1.1. *The set of lattice points forms an abelian group under vector addition. A lattice is a set of points in n-dimensional space with certain properties:*

(1) Closure under addition: For any two points v, w in the lattice, their sum $v + w$ is also in the lattice.

(2) Associativity of addition: Addition in the lattice is associative.

(3) Identity element: There exists a unique point 0 in the lattice such that $v + 0 = v$ for all v in the lattice.

(4) Inverse elements: For every point v in the lattice, there exists a unique point $-v$ in the lattice such that $v + (-v) = 0$.

(5) Commutativity of addition: Addition in the lattice is commutative, meaning $v + w = w + v$ for all v and w in the lattice.

Theorem 1.2. *Given a lattice L with basis vectors $b_1, b_2, \dots, b_n$, the set $\{b_1, b_2, \dots, b_n\}$ generates L as an abelian group.*

Proof. Every vector v in L is a linear combination of the basis vectors $b_1, b_2, \dots, b_n$ with integer coefficients. This is because L is defined as $\left\{\sum_{i=1}^{n} x_i b_i : x_i \in Z\right\}$. Therefore, the set $\{b_1, b_2, \dots, b_n\}$ generates every vector in L under vector addition. Vector addition is inherently associative. The zero vector $\mathbf{0}$ is in L, and it can be expressed as the trivial linear combination $\sum_{i=1}^{n} 0.b_i$ Therefore, the set $\{b_1, b_2, \dots, b_n\}$ contains the additive identity. For any vector v in L, its additive inverse $-v$ is also in L and can be expressed as $\sum_{i=1}^{n} -1.b_i$ Therefore, the set $\{b_1, b_2, \dots, b_n\}$ contains additive inverses. Vector addition in L is commutative. □

Example 1.3. *Consider the set of points in the plane with integer coordinates: $L = \{(x, y) \mid x, y \in Z\}$. Let's verify the properties of a lattice for this example:*

- ***Closure under addition:*** *If (a, b) and (c, d) are in L, then $(a + c, b + d)$ is also in L since $a + c$ and $b + d$ are integers.*
- ***Associativity of addition:*** *Addition in the plane is associative.*
- ***Identity element:*** *The point $(0, 0)$ is the identity element since $(x, y) + (0, 0) = (x, y)$ for all (x, y) in L.*
- ***Inverse elements:*** *For any point (x, y) in L, its inverse is $(-x, -y)$ since $(x, y) + (-x, -y) = (0, 0)$.*

- ***Commutativity of addition:*** *Addition of points in the plane is commutative.*

Proposition 1.4. *Given a lattice L and a subgroup H, the left cosets of H in L partition L into distinct sets of equal size.*

Theorem 1.5. *Given a lattice L and a subgroup H, L can be decomposed into a disjoint union of left cosets.*

Proof. By definition, the left cosets of H in L are sets of the form $aH = \{a + h : h \in H\}$ for all $a \in L$. Each element $x \in L$ belongs to some left coset aH for a unique $a \in L$. $L = \bigcup_{a \in L} aH$ The left cosets are pairwise disjoint, meaning that no element is in more than one left coset. This ensures distinct sets in the partition. $aH \cap bH = \emptyset$ for $a, b \in L$ with $a \neq b$. Each left coset aH has the same size as H. $|aH| = |H|$ for all $a \in L$. The properties of closure, associativity, identity, inverses, and commutativity can be demonstrated by applying the basic rules of addition and the fact that the sum or difference of integers is still an integer. The set of integer points in the plane forms a lattice under addition, satisfying the properties of an additive abelian group. The extension of this concept to higher dimensions and more complex lattices follows similar principles. □

1.2 Module Structure

Lattices often have a module structure, and modules are mathematical structures that generalize the notion of vector spaces. This module structure is crucial in formulating problems that are hard to solve, forming the basis for cryptographic primitives. Get the basis of this, by reading the work on [15]. Lattice-based cryptography often relies on the modular structure of lattices. Modular operations within lattices play a crucial role in the design of cryptographic primitives such as encryption and decryption. Let's explore the modular structure of lattices and how it can be applied to lattice-based cryptography, providing an example with encryption and decryption.

Definition 1.6 (Lattice)**.** *A lattice in Z^n is defined as $L = \{v = Ax | x \in Z^n\}$, where A is an $n \times n$ matrix with linearly independent columns.*

Definition 1.7 (Modular Structure)**.** *Points in the lattice can be taken modulo a certain value. This introduces a modular structure, making computations more manageable.*

1.3 Encryption and Decryption with Modular Lattices

Now, let's consider how lattice-based cryptography utilizes the modular structure for encryption and decryption. The Learning With Errors (LWE) problem is often a basis for lattice-based cryptography.

Definition 1.8 (Encryption)**.**

- *Given a lattice L and a public key $\boldsymbol{p}$ representing a point in L, encryption involves adding a small error term $\boldsymbol{e}$ to $\boldsymbol{p}$ and sending the result $c = p + e$ as the ciphertext.*

- *Mathematically,* $c = p + e \mod q$*, where* q *is a chosen modulus.*

Definition 1.9 (Decryption)**.**

- *Decryption involves finding the closest lattice point to the received ciphertext* **c** *and then subtracting the error term to recover the original point* ***p***.
- *Mathematically,* $p = Round(c \mod q) - e$*, where* $Round(\cdot)$ *represents the rounding operation.*

Example 1.10. *Let's consider a simple two-dimensional lattice defined by* $L = \{(a, b)|a, b \in Z\}$ *and a public key* $p = (3, 4)$*. The error term is* $e = (1, -2)$ *and the modulus is* $q = 7$*.*

1. ***Encryption:*** $C = p + e \mod q = (3, 4) + (1, -2) \mod 7 = (4, 2)$.

   Encryption Matrix Table: *Consider the vertices labeled A, B, C, D, E, F, G, H corresponding to the ordered pairs (1,0), (0,1), (1,1), (0,0), (1,0), (0,1), (1,1), (0,0) respectively. Let the encryption matrix M be;*

$$M = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$$

   The encryption process involves multiplying the matrix M with the ordered pair representing each vertex.

Vertex Label	Ordered Pair (Original)	Encryption Matrix (M)	Encrypted Ordered Pair	Vertex Label
A	(1, 0)	$M\cdot$(1,0)	(2,3)(2,3)	C
B	(0, 1)	$M\cdot$(0,1)	(1,2)(1,2)	B
C	(1, 1)	$M\cdot$(1,1)	(5,8)(5,8)	H
D	(0, 0)	$M\cdot$(0,0)	(0,0)(0,0)	D
E	(1, 0)	$M\cdot$(1,0)	(2,3)(2,3)	C
F	(0, 1)	$M\cdot$(0,1)	(1,2)(1,2)	B
G	(1, 1)	$M\cdot$(1,1)	(5,8)(5,8)	H
H	(0, 0)	$M\cdot$(0,0)	(0,0)(0,0)	D

   In this example, the encryption matrix M transforms the ordered pairs representing the vertices into permuted ciphertexts. The resulting encrypted ordered pairs are then associated with the corresponding vertex labels. This is a simplified illustration, and in practical lattice-based cryptography, more sophisticated mathematical operations and parameters is involved.

2. ***Decryption:*** $P = Round((4, 2) \mod 7) - e = (4, 2) - (1, -2) = (3, 4)$.

   Decryption Matrix Table: *Let the inverse of the encryption matrix M be denoted as* M^{-1}*:*

$$M^{-1} = \begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix}$$

   The decryption process involves multiplying the inverse matrix M^{-1} *with the permuted ciphertext.*

Vertex Label	Encrypted Ordered Pair	Decryption Matrix (M^{-1})	Decrypted Ordered Pair	Vertex Label (Recovered)
C	(2,3)(2,3)	M^{-1}·(2,3)	(1,0)(1,0)	A
B	(1,2)(1,2)	M^{-1}·(1,2)	(0,1)(0,1)	B
H	(5,8)(5,8)	M^{-1}·(5,8)	(1,1)(1,1)	C
D	(0,0)(0,0)	M^{-1}·(0,0)	(0,0)(0,0)	D
C	(2,3)(2,3)	M^{-1}·(2,3)	(1,0)(1,0)	A
B	(1,2)(1,2)	M^{-1}·(1,2)	(0,1)(0,1)	B
H	(5,8)(5,8)	M^{-1}·(5,8)	(1,1)(1,1)	C
D	(0,0)(0,0)	M^{-1}·(0,0)	(0,0)(0,0)	D

In this example also, the decryption matrix M^{-1} is used to reverse the encryption, recovering the original ordered pairs associated with the vertices. The vertex labels are then recovered based on the decrypted ordered pairs. This is a simplified illustration, and in practical lattice-based cryptography, more sophisticated mathematical operations and parameters is involved. See [14] to study on Homomorphic encryption.

1.4 Trapdoor Functions

Certain lattice problems, such as the Learning With Errors (LWE) problem and the Ring-LWE problem, involve finding a "trapdoor" in the lattice. Group theory concepts come into play in the formulation of these problems and in proving their hardness. The LWE problem involves finding a short lattice vector in the presence of random noise. The problem is often described using a system of linear equations. Suppose we have a matrix A and vectors s (the secret) and e (noise) such that:

$$A \cdot s + e = b \mod q$$

- A is a random matrix.
- s is the secret vector we want to recover.
- e is a noise vector.
- b is the observed vector.
- q is a modulus.

This equation represents a system of linear equations with noise. Solving this problem is computationally hard, and its hardness is related to the mathematical properties of lattices.

Connection to Group Theory:

1. **Lattice as an Abelian Group:** The set of lattice points forms an abelian group under vector addition. The LWE problem involves finding a short lattice vector, which can be considered an element of this group.

2. **Modular Arithmetic and Group Structure:** The modulo operation (mod q) introduces a group structure. In the LWE problem, the equations are considered modulo q, emphasizing a group-theoretic aspect.

3. **Hardness Assumption:** The hardness of the LWE problem is based on the assumed difficulty of finding the secret vector s given the observed vector b. This difficulty is related to the structure and properties of the lattice, which is essentially a group.

Proof. **Claim:** Solving the LWE problem is at least as hard as solving certain problems related to lattice-based groups.
Assume there exists an efficient algorithm A^I that solves the LWE problem, i.e., it efficiently recovers the secret vector s from the observed vector b. We can use A^I to solve a related group problem. Given an element g in the lattice-based group, we can construct an LWE instance where $b = g$ and attempt to recover s. If A^I succeeds, we have solved the group problem. □

1.5 Ring Structure

In some lattice-based cryptographic schemes, rings and ring structures are utilized. You should read [3] and [11] extensively. The algebraic structure of rings is leveraged in the construction of cryptographic protocols, including key exchange and digital signatures. A ring is an algebraic structure consisting of a set equipped with two binary operations: addition and multiplication. For a set R to form a ring, it must satisfy the following properties:

1. **Additive Abelian Group:**
   - The set R with addition forms an abelian group.
   - Closure under addition.
   - Associativity of addition.
   - Existence of an additive identity.
   - Existence of additive inverses.
   - Commutativity of addition.

2. **Multiplicative Closure:**
   - Multiplication is closed in R.

3. **Distributive Laws:**
   - Left and right distributive laws between addition and multiplication.

Leveraging Rings in Lattice-Based Cryptography: The Ring Learning With Errors (Ring-LWE) problem is a variant of LWE formulated over a polynomial ring. In Ring-LWE key exchange, two parties generate public and private keys based on the hardness of the Ring-LWE problem [9, 10]. Here's a high-level overview:

1. **Public Key Generation:**
   - Select a random polynomial $a(x)$ and compute $b(x) = a(x) \cdot s(x) + e(x)$, where $s(x)$ is a small secret polynomial and $e(x)$ is a small error polynomial.
2. **Private Key Generation:**
   - Keep $s(x)$ as the secret key.
3. **Shared Key Generation:**
   - The shared key is derived by the other party using $b(x)$ and their secret key.

The hardness of Ring-LWE implies the security of the key exchange protocol. Assume there exists an efficient algorithm A that can break the Ring-LWE problem. We can use A to break the Ring-LWE-based key exchange. Given a public key $b(x)$, A can recover the secret key $s(x)$ (See [7] and [9]).

1.6 Error-Correcting Codes

Lattice-based cryptography often uses error-correcting codes, which can be viewed as linear codes with a group structure. These codes play a crucial role in designing cryptographic schemes that are secure against various attacks. I recommend you to read the work of [4]. An error-correcting code is a linear code if the set of codewords forms a vector space over a finite field. Let's denote a linear code as C with codewords $c_1, c_2, \ldots, c_k$. In the context of lattice-based cryptography, the set of codewords C forms an abelian group. Consider the (7, 4) Hamming code, a classic linear error-correcting code. It corrects single errors in a block of data.
Codewords in the Hamming Code: $C = \{(0000), (0001), (0010), \ldots, (1111)\}$. The Hamming code forms a group under vector addition.

1.7 Hardness Assumptions

Many lattice-based cryptographic protocols rely on the hardness of problems associated with lattices, and these problems are formulated in a way that involves group-theoretic concepts. The security of lattice-based cryptography is often based on the assumed hardness of solving certain lattice problems, which are inherently connected to group theory.

Proposition 1.11 ([19])**.** *The hardness of certain lattice problems is crucial for the security of lattice-based cryptographic schemes.*

Assume, for the sake of argument, that there exists an efficient algorithm A that can solve the LWE problem (or its variants) for certain instances. We can use A to break the security of lattice-based cryptographic schemes. For example, the ability to solve LWE may allow an attacker to recover secret keys or decrypt ciphertexts in lattice-based encryption schemes. The assumption that an efficient algorithm exists to solve the lattice problem contradicts the presumed hardness of these problems. If lattice problems are easy to solve, the security of lattice-based cryptographic schemes would be compromised.

2. Proposed Scheme for Group Theory in Lattice-Based Cryptography

1. Define a group structure on a lattice L.
   - (a)
     - i. Let L be a lattice with basis vectors $b_1, b_2, \ldots, b_n$.
     - ii. Define the group operation $\oplus$ as vector addition in $L : v \oplus u = v + u$.
     - iii. L forms an abelian group under $\oplus$.
2. Create subgroups within the lattice group for cryptographic applications.
   - (a)
     - i. Select a subset of basis vectors $\{b_{i1}, b_{i2}, \ldots, b_{iA}\}$ to form a subgroup H of L.
     - ii. $H = \left\{ \sum_{j=1}^{k} x_j b_{ij} : x_j \in Z \right\}$
     - iii. H is a subgroup under $\oplus$.
3. Utilize left cosets for cryptographic operations.
   - (a) Given a subgroup H in L, define left cosets gH for all $g \in L$.
     - i. $gH = \{g \oplus h : h \in H\}$.
   - (b) Use left cosets in key exchange or encryption schemes.
     - i. For example, in Ring-LWE key exchange, parties can use left cosets to exchange information securely.
4. Integrate public key cryptography into the lattice group. I recommend you to read [1].
   - (a)
     - i. Public Key: Choose a random lattice point P as the public key.
     - ii. Private Key: Keep the basis vectors used to generate P as the private key.
   - (b) Encryption and Decryption:
     - i. Encrypt a message M by adding it to the public key: $C = M \oplus P$.
     - ii. Decrypt by subtracting the ciphertext from the private key: $M = C \ominus P$.
5. Implement digital signatures using lattice-based group operations. See the work of [2]
   - (a)
     - i. Choose a random lattice point R as the signature.
     - ii. Compute the public key P.
     - iii. The signature is $S = R \oplus P$.

iv. Verify a signature S by checking $S \in gH$ for the appropriate subgroup H and public key P.

6. Utilize group homomorphisms for secure transformations.

   (a) i. Define a group homomorphism between two lattice groups, allowing for secure transformations of group elements.

3. Application

To generate a cipher for a word, say "MICHAEL" using our proposed scheme. Keep in mind that real lattice-based cryptography involves more complex mathematical structures and algorithms. This example is for illustrative purposes. Let's represent each letter of the word "MICHAEL" as vectors in a lattice group. Assume we have a lattice L with basis vectors $b_1, b_2, \ldots, b_n$. The letters will be represented as linear combinations of these basis vectors. Let's use the letters' positions in the English alphabet as coefficients in the linear combinations. For simplicity, let's use a 2-dimensional lattice with basis vectors $b_1 = (1, 0)$ and $b_2 = (0, 1)$.

1. **Define the Basis:**
   (a) $b_1 = (1, 0)$
   (b) $b_2 = (0, 1)$
2. **Represent Letters:**
   (a) $2M = 13b_1 + 9b_2$
   (b) $2I = 9b_1 + 8b_2$
   (c) $2C = 3b_1 + 2b_2$
   (d) $2H = 8b_1 + 7b_2$
   (e) $2A = 1b_1 + 0b_2$
   (f) $2E = 5b_1 + 4b_2$
   (g) $2L = 12b_1 + 11b_2$
3. **Generate Cipher:**
   (a) Let's assume a public key $2P = 20b_1 + 15b_2$.
   (b) $CM = M \oplus P$
   (c) $CI = I \oplus P$
   (d) $CC = C \oplus P$
   (e) $CH = H \oplus P$

(f) $CA = A \oplus P$

(g) $CE = E \oplus P$

(h) $CL = L \oplus P$

(i) Compute each C_i using vector addition in the lattice.

This is a simplified example, and in real-world lattice-based cryptography, more sophisticated algorithms and structures would be used. The example illustrates the basic concept of representing letters as lattice vectors and encrypting them using lattice-based operations. For simplicity, we will represent each letter with its corresponding position, the word "MICHAEL" is represented by lattice points corresponding to the letters.
MICHAEL: (2,3) : (1,4) : (5,2) : (3,1) : (4,5) : (6,6) : (7,7)
Permute Cipher: MICHAEL: (5,2) : (3,1) : (6,6) : (4,5) : (1,4) : (7,7) : (2,3)
Ordered Pair: MICHAEL: (5,2) : (3,1) : (6,6) : (4,5) : (1,4) : (7,7) : (2,3)
Vertex Labeling: Label each lattice point with the corresponding letter.

- (5,2) : C
- (3,1) : I
- (6,6) : E
- (4,5) : H
- (1,4) : A
- (7,7) : L
- (2,3) : M

Table Representation:

Original Text	Permute Cipher	Ordered Pair	Vertex Labeling
M	C	(5, 2)	C
I	I	(3, 1)	I
C	E	(6, 6)	E
H	H	(4, 5)	H
A	A	(1, 4)	A
E	L	(7, 7)	L
L	M	(2, 3)	M

Please note that this is a conceptual example, and the actual encryption process in lattice-based cryptography may involve more sophisticated mathematical operations and security considerations. The example aims to provide an intuitive representation of the encryption process using group theory in a lattice-based cryptographic context. Another way you can create a cipher for the word "MICHAEL" using group theory in lattice-based cryptography involves representing each letter as a vector in a lattice space. For simplicity, let's consider a 2-dimensional lattice. See[11] also for Cryptographic multilinear maps.

Scheme for Cipher Calculation:

1. **Assign Vectors to Letters:** Assign each letter a vector in a 2-dimensional lattice space. For example:
   - $M \to (1, 0)$
   - $I \to (0, 1)$
   - $C \to (1, 1)$
   - $H \to (2, 1)$
   - $A \to (2, 0)$
   - $E \to (3, 2)$
   - $L \to (3, 1)$
2. **Original Text:**
   - Represent the original word "MICHAEL" as a sequence of vectors: (1,0), (0,1), (1,1), (2,1), (2,0), (3,2), (3,1)
3. **Cyclic Permutation:**
   - Define a cyclic permutation to permute the vectors. For example, let's rotate the vectors to the right by 2 positions.
   - Apply the cyclic permutation to the original text:
     - (3,2),(3,1),(1,0),(0,1),(1,1),(2,1),(2,0)(3,2)
4. **Ordered Pair:**
   - Form ordered pairs using the original and permuted texts:
   - Ordered Pair: (1,0), (3,2)(0,1), (3,1), (1,1), (1,0)(2,1), (0,1), (2,0), (1,1)(2,0), (1,1), (3,2), (2,1), (3,1), (2,0)
5. **Vertex Labeling:**
   - Label each vector with the corresponding letter:
     - Vertex Labeling: (M,E), (I,L), (C,M), (H,I), (A,C), (E,H), (L,A)

This table like the other one illustrates the process of creating a cipher for the word "MICHAEL" using group theory in lattice-based cryptography. The cyclic permutation provides a form of encryption, and the ordered pairs and vertex labeling offer a representation of the encrypted text. Note that the specific choice of lattice and permutation can be adjusted based on the cryptographic scheme [16].

Original Text	Permuted Text	Ordered Pair	Vertex Labeling
(1, 0)	(3, 2)	(1, 0), (3, 2)	(M, E)
(0, 1)	(3, 1)	(0, 1), (3, 1)	(I, L)
(1, 1)	(1, 0)	(1, 1), (1, 0)	(C, M)
(2, 1)	(0, 1)	(2, 1), (0, 1)	(H, I)
(2, 0)	(1, 1)	(2, 0), (1, 1)	(A, C)
(3, 2)	(2, 1)	(3, 2), (2, 1)	(E, H)
(3, 1)	(2, 0)	(3, 1), (2, 0)	(L, A)

Table 1: Cipher Table

3.1 Algorithm

Implementing group theory in lattice-based cryptography involves creating algorithms for various operations such as permutation, encryption, and decryption. Read [19] to get more insight on computational group theory. Below is a simple Python code example for a 2-dimensional lattice that includes functions for permutation, encryption, and decryption. Read also, [4], and [13].

```
import numpy as np

# Define the basis vectors for the lattice
basis_vectors = np.array([[1, 0], [0, 1]])
# Define the original word "MICHAEL" as a sequence of vectors
original_text = [np.array([1, 0]), np.array([0, 1]), np.array([1, 1]),
    np.array([2, 1]), np.array([2, 0]), np.array([3, 2]), np.array([3, 1])]

# Define a cyclic permutation function
def cyclic_permutation(text, positions):
    return [text[(i - positions) % len(text)] for i in range(len(text))]

# Define a function for encryption
def encrypt(text, basis_vectors):
    encrypted_text = [np.dot(basis_vectors, vector) for vector in text]
    return encrypted_text

# Define a function for decryption
def decrypt(text, inverse_basis_vectors):
    decrypted_text = [np.dot(inverse_basis_vectors, vector) for vector in text]
    return decrypted_text

# Define a function for vertex labeling
```

```
def vertex_labeling(text):
    letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    return [(letters[int(vector[0])], letters[int(vector[1])]) for vector in text]

# Perform cyclic permutation
permuted_text = cyclic_permutation(original_text, positions=2)
# Perform encryption
encrypted_text = encrypt(permuted_text, basis_vectors)

# Perform decryption
inverse_basis_vectors = np.linalg.inv(basis_vectors)
decrypted_text = decrypt(encrypted_text, inverse_basis_vectors)

# Perform vertex labeling
vertex_labels = vertex_labeling(decrypted_text)

# Display the results
print("Original Text:", original_text)
print("Permuted Text:", permuted_text)
print("Encrypted Text:", encrypted_text)
print("Decrypted Text:", decrypted_text)
print("Vertex Labeling:", vertex_labels)
```

This code provides a simple example of group theory operations in lattice-based cryptography. The cyclic_permutation function simulates a cyclic permutation, the encrypt function performs encryption using lattice vectors, the decrypt function reverses the encryption, and the vertex_labeling function assigns letters to lattice points. The results for the word "MICHAEL" are displayed, showcasing the different steps in the lattice-based cryptographic process. Note that this is a simplified example, and actual implementations may involve more complex mathematical structures and considerations.

4. Conclusion

The integration of group theory into lattice-based cryptography proves to be a powerful and versatile approach. The abstract algebraic structures formed by lattices and their subgroups offer a robust foundation for designing cryptographic schemes resistant to classical and quantum attacks. The hardness of lattice problems, formulated using group-theoretic concepts, contributes to the security of lattice-based protocols. As research progresses, further exploration of advanced group-theoretic techniques promises to enhance the resilience and efficiency of lattice-based cryptographic systems, establishing them as viable alternatives in the evolving landscape of secure communications and data protection. In this paper, we have outlined a cryptographic scheme based on group theory in lattice-based. Here, we also provide example algorithm. One can modify this by changing the considered permutation or edit the algorithm.

References

[1] I. Anshel, M. Anshel and D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Let., 6(1999), 287–291.

[2] I. Anshel, D. Atkins, D. Goldfeld and P. E. Gunnells, *WalnutDSA: a group theoretic digital signature algorithm*, International Journal of Computer Mathematics: Computer Systems Theory, 6(4)(2021), 260–284.

[3] C. Battarbee, D. Kahrobaei and S. F. Shahandashti, *Cryptanalysis of semidirect product key exchange using matrices over non-commutative rings, in MathCrypt 2021*, Journal of Mathematical Cryptology, 1(2)(2021), 2–9.

[4] R. Flores, D. Kahrobaei and T. Koberda, *Algorithmic problems in right-angled artin groups: complexity and applications*, J. Algebra, 519(2019), 111–129.

[5] J. Gryak and D. Kahrobaei, *The status of polycyclic group-based cryptography: A survey and open problems*, Groups Complexity Cryptology, 8(2016), 171–186.

[6] J. Gryak, D. Kahrobaei and C. Martinez-Perez, *On the conjugacy problem in certain metabelian groups*, Glasgow Mathematical Journal, 61(2)(2019), 251–269.

[7] K. Horan and D. Kahrobaei, *Hidden Subgroup Problem and Post-quantum Group-based Cryptography*, International Congress on Mathematical Software – ICMS 2018, LNCS, (2018), 218–226.

[8] M. Habeeb, D. Kahrobaei, C. Koupparis and V. Shpilrain, *Public key exchange using semidirect product of (semi) groups*, Applied Cryptography and Network Security, 2013(2013), 475–486.

[9] D. Kahrobaei and C. Koupparis, *Noncommutative digital signatures using noncommutative groups, Groups, Complexity, Cryptology*, De Gruyter, (2012), 377–384.

[10] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang and C. Park, *New public-key cryptosystem using braid groups*, Advances in Cryptology, 1880(2000), 166–183.

[11] D. Kahrobaei, H. Lam and V. Shpilrain, *System and method for private-key fully homomorphic encryption and private search between rings*, US Patent 10, 396, 976(2019).

[12] D. Kahrobaei and V. Shpilrain, *Using semidirect product of (semi) groups in public key cryptography*, Computability in Europe, LNCS, (2016), 132–141.

[13] D. Kahrobaei and M. Stanojkovski, *Cryptographic multilinear maps using pro-p groups*, Advances in Mathematics of Communications, (2021), 1–14.

[14] NIST, *Post-Quantum Cryptography PQC*, (2022), https://csrc.nist.gov/News/2022/ pqc-candidates-to-be-standardized-and-round-4.

[15] A. Sutherland, *Structure computation and discrete logarithms in finite abelian p-groups*, Mathematics of Computation, 80(2011), 477–500.

[16] A. Wood, K. Najarian and D. Kahrobaei, *Homomorphic encryption for machine learning in medicine and bioinformatics*, ACM Comput. Surv., 53(4)(2020), 1–35.

[17] O. G. Udoaka and E. A. Frank, *Finite Semi-group Modulo and Its Application to Symmetric Cryptography*, International Journal of Pure Mathematics, (2022).

[18] Joan S. Birman, Volker Gebhardt and Juan Gonzalez-Meneses, *Conjugacy in Garside groups I: cycling, powers and rigidity, Groups Geom*, Dynamics, 1(2007), 221–279.

[19] Oded Goldreich, Shafi Goldwasser and Shai Halevi, *Public-key cryptosystems from lattice reduction problems*, in Advances in Cryptology – CRYPTO 97 (B.S. Kaliski Jr, ed.), Lecture Notes in Computer Science 1294 (Springer, Berlin, 1997), 112–131.

[20] Michael N. John and O. G. Udoaka, *Algorithm and Cube-Lattice-Based Cryptography*, International journal of Research Publication and Reviews, 4(10)(2023), 3312-3315.

[21] Michael N. John, and O. G. Udoaka, *Computational GroupTheory and Quantum-Era Cryptography*, International Journal of Scientific Research in Science, Engineering and Technology, 10(6)(2023), 1-10.

[22] Iris Anshel, Michael Anshel and Dorian Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett., 6(1999), 287-291.