MSc Dissertation

by Kalyani Chaturvedi

Submission date: 20-May-2026 05:42PM (UTC+0530)
Submission ID: 2965612662
File name: Kalyani_Dissertation.docx (1.38M)
Word count: 9632
Character count: 55858

LINEAR ALGEBRA IN CRYPTOGRAPHY AND SECURE COMMUINCATION

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF SCIENCE
IN
APPLIED MATHEMATICS

Submitted by

KALYANI CHATURVEDI (24/MSCMAT/60)

Under the supervision of
Asst. Prof. Mr. Jamkhongam Touthang

DEPARTMENT of APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2026

DEPARTMENT OF APPLIED MATHEMATICS

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, KALYANI CHATURVEDI, Roll No – 24/MSCMAT/60 student of MSc. (Applied Mathematics), hereby declare that the project Dissertation titled "**LINEAR ALGEBRA IN CRYPTOGRAPHY AND SECURE COMMUINCATION**" which is submitted by me to the Department of Applied Mathematics, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Science, is original and not copied from any source without proper citation. The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Place: Delhi — Kalyani Chaturvedi

Date: 23.05.2026 — 24/MSCMAT/60

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

DEPARTMENT OF APPLIED MATHEMATICS

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**LINEAR ALGEBRA IN CRYPTOGRAPHY AND SECURE COMMUINCATION**" which is submitted by KALYANI CHATURVEDI, Roll No – 24/MSCMAT/60, Department of Applied Mathematics, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Science, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 23.05.2026

Asst. Prof. Mr. Jamkhongam Touthang
SUPERVISOR

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to Asst. Prof. Mr. Jamkhongam Touthang for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve our targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi — Kalyani Chaturvedi
Date: 23.05.2026 — 24/MSCMAT/60

Abstract

This study examines the role of linear algebra in the development and analysis of cryptographic systems, motivated by the growing need for mathematically robust security frameworks in an era of expanding digital infrastructure.

The research analyzes both classical and modern encryption techniques — including the Hill cipher, affine transformations, RSA, and AES — to trace how algebraic structures directly shape cryptographic design. Core concepts such as matrix invertibility, modular arithmetic, and finite field operations were studied in detail, with particular attention to how these properties determine the security and reversibility of encryption schemes. The Hill cipher served as a foundational case, illustrating how the invertibility of a key matrix under modular arithmetic is not merely a mathematical convenience but the actual mechanism of decryption. AES extended this further, where MixColumns operates as a matrix multiplication over $GF(2^8)$, selected for measurable diffusion strength rather than arbitrary construction.

Decomposition methods — LDU, QR, and SVD — were also examined for their computational relevance, particularly in efficient matrix operations and their emerging role in lattice-based cryptographic frameworks that aim to resist quantum attacks.

The findings confirm that linear algebra is central to cryptographic design, governing both security guarantees and implementation efficiency across the systems studied.

Overall, this study establishes that linear algebra provides not just theoretical grounding for cryptography but the actual design logic that makes modern encryption systems function — and holds up under adversarial conditions.

Contents

List of Tables

List of Figures

List of Symbols, Abbreviations

AES	Advanced Encryption Standard
RSA	Rivest–Shamir–Adleman (Public-Key Cryptosystem)
LWE	Learning With Errors
RLWE	Ring Learning With Errors
GF(2^8)	Galois Field with 256 elements (used in AES)
Z_{26}	Integer ring modulo 26 (used in Hill Cipher)
NIST	National Institute of Standards and Technology
PQC	Post-Quantum Cryptography
ECC	Elliptic Curve Cryptography
SVD	Singular Value Decomposition
LDU	Lower–Diagonal–Upper Decomposition
QR	QR Decomposition (Orthogonal-Triangular)
NTT	Number Theoretic Transform
MFA	Multi-Factor Authentication
LDPC	Low-Density Parity-Check (Code)
SEC	Single Error Correcting
FEC	Forward Error Correction
CIA+N	Confidentiality, Integrity, Authentication, Non-repudiation
det(K)	Determinant of key matrix K
gcd(a,b)	Greatest Common Divisor of a and b
mod n	Modulo n operation

$\mathbf{K^{-1}}$	Modular inverse of key matrix K
C	Ciphertext vector
P	Plaintext vector
H	Parity-check matrix
G	Generator matrix
S	Syndrome vector
e	Error vector (in coding theory / LWE)
s	Secret vector (in LWE)
A	Public random matrix (in LWE)
b	Observed output vector (in LWE)
q	Modulus in LWE lattice problems
Σ (Sigma)	Diagonal matrix of singular values (in SVD)
U, $\mathbf{V^T}$	Orthogonal matrices in SVD decomposition $A = U\Sigma V^T$
SVP	Shortest Vector Problem
$\mathbf{d_m Z^n}$	Minimum Hamming distance of a code
w(x)	Hamming weight of vector x
m(x)	Irreducible polynomial (AES: $x^8 + x^4 + x^3 + x + 1$)
R_q	Polynomial ring $Z_q[x]/(x^n + 1)$ (Ring-LWE)

Chapter 1

Introduction

1.1 Background

Cryptography is the discipline that converts information into a protected form so that it can travel through an insecure environment without becoming intelligible to an unauthorized reader. Earlier methods of secrecy often depended on clever substitutions or rearrangements of letters. Contemporary cryptography, by contrast, is built on exact mathematical structures. The change is important: a modern cipher is not judged by how complicated it looks, but by whether its security can be described and tested through algebra, number theory, probability, and computation.

Linear algebra enters this discussion in a natural way. Digital messages are represented as strings of numbers; blocks of such numbers are vectors; encryption rules often act on these vectors by matrices, finite-field operations, or structured transformations. Once a message is expressed in this language, encryption can be studied as a map from a plaintext space to a ciphertext space. Decryption then becomes the problem of reversing that map with the correct key.

The present dissertation studies this algebraic viewpoint across three stages of cryptographic development. The first stage is classical matrix encryption, represented by the Hill cipher and affine transformations. The second stage is modern symmetric and communication security, especially the use of finite fields and matrix operations in AES and the use of linear codes for reliable transmission. The third stage is post-quantum cryptography, where high-dimensional vector and lattice problems have become central to the design of quantum-resistant systems.

The central idea of the work is therefore not that every cryptographic system is linear. Many secure systems deliberately combine linear operations with nonlinear ones. Rather, the claim is that linear algebra supplies the language in which a large part of cryptographic design, analysis, implementation, and error control can be clearly expressed.

1.1.1 Mathematical Function in Cryptography

Mathematics gives cryptography its precision. A secure communication system must specify how a message is transformed, what information is public, what information is secret, and why an adversary should not be able to reverse the process efficiently. Linear algebra contributes to each of these questions. It represents messages as vectors, records keys as matrices or structured algebraic objects, and gives exact conditions for reversibility.

In the Hill cipher, for example, a plaintext block is written as a column vector and multiplied by a key matrix. The computation is performed modulo 26 when the English alphabet is used. The ciphertext can be decrypted only when the key matrix has a modular inverse. Thus the determinant condition is not an optional calculation; it is the mathematical reason the cipher can be reversed by the intended receiver.

The same pattern, in a more sophisticated form, appears in AES. The MixColumns layer in AES is a matrix transformation over the finite field GF(2^8). This operation diffuses each byte of a column across the other bytes, ensuring that a small input change influences several output positions. In post-quantum systems such as Learning With Errors (LWE), the public data are

often expressed using matrices and vectors modulo a large integer, while carefully added noise prevents direct solution of the underlying linear equations.

- modular arithmetic, which keeps operations inside a fixed symbol set or finite algebraic domain;
- matrix multiplication, which provides an organized mechanism for block transformation;
- determinants and inverses, which decide whether a key transformation can be undone;
- finite fields, which allow well-defined addition, multiplication, and division in modern ciphers;
- vector spaces and subspaces, which support coding theory, syndrome decoding, and lattice-based security.

1.2 Problem Statement

Many introductory accounts of cryptography describe algorithms as separate techniques: the Hill cipher as a historical matrix cipher, AES as a modern block cipher, error-correcting codes as a communication tool, and LWE as a post-quantum construction. This separation can hide the common mathematical structure behind them. The problem addressed in this dissertation is the need for a unified explanation of how linear algebra appears in encryption, decryption, error control, and post-quantum security.

The study asks how matrix transformations, modular arithmetic, finite fields, and vector-space methods operate within cryptographic systems, and where their strengths and limitations become visible. It also examines why purely linear encryption is not sufficient for modern security, even though linear algebra remains essential in advanced designs.

1.3 Objective of the Study

The main objective is to explain the role of linear algebra as a working mathematical foundation for cryptography and secure communication. The dissertation is analytical rather than experimental: it studies established algorithms and mathematical models, clarifies the algebra behind them, and compares their roles in different security settings.

- to describe how matrices and vectors model message blocks, keys, ciphertexts, and codewords;
- to explain why modular arithmetic and finite fields are required for reversible and bounded computation;
- to analyze the Hill cipher as both a useful teaching model and a weak practical cipher;
- to connect AES diffusion, coding theory, and lattice-based cryptography through algebraic transformations;
- to outline the relevance of matrix and vector problems in post-quantum cryptography.

1.4 Scope and Limitation

The scope of the dissertation is organized in three levels. At the classical level, the Hill cipher and affine cipher are used to demonstrate encryption by modular linear transformation. At the modern level, AES is discussed through its finite-field and matrix-based components, and error-correcting codes are studied as a reliability layer for secure communication. At the post-quantum level, lattice and module-lattice ideas are introduced through LWE, Ring-LWE, and the standardized family of lattice-based constructions such as ML-KEM and ML-DSA.

The study does not introduce a new cryptographic algorithm, nor does it claim to provide implementation-level benchmarks. Numerical examples are included for explanation, but they serve as mathematical illustrations rather than as software experiments. Side-channel attacks, full protocol engineering, hardware performance, and formal reduction proofs are outside the main scope. These limitations are important because practical cryptographic security depends not only on correct mathematics but also on implementation, protocol use, randomness, and system design.

1.5 Contribution of the Study

The dissertation contributes a unified reading of linear algebra across several cryptographic settings. It does not treat linear algebra only as a set of background formulas. Instead, it shows how algebraic structure determines whether a cipher can be reversed, whether an error can be detected, whether a message can be reconstructed, and why some high-dimensional problems remain difficult for known algorithms.

1. It presents classical, modern, and post-quantum examples through one mathematical language: vectors, matrices, finite fields, and modular equations.
2. It explains the known-plaintext weakness of the Hill cipher as a direct consequence of linearity rather than as a separate cryptanalytic trick.
3. It links cryptography and coding theory by showing that generator and parity-check matrices perform structured transformations similar in spirit to encryption and verification.
4. It introduces LWE and Ring-LWE as noisy versions of modular linear systems, clarifying why noise changes an easy linear algebra problem into a difficult cryptographic one.
5. It gives a reorganized account suitable for readers who know elementary linear algebra and want to see how it supports secure communication.

1.6 Thesis Organization

Chapter 1 introduces the topic, states the motivation and objectives, and defines the scope of the work.

Chapter 2 develops the required algebraic background, including matrices, modular arithmetic, finite fields, and the security principles used in cryptographic communication.

Chapter 3 explains the methodology and applies linear algebra to selected cryptographic models, including the Hill cipher, AES-style finite-field transformations, matrix factorizations, and post-quantum examples.

Chapter 4 focuses on error correction and secure communication channels. It studies generator matrices, parity-check matrices, syndrome decoding, and Hamming codes. Chapter 5 discusses

the growth and future direction of matrix-based cryptography, with emphasis on LWE, Ring-LWE, and current post-quantum schemes.

Chapter 6 summarizes the findings, answers the research questions, states limitations, and suggests directions for further work.

Chapter 2

Foundation of Linear Algebra for Cryptography

2.1 Vector Spaces and Their Properties

A vector space is a collection of objects that can be added together and multiplied by scalars while remaining inside the same collection. In cryptography, the objects are usually not geometric arrows but blocks of data. A pair of letters, a byte, a column of four bytes, or a binary codeword can all be treated as vectors once a suitable algebraic domain has been fixed.

For alphabetic ciphers, the domain may be the residue class ring Z_{26}. For binary systems it is often the field $GF(2)$, and for AES it is $GF(2^8)$. This choice of domain is decisive. It determines the meaning of addition, multiplication, inverses, and zero. The same matrix can behave very differently over the real numbers, over Z_{26}, and over a finite field.

A matrix represents a linear transformation. If P is a plaintext vector and $K$is a key matrix, a simple matrix cipher can be written as

$$C = KP(mod\ n).$$

Here C denotes the ciphertext vector and n is the modulus. The legitimate receiver recovers the plaintext by applying the inverse transformation:

$$P = CK^{-1}(mod\ n).$$

The formula looks simple, but it contains the essential issue: decryption is possible only if K^{-1} exists in the chosen modular system. For a square matrix over Z_{26} , the determinant of K must be relatively prime to 26. If this condition fails, different plaintext blocks may collapse into the same ciphertext block, and unique recovery becomes impossible.

Determinants therefore play a security and correctness role in matrix ciphers. They do not make the Hill cipher secure against modern attacks, but they decide whether the intended decryption operation is even mathematically valid. In this way, the determinant becomes a practical cryptographic test.

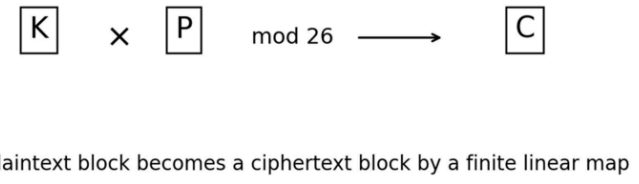

Figure 2.1 Matrix operation in encryption (Hill cipher)

2.1 (A) Finite Fields and $GF(2^8)$: The Mathematical Engine for AES

The Hill cipher works with residues modulo 26, which is convenient for alphabetic examples but not a field because 26 is composite. AES uses a more suitable structure: the finite field $GF(2^8)$. This field contains 256 elements, exactly the number of possible byte values. Each byte can be viewed as a polynomial of degree at most seven with coefficients in $GF(2)$.

For example, the byte 01010111 represents the polynomial

$$x^6 + x^4 + x^2 + x + 1.$$

Addition in this field is bitwise XOR. Multiplication is polynomial multiplication followed by reduction modulo a fixed irreducible polynomial. This construction allows every nonzero byte to have a multiplicative inverse, which is a property needed in several parts of AES.

2.1 (A.1) What is a Galois Field $GF(2^8)$?

A Galois field $GF(p^r)$ is a finite field with p^r elements, where p is prime and r is a positive integer. In AES, p = 2 and r = 8. Thus $GF(2^8)$is the field of 256 elements used to process bytes. The representation by binary polynomials is not merely a change of notation; it is the rule that defines the arithmetic of the cipher.

2.1 (A.2) The Irreducible Polynomial

AES defines multiplication in $GF(2^8)$ using the irreducible polynomial

$$m(x) = x^8 + x^4 + x^3 + x + 1.$$

Reduction by this polynomial keeps the product of two bytes inside the set of byte-sized field elements. It plays a role similar to a modulus, but unlike reduction modulo 26, it is chosen so that the resulting structure is a field.

2.1 (A.3) Worked Example: Multiplication in $GF(2^8)$

Suppose two bytes values are represented by the polynomials $a(x) = x^6 + x^4 + x^2 + x + 1$ and $b(x) = x^7 + x^1 + 1$.

Their product is first computed as an ordinary polynomial over $GF(2)$, where terms with even coefficient cancel. The resulting polynomial is then reduced modulo m(x) until its degree is less than eight. The final byte is the representative of the product in $GF(2^8)$.

This example illustrates the difference between ordinary arithmetic and finite-field arithmetic. In a field used for cryptography, the aim is not to obtain a large integer or polynomial, but to keep every result inside a controlled algebraic universe where inverse operations are well defined.

Mathematical connection: *This is directly analogous to the Hill Cipher computing C = KP (mod 26). In both cases, multiplication is performed in a finite structure and the result is reduced to remain within a bounded set. GF(2⁸) is simply a more sophisticated and secure finite structure than Z_{26}.*

2.1 (A.4) The Mix-Columns Matrix Operation in AES

The Mix-Columns step in AES applies a fixed 4 by 4 matrix to each column of the internal state. The entries 01, 02, and 03 are field elements, and every addition and multiplication is performed in $GF(2^8)$.

The purpose of this layer is diffusion. A change in one input byte is spread across the output column. Unlike the Hill cipher, AES does not rely only on linear transformation; it combines linear diffusion with nonlinear substitution. This mixture is what prevents the simple linear recovery attacks that defeat classical matrix ciphers.

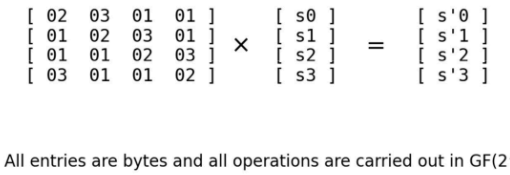

Figure 2.2 Mix-Columns matrix operation in AES over $GF(2^8)$

2.2 Mathematical Principles of Cryptography and Secure Communication

2.2.1 Classical Cryptography

Classical cryptography includes substitution, transposition, affine, and matrix ciphers. These systems are useful for learning because their rules can be written explicitly and calculated by hand. Their weakness is also visible: most of them preserve too much structure. If the transformation is linear or nearly linear, an attacker can often exploit frequency patterns or solve a system of equations.

The Hill cipher is historically important because it was among the earliest ciphers to use matrix algebra directly. It showed how a block of letters could be encrypted together rather than letter by letter. However, its linearity makes it unsuitable for real security today.

2.2.2 Modern Encryption

Modern encryption systems are designed for digital data and adversaries with substantial computational power. They must protect communication over public networks where messages can be intercepted, copied, delayed, altered, or replayed. A secure design therefore needs confidentiality, integrity, authentication, and often non-repudiation.

Symmetric encryption uses a shared secret key. AES is the most common example. Asymmetric encryption uses a public key for one operation and a private key for the reverse operation. RSA and elliptic-curve schemes are standard examples of pre-quantum public-key cryptography. Post-quantum systems increasingly use lattice, code, hash, and multivariate constructions.

2.2.3 Core Principles of Cryptography

The goals of secure communication may be summarized as confidentiality, integrity, authentication, and non-repudiation. Confidentiality prevents unauthorized reading. Integrity detects unauthorized alteration. Authentication verifies identity or origin. Non-repudiation links a digital action to the party that performed it so that the action cannot later be denied.

Linear algebra contributes most directly to the transformation and verification components of these goals. A matrix may hide data by transforming it, a parity-check matrix may reveal whether a received word is valid, and a lattice-based public key may conceal a secret vector behind noisy modular equations. These operations do not by themselves solve every security requirement, but they provide essential computational machinery.

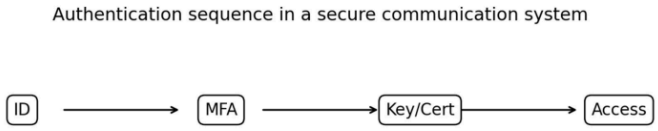

Figure 2.3 Authentication procedure in cryptographic communication

2. 4 Linear Algebraic Methods in Symmetric and Asymmetric Cryptosystems

2.4.1 Hill Cipher and Affine Cipher

An affine cipher maps a plaintext symbol $x\ to\ ax + b(mod26)$. The parameter a must be invertible $(mod26)$ otherwise, two different letters may produce the same ciphertext. The Hill cipher generalizes this idea from single symbols to vectors. It replaces multiplication by a scalar with multiplication by a key matrix.

$$C = KP + b(mod26)$$

In the pure Hill cipher, b is absent and the transformation is linear. In an affine block cipher, b adds a translation. Both systems require modular inverses for decryption. Both are also vulnerable because their algebraic structure is too transparent once enough plaintext-ciphertext pairs are known.

To encrypt the letter **C**, we first convert it into its numerical equivalent. Using the standard mapping (A = 0, B = 1, C = 2, …, Z = 25), the letter **C** corresponds to 2.
If we choose the encryption parameters $a = 3$and $b = 8$, the affine encryption function is defined as [12]:

$$E = 3m + 8(\text{mod } 26)$$

Substituting $m = 2$:

$$E(2) = 3(2) + 8 = 6 + 8 = 14$$

Since the number **14 corresponds to the letter O**, the encrypted form of **C is O [12].**

Decryption Process

To recover the original message, we reverse the encryption process. Let *s*represent the ciphertext value. The encryption equation is:

$$s \equiv 3m + 8 \ (\text{mod } 26)$$

First, subtract 8 from both sides:

$$s - 8 \equiv 3m \text{ mod } 26$$

Both sides are then scaled by the multiplicative inverse of 3 in Z_{26}. This inverse is 9, since $3 * 9 = 27$, which reduces to 1 under mod 26 arithmetic [5]. Applying this gives:

$$m \equiv 9(s - 8)(mod 26)$$

The value of m obtained here is the numerical equivalent of the original plaintext character, from which the corresponding alphabetic letter is directly retrieved [18].

2.4.2 Linear Codes

A linear code is a subspace of a finite vector space. A message vector is multiplied by a generator matrix G to produce a codeword. The receiver uses a parity-check matrix H to test whether the received vector lies in the valid code space. This is again a linear algebraic method, but its purpose is reliability rather than secrecy

$$c = mG \ , Hc^T = 0.$$

The power of linear codes comes from structured redundancy. The added parity bits allow the receiver to detect, and sometimes correct, errors introduced by noise or interference.

2.4.3 Matrix Based Error Detection

When a codeword c is transmitted and an error vector e is added by the channel, the receiver obtains $r = c + e$. The syndrome is computed by multiplying by the parity-check matrix:

$$S = Hr^T = H(c + e)^T = Hr^T \ .$$

Since $Hc^T = 0$ for every valid codeword, the syndrome depends only on the error pattern. For well-designed codes, each correctable error has a distinctive syndrome. This allows efficient correction without testing every possible original message.

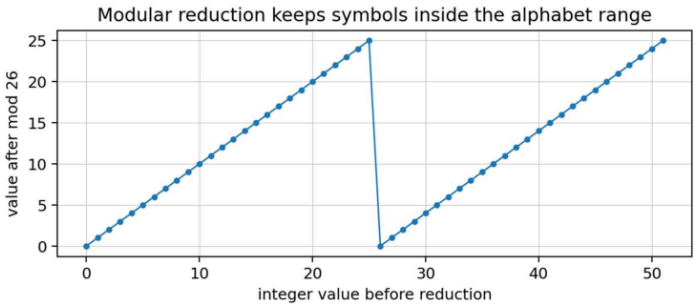

Figure 2.4 Modular arithmetic in a matrix-based cipher

2.5 Advances and Future Directions in Linear Algebra-Based Cryptography

2.5.1 Post-Quantum Cryptography

Quantum computing changes the risk profile of public-key cryptography. Shor's algorithm threatens RSA and elliptic-curve systems because it can solve factoring and discrete logarithm problems efficiently on a sufficiently powerful quantum computer. Lattice and code-based cryptography respond by relying on different mathematical problems, many of which are naturally expressed using vectors, matrices, and finite rings.

In lattice-based schemes, the central difficulty is not matrix multiplication itself. The hard problem arises because the public equations include noise or because the solution sought is unusually short in a high-dimensional lattice. Linear algebra supplies the setting; computational hardness comes from the geometry and the error structure.

2.5.2 Emerging Linear-Algebraic Protocols

Current research also studies matrix and tensor methods, code-based public-key encryption, homomorphic operations, and secure multi-party computation. In each area, algebraic representation is used to control how information is transformed and what can be learned from public data. The challenge is to keep legitimate computation efficient while making unauthorized inversion infeasible.

2.5.3 Open Research Challenges

The main open challenges include efficient implementation, resistance to side-channel leakage, precise parameter selection, and long-term standardization. A scheme that is secure in a mathematical model may still fail if implemented with poor randomness or if physical leakage reveals secret-dependent information. Future work must therefore combine algebraic design with careful engineering.

Chapter 3

Methodology and Applications of Linear Algebra in Cryptography

3.1 Introduction

This chapter applies the algebraic tools from Chapter 2 to selected cryptographic and communication models. The method is analytical: each system is described by identifying its data representation, transformation rule, reversibility condition, and security implication. The aim is not to provide software performance results but to show how the mathematics controls the behavior of the system.

The discussion proceeds from simple to advanced examples. The Hill cipher is used because its full algebra can be written down and checked. AES is discussed through its finite-field diffusion layer. Error-control coding is treated as a companion technology for secure communication. Finally, LWE is introduced as a post-quantum problem obtained by disturbing modular linear equations with small errors.

3.2 Cryptography Modelling with Linear Algebra

A cryptographic model begins by choosing a space for the data. In a letter-based Hill cipher, the space is Z_{26}^{n}. In AES, bytes live in $GF(2^8)$ and the state is arranged as a matrix of bytes. In LWE, public data may be represented by A in Z_{26}^{n*m} and b in Z_q^m. In every case, the selected algebraic domain determines what operations are legal and how inverses are computed.

The Hill cipher encrypts an n-letter block by multiplying a plaintext vector P by an n by n key matrix K modulo 26:

$$C = KP + b(mod26).$$

The decryption formula is

$$P = CK^{-1}(mod\ n).$$

For the inverse to exist, $\gcd(\det(k), 26) = 1$. This condition is easy to check and demonstrates the importance of modular invertibility. However, it also shows why the cipher is mathematically exposed: the map is linear, and linear maps can be recovered from enough input-output data.

Consider the key matrix

$$K = [[3,3].[2,5]]$$

Its determinant is 9, and $\gcd(9,26) = 1$. Therefore, it has a modular inverse. If the plaintext HI is encoded as $P = [7,8]^T$, then multiplication modulo 26 produces $C = [19,2]^T$, corresponding to TC. The calculation is small, but it displays the complete mechanism of a matrix cipher.

For a 3 by 3 version, the same principle applies. Larger blocks increase diffusion because each ciphertext symbol can depend on several plaintext symbols. Nevertheless, increasing the matrix size alone does not create modern security. If the transformation remains linear, an attacker can still recover the key after collecting enough independent examples.

3.2 (A) Vulnerability Analysis: The Known-Plaintext Attack on the Hill Cipher

The known-plaintext attack is the clearest demonstration of the Hill cipher's limitation. Suppose an attacker obtains n plaintext blocks and their corresponding ciphertext blocks for an n by n Hill cipher. Place the plaintext vectors as columns of a matrix P and the ciphertext vectors as columns of a matrix C. The encryption relation becomes

$$C = K \times P(\text{mod}26).$$

If P is invertible modulo 26, the attacker can multiply by P^{-1} and recover the secret key:

$$K = P^{-1} \times C(\text{mod}26).$$

This is complete key recovery. No exhaustive search is needed. The attack uses the same linear algebra that the legitimate receiver uses, except that it solves for the key rather than for the plaintext.

3.2 (A.1) Mathematics Basis of the Attack

For an n by n key matrix, n linearly independent plaintext blocks are enough to determine the transformation. The requirement is that the matrix P formed from those plaintext columns must have a determinant relatively prime to 26. If the first collected blocks do not satisfy this condition, the attacker can use different blocks. In a chosen-plaintext setting this is even easier because the attacker may select convenient vectors.

The attack shows the difference between correctness and security. The determinant condition makes the cipher reversible for the legitimate user, but it does not hide the linear relation. A correct cipher may still be cryptographically weak.

3.2 (A.2) Worked Numerical Example

Assume the unknown key is K = [[3, 3], [2, 5]], but the attacker observes two plaintext-ciphertext pairs. For the plaintext HI, $P_1 = [7,8]^T$ and $C_1 = [19,2]^T$. If the attacker also observes a second pair chosen so that the plaintext matrix is invertible modulo 26, then P and C can be assembled and the formula $K = CP^{-1}(mod26)$ gives the key directly.

Plaintext pair 1: "HI" $\rightarrow \boldsymbol{P_1 = [7,8]^T = C_1 = [19,2]^T =}$ **"TC"**

Plaintext pair 2: "LO" $\rightarrow \boldsymbol{P_2 = [11,14]^T = C_2 = KP_2(mod26)}$

First, let us compute C_2:

$\boldsymbol{C_2} = \begin{bmatrix}3 & 3\\ 2 & 5\end{bmatrix} * \begin{bmatrix}11\\ 14\end{bmatrix} = \begin{bmatrix}33+42\\ 22+70\end{bmatrix} = \begin{bmatrix}75\\ 92\end{bmatrix} (\text{mod } 26) \rightarrow$ **"XM"**

Attack step 1 — Assemble the matrices:

P = $\begin{bmatrix}7 & 11\\ 8 & 14\end{bmatrix}$, **C** = $\begin{bmatrix}19 & 23\\ 2 & 14\end{bmatrix}$

Attack step 2 — Compute det(P):

det(P) = (7)(14) − (11)(8) = 98 − 88 = 10

Check: $\gcd(10,26) = 2 \neq 1$. The attacker chooses a different second pair. This is realistic attackers can choose known plaintexts adaptively.

Using plaintext pair 2 as "AT" → $P_2 = [0,9]^T$.

$\mathbf{C_2} = \begin{bmatrix}3 & 3\\ 2 & 5\end{bmatrix} * \begin{bmatrix}0\\ 19\end{bmatrix} = \begin{bmatrix}57\\ 95\end{bmatrix}$ **→ "FT"**

Reassemble Matrices:

$$P = \begin{bmatrix}7 & 0\\ 8 & 19\end{bmatrix} C = \begin{bmatrix}19 & 5\\ 2 & 17\end{bmatrix}$$

Step 3: Compute Determinant of *P*

$$\det(P) = (7 \times 19) - (0 \times 8) = 133$$
$$133 \equiv 133 - 5(26) = 3 \pmod{26}$$
$$\gcd(3,26) = 1 \Rightarrow P \text{ is invertible mod } 26$$

Step 4: Find Modular Inverse of Determinant

We need:

$$3^{-1} \pmod{26}$$
$$3 \times 9 = 27 \equiv 1 \pmod{26}$$
$$\Rightarrow 3^{-1} = 9$$

Step 5: Compute P^{-1}(mod 26)

Using inverse formula:

$$P^{-1} = 9 \cdot \begin{bmatrix}19 & 0\\ -8 & 7\end{bmatrix} \pmod{26}$$
$$-8 \equiv 18 \pmod{26}$$
$$P^{-1} = 9 \cdot \begin{bmatrix}19 & 0\\ 18 & 7\end{bmatrix} = \begin{bmatrix}171 & 0\\ 162 & 63\end{bmatrix} \pmod{26}$$
$$P^{-1} = \begin{bmatrix}15 & 0\\ 6 & 11\end{bmatrix}$$

Step 6: Recover Key Matrix *K*

$$K = C \cdot P^{-1} \pmod{26}$$
$$K = \begin{bmatrix}19 & 5\\ 2 & 17\end{bmatrix} \cdot \begin{bmatrix}15 & 0\\ 6 & 11\end{bmatrix}$$
$$= \begin{bmatrix}(19 \cdot 15 + 5 \cdot 6) & (19 \cdot 0 + 5 \cdot 11)\\ (2 \cdot 15 + 17 \cdot 6) & (2 \cdot 0 + 17 \cdot 11)\end{bmatrix}$$
$$= \begin{bmatrix}315 & 55\\ 132 & 187\end{bmatrix} \pmod{26}$$
$$K = \begin{bmatrix}3 & 3\\ 2 & 5\end{bmatrix}$$

The important point is not the particular numbers but the mechanism: a linear cipher can be reconstructed from its action on a basis. Once the basis images are known, the entire transformation is known. This is a fundamental fact of linear algebra and the source of the Hill cipher's vulnerability.

3.2 (A.3) Implications and Countermeasures

The practical implication is that the Hill cipher should not be used for real secure communication. It is valuable as a teaching model but not as a modern security tool. A realistic block cipher must prevent the attacker from reducing the problem to a solvable linear system. AES achieves this by combining linear diffusion with nonlinear substitution, key addition, and multiple rounds. The nonlinear layer destroys the simple relation that the Hill cipher exposes.

Countermeasures such as increasing the matrix size or changing the alphabet may delay a simple manual attack, but they do not remove the algebraic weakness. Security requires a design that resists known-plaintext, chosen-plaintext, and other standard attack models.

3.3 Error Detection and Secure Communication Channels

Secure communication requires more than secrecy. A receiver must also know whether the message has arrived without accidental corruption or malicious alteration. Error-control coding addresses this reliability problem. Linear algebra is again central because codewords form structured subsets of vector spaces, and errors are detected by matrix multiplication.

If m is a message vector and G is a generator matrix, the transmitted codeword is c = mG. If noise adds an error vector e, the receiver obtains r = c + e. The parity-check matrix H gives the syndrome S = Hr^T. A zero syndrome indicates that r satisfies the code constraints; a nonzero syndrome indicates an error pattern.

In practice, encryption and error correction may be arranged in different orders depending on the system design. The important principle is that the reliability layer should protect the data path without exposing the secret key. For high-risk communication such as satellite links, military communication, and long-distance data transfer, this layer is indispensable.

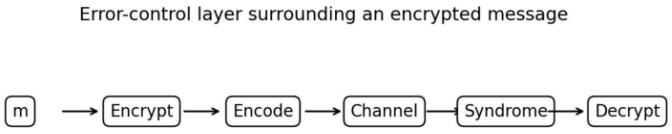

Figure 3.1 Error detection and correction method in secure communication using linear algebra

3.3.1 Simulated Analysis and Experimental Dataset

For comparative discussion, this dissertation uses theoretical performance indicators rather than laboratory measurements. The values in the following table should be read as illustrative benchmark-style estimates. They show the general trade-off between speed, detection ability, correction capacity, and algebraic complexity. They are not presented as new experimental measurements.

Table 3.1 Performance comparison of linear algebra-based cryptographic and coding mechanisms

Method	Main algebraic operation	Detection/Integrity role	Relative speed	Main limitation
Hill cipher	Matrix multiplication over Z_26	None by itself	High	Linear key recovery under known plaintext
AES MixColumns	Matrix multiplication over GF(2^8)	Diffusion within encryption	High	Security depends on full round design, not MixColumns alone
Hamming code	Generator and parity-check matrices over GF(2)	Single-error correction	High	Limited correction capacity
Reed-Solomon code	Polynomial algebra over finite fields	Burst-error correction	Medium	More complex decoding
LWE-based KEM	Noisy modular linear equations	Key establishment	Medium	Larger parameters than classical public-key systems

The comparison shows a repeated pattern. Methods with simple linear structure are fast and easy to explain but may be weak as encryption schemes. Methods that add nonlinear operations, structured redundancy, or controlled noise achieve stronger security or reliability at the cost of more complex computation.

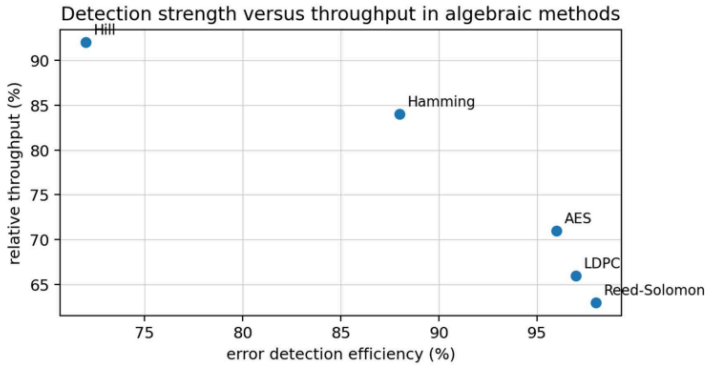

Figure 3.2 Relationship between error detection efficiency and throughput

3.4 Matrix Factorization in Cryptography Security

Matrix factorizations decompose a complicated matrix into simpler components. In numerical linear algebra, decompositions such as LU, QR, and SVD are used to solve systems, approximate matrices, compress data, and improve stability. Their cryptographic role is more specialized. They are useful in implementation, signal processing, image security, watermarking, and some approaches to lattice computation.

3.4.1 Singular Value Decomposition (SVD)

The singular value decomposition writes a matrix A as A = U Sigma V^T, where U and V are orthogonal matrices and Sigma contains the singular values. In image processing and watermarking, SVD separates structural information from intensity or energy information. Modifying selected singular values can embed data or disguise an image while retaining controlled reconstruction properties.

SVD should not be mistaken for a complete encryption method by itself. It is a mathematical tool that can support secure data processing when combined with keys, scrambling, masking, or other cryptographic operations. Its value lies in controlled decomposition and reconstruction.

3.4.2 LU and QR Decomposition

LU decomposition factors a matrix into lower and upper triangular components. QR decomposition factors a matrix into an orthogonal component and an upper triangular component. These methods are important for efficient computation, especially when repeated linear solves are required. In cryptographic contexts, they are relevant where large matrix operations must be optimized without changing the mathematical problem being solved.

3.5 Post-Quantum Linear Algebra-Based Cryptography

Post-quantum cryptography studies schemes intended to remain secure even if large-scale quantum computers become available. Lattice-based cryptography is a leading direction because its underlying problems involve high-dimensional algebra and geometry rather than factoring or discrete logarithms. In 2024, NIST published standards for ML-KEM, ML-DSA, and SLH-DSA; the first two are derived from the CRYSTALS-Kyber and CRYSTALS-Dilithium submissions, respectively.

The Learning with Errors problem is a useful entry point. Choose a public matrix A, a secret vector s, and a small error vector e. Compute

$$b = As + e(modq)$$

The pair (A, b) is public, while s is secret. If e was zero, the system could be solved by linear algebra. The small error vector changes the problem: the equations are nearly linear but not exact, and recovering s becomes computationally difficult for suitable parameters.

3.6 Summary

This chapter showed how linear algebra operates in several cryptographic tasks. The Hill cipher demonstrates reversible matrix encryption but also exposes the danger of pure linearity. Error-correcting codes use matrices to detect and correct corrupted data. Matrix factorizations support efficient computation and selected security applications. LWE-based systems show how high-dimensional modular linear equations, when combined with small errors, become a basis for post-quantum cryptography.

Chapter 4

Noise Resilience and Secured Transmission Protocols

4.1 Preface

A secure channel must preserve both secrecy and correctness. Encryption protects the meaning of data from unauthorized readers, but it does not automatically repair corruption caused by noise, packet loss, interference, or faulty storage. Error correction fills this gap. It adds structured redundancy so that a receiver can recognize and repair certain errors before the message is used.

The algebraic theory of error correction is closely connected to linear algebra. Messages and codewords are vectors over finite fields. Encoding is a linear transformation. Valid codewords form a subspace or a structured set. A parity-check matrix tests membership in that set. These ideas make coding theory a natural companion to cryptographic communication.

4.2 Foundation of Coding Theory

A code is a collection of valid words used for transmission or storage. In a binary linear code, the codewords form a subspace of F_2^n.. If the message has k bits and the codeword has n bits, then the code is called an (n, k) code. The difference n - k represents the number of redundancy bits added for protection.

$$C = \{mG : m \; in \; F_2^k\}.$$

The minimum Hamming distance $d_{\min}$ is the smallest number of positions in which any two distinct codewords differ. This distance measures error tolerance. A code with minimum distance $d_{\min}$ can detect up to $d_{\min} - 1$ errors and correct up to floor $\left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ errors. Thus, distance is the algebraic measure of reliability.

4.3 Linear Codes and Their Structure

4.3.1 Generator Matrix (G)

The generator matrix G defines the encoding rule. A message vector m is multiplied by G to produce a codeword c. When G is in systematic form, the original message bits appear directly in the codeword and the remaining positions are parity bits. This form is useful because it keeps encoding transparent while still adding redundancy.

$$C = mG.$$

For the Hamming (7,4) code, four information bits are expanded to seven transmitted bits. The three additional bits are not arbitrary; they are chosen so that every single-bit error produces a syndrome identifying its position.

4.3.2 Parity-Check Matrix (H)

The parity-check matrix H defines the constraints that every valid codeword must satisfy. For a valid codeword c,

$$Hc^T = 0.$$

The rows of H represent parity relations. If a received vector violates one or more of these relations, the syndrome reveals the violation. This is a linear test and can be performed efficiently.

4.3.3 Error Detection and Correction Process

1. The sender writes the information as a vector m over $GF(2)$.
2. The vector is encoded as $C = mG$.
3. During transmission, noise may add an error vector $r = c + e$.
4. The receiver computes $Hr^T = S$.
5. If $S = 0$, the word satisfies the parity checks. If S is nonzero, it indicates an error pattern.
6. For a correctable error, the receiver flips the identified bit and recovers the original codeword.

4.4 Parity-Check Matrix and Syndrome Decoding

Syndrome decoding is efficient because it avoids comparing the received vector with every possible codeword. Since $r = c + e$,

$SHr^T = H(c + e)^T = Hc^T + He^T = He^T$.

The syndrome depends only on the error vector. In a single-error-correcting code, each column of H corresponds to a possible one-bit error. If the syndrome equals the sixth column of H, the receiver knows that the sixth bit has been corrupted. This is a direct and elegant use of linear algebra.

4.5 Example: Hamming Codes

4.5.1 Hamming (7,4) Code

The Hamming (7,4) code encodes four data bits into seven bits and has minimum distance three. It can correct one error and detect two errors. Let m = [1 0 1 1]. After multiplication by a suitable generator matrix, one possible codeword is c = [1 0 1 1 0 1 0].

If one bit changes during transmission, the received vector r differs from c in exactly one position. The syndrome $Hr^T = S$ identifies that position. After correcting the bit, the receiver obtains the valid codeword and extracts the original message.

Table 4.2 Parameters of Hamming (7,4) and Hamming (15,11) codes

Parameter	Hamming (7,4)	Hamming (15,11)
Message bits k	4	11
Codeword length n	7	15
Parity bits n - k	3	4
Minimum distance $d_{\min}$	3	3

Parameter	Hamming (7,4)	Hamming (15,11)
Detection capability	Up to 2-bit detection	Up to 2-bit detection
Correction capability	1-bit correction	1-bit correction

4.6 Matrix-Based Decoding and Error Correction

Matrix-based decoding makes error correction systematic. The received word is not judged by appearance or by probability alone; it is tested against exact parity equations. This is particularly important when encrypted data are transmitted. A single corrupted bit may cause decryption failure or produce meaningless output. Correcting errors before decryption improves reliability.

In satellite communication, retransmission can be expensive or impossible. In storage systems, fragments may be lost or damaged. In financial communication, corrupted transaction data may cause serious consequences. Linear codes provide a mathematical way to maintain accuracy across these settings.

4.7 Applications in Secure Transmission Channels

Linear error-control methods appear in wireless networks, storage systems, digital payment infrastructure, satellite communication, and defence communication. In each case, the same broad pattern appears: the sender adds structured redundancy, the channel introduces possible errors, and the receiver uses algebraic checks to restore a valid message.

For wireless and cellular systems, codes such as LDPC and Reed-Solomon support reliable high-throughput communication. For cloud storage, polynomial and matrix-based reconstruction methods allow lost fragments to be recovered from surviving fragments. For secure payment and authentication systems, error detection protects the data path so that altered information can be rejected before it enters the cryptographic decision process.

In a combined system, correctness and secrecy can be written in two conditions:

$$Hr^T = 0,\ D_{key}\left(E_{key(m)}\right) = m.$$

The first condition expresses transmission validity; the second expresses cryptographic correctness. A secure communication system must respect both.

4.8 Summary:

Algebraic framework serves as the key mathematical framework in safe transmission systems. On combining the underlying rules of coding methods and secure encryption science, it provides 3 crucial outcomes:

1. Error test matrix: via checking rule matrix

2. Correction of wrong bits: check and fix decoding

3. Protecting the information: through locking the information

These operations guarantee that even under noise, external disturbance, or misuse of data, the forwarded information remains accurate.

Chapter 5

Growth and Future Scope in Matrix-Based Cryptography

5.1 Introduction

The future of public-key cryptography is strongly influenced by quantum computing. RSA and elliptic-curve cryptography rely on number-theoretic problems that are vulnerable to Shor's algorithm on a sufficiently large quantum computer. This does not make all cryptography obsolete, but it does require new public-key systems based on different assumptions.

Matrix-based and vector-based problems are central to this transition. Lattice-based cryptography, code-based cryptography, and related constructions use high-dimensional algebraic structures. Their security is based on the difficulty of recovering hidden information from public equations, noisy samples, or structured codes.

5.2 Post-Quantum Cryptography

Post-quantum cryptography aims to build algorithms that remain secure against both classical and quantum adversaries. Lattice-based methods are prominent because they offer efficient key establishment and signatures with strong mathematical foundations. Code-based cryptography is also important because it rests on different hardness assumptions and can provide diversity against future cryptanalytic breakthroughs.

Table 5.1 Hardness-Based Assumptions and Feasibility Comparison of Post-Quantum Cryptographic Protocols

Protocol	Mathematical Foundation	Hardness Assumption	Advantages	Limitations / Drawbacks
LWE (Learning With Errors)	Random linear equations with small errors (matrix-vector systems)	Solving noisy linear equations is NP-hard, even for quantum computers	High security level, versatile (supports encryption, key exchange, and digital signatures), NIST finalist	Large key size; slower performance compared to RSA/ECC
Ring-LWE	Extension of LWE using polynomial equations in modular rings	Decoding ring-based equations remains computationally difficult	Compact representation; ideal for IoT and lightweight devices due to smaller key sizes	Sensitive parameters; higher implementation complexity
NTRU	Polynomial convolution over arithmetic lattices	Polynomial convolution over arithmetic lattices	Fast encryption and decryption; proven long-term security	Some parameter sets may reduce the security margin
Code-based (McEliece, Niederreiter)	Matrix-based codes using generator matrices	Matrix-based codes using generator matrices	Extremely strong; tested over decades; resistant to most attacks	Very large public keys; impractical for small or resource-limited devices
MQ (Multivariate Quadratic)	Solving multivariate quadratic equations over finite fields	MQ problem is NP-complete	Efficient for digital signatures; mathematically elegant	Newer field; lacks mature proofs and large-scale validation
MAKE (Matrix Action Key	Matrix group actions (semidirect product structures)	Hardness of the matrix action problem	Resistant to Shor's algorithm; innovative algebraic structure	Newer field; lacks mature proofs and large-scale validation

NIST's first finalized post-quantum standards include ML-KEM for key encapsulation, ML-DSA for digital signatures, and SLH-DSA as a hash-based signature scheme. ML-KEM and

ML-DSA are derived from the CRYSTALS-Kyber and CRYSTALS-Dilithium submissions, respectively. This standardization confirms that algebraic problems involving modules, lattices, and finite rings are no longer only theoretical topics; they are becoming part of deployed security infrastructure.

5.3 The Learning with Errors Problem

The Learning with Errors problem can be understood as a noisy system of modular linear equations. Let A be a public m by n matrix over Z_q, let s be a secret vector in Z_q^n, and let e be a small error vector. The public vector is

$$b = As + e(mod q)$$

If e were absent, recovering s from A and b would be a standard linear algebra problem. The error term prevents exact solving. Although each error is small, its presence across many equations makes recovery difficult when parameters are chosen properly. The security is therefore based on the gap between easy forward computation and hard inverse recovery.

This idea is powerful because it is simple to state yet difficult to attack. It also scales naturally to high dimensions and admits efficient variants based on rings and modules.

The public pair (A, b) hides s because the small error e disturbs exact linear solving.

Figure 5.1 LWE instance: public matrix combined with secret vector and small error

5.3 A Ring-LWE and its Connection to Crystals-Kyber

Standard LWE may require large public matrices. Ring-LWE and Module-LWE improve efficiency by replacing general matrices with structured algebraic objects. A common setting uses the quotient ring

$$R_q = \frac{Z_q(x)}{x^n+1}.$$

Elements of this ring are polynomials with coefficients modulo q, reduced by cost. Polynomial multiplication in this ring acts like a structured linear transformation. Efficient algorithms such as the Number Theoretic Transform make these operations fast.

CRYSTALS-Kyber, standardized in renamed form as ML-KEM, is based on module-lattice ideas. The use of structured algebra reduces key size and improves performance while retaining security based on hard lattice problems. This illustrates a key theme of the dissertation: linear algebra becomes more powerful when combined with carefully selected algebraic structure.

Table 5.2 Algebraic Underpinnings and Practical Trade-offs in Classical and Modern Cryptographic Constructions

Mechanism	Algebraic structure	Main purpose	Security idea	Practical issue
Hill cipher	Matrices over Z_2	Classical encryption	Secret invertible matrix	Linear and easy to break
AES diffusion layer	Matrix over $GF(2^8)$	Symmetric block encryption	Diffusion with nonlinear round design	Must be analyzed as full AES
Hamming code	Subspace of $F(2^n)$	Error correction	Syndrome identifies error	Limited to small error patterns
McEliece type systems	Linear codes	Post-quantum public key	Hard decoding of random-looking codes	Large public keys
LWE/Module-LWE	Noisy modular linear equations	KEM/signatures	Hard recovery of secret vector	Parameter and implementation care

Table 5.3 Computational Complexity and Structural Distinctions Between Standard LWE and Ring-LWE Variants

Property	Standard LWE	Ring-LWE
Vector dimension	$O(n^2)$	$O(n)$
Multiplication	$O(n^2)$	O(nlogn)via NTT
Security basis	LWE hardness	RLWE hardness
Used in	Frodo KEM	CRYSTALS-Kyber

5.4 Mathematical Expression Example

A small numerical example clarifies the LWE idea. Let q = 37, choose a public matrix A, a secret vector s, and a small error vector e. The public value b is computed by $As + e(mod37)$. Anyone can see A and b, but recovering s requires dealing with the disturbance introduced by e. In real systems the dimensions and moduli are much larger, and the error distribution is chosen carefully.

The LWE equation: $b = As + e(modq)$

Assuming q = 37

$s = \begin{bmatrix} 4 \\ 9 \\ 7 \end{bmatrix}$ {secret vector}

And public key matrix be A $= \begin{bmatrix} 6 & 11 & 3 \\ 14 & 2 & 8 \\ 5 & 7 & 10 \end{bmatrix}$

And e to be e $= \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$, then learning with error equation becomes:

$$b = As + e(mod 37)$$

The example should not be interpreted as a secure parameter choice. Its purpose is conceptual: exact modular linear algebra is easy to solve, while noisy modular linear algebra can be made cryptographically hard.

5.5 Open Challenges

The first challenge is parameter selection. Stronger parameters improve security but may increase key size, ciphertext size, or computation time. The second challenge is implementation security. Timing leakage, power analysis, fault injection, and poor randomness can damage even mathematically sound schemes. The third challenge is migration. Real systems contain old protocols, stored data, certificates, hardware constraints, and interoperability requirements.

A further challenge is mathematical diversity. Lattice-based schemes are efficient and prominent, but relying on one family alone is risky. Code-based and hash-based schemes provide alternative assumptions. The future of cryptography will likely require a portfolio approach in which different algebraic structures support different security tasks.

Chapter 6

Final Assessment and Directions for Subsequent Work

6.1 Overview of Key Findings

This dissertation examined linear algebra as a connecting framework for cryptography and secure communication. The study began with classical matrix ciphers and moved toward modern finite-field operations, error-correcting codes, and post-quantum cryptography. Across these topics, the same mathematical language repeatedly appeared: vectors represent data, matrices represent transformations, finite fields define legal operations, and subspaces or lattices provide structure.

The Hill cipher demonstrated both the usefulness and the danger of linearity. It is an excellent model for understanding encryption as a reversible transformation, but it fails as a secure cipher because the key can be recovered from enough plaintext-ciphertext pairs. AES showed a more mature design philosophy: linear diffusion is valuable, but it must be combined with nonlinear components and repeated rounds. Coding theory showed that linear algebra also protects reliability, not only secrecy. LWE and Ring-LWE showed how high-dimensional noisy linear problems support post-quantum security.

6.2 Direct Answers to Research Questions

The first research question asked how linear algebra is used in classical cryptography, especially the Hill cipher. The answer is direct: plaintext blocks are vectors, the key is a matrix, encryption is matrix multiplication modulo 26, and decryption requires the modular inverse of the key matrix.

The second question asked what conditions make a Hill cipher key valid. A key matrix K is valid over Z_{26}precisely when $\gcd(\det(k), 26) = 1$. This condition guarantees that K has an inverse $modulo\ 26$ and that decryption is uniquely defined.

The third question asked about the role of modular arithmetic. Modular arithmetic keeps all computations inside a finite symbol set or field. It also determines invertibility and ensures that ciphertext symbols remain valid elements of the chosen domain.

The fourth question concerned the weakness of the Hill cipher. Its weakness is complete linearity. If an attacker obtains enough independent plaintext-ciphertext pairs, the key matrix can be recovered by solving $K = CP^{-1}(mod 26)$.

The fifth question asked how matrix dimension affects security and efficiency. Increasing dimension improves diffusion and increases the amount of data processed per block, but it does not remove the known-plaintext vulnerability. Higher dimension alone cannot replace nonlinear design.

6.3 Unified Mathematical Insight

The unified insight is that cryptographic systems often create a controlled asymmetry between the legitimate user and the adversary. The legitimate user has a key, inverse, secret vector, or decoding structure that makes the problem easy. The adversary sees only public or transformed data and faces a hard inverse problem. Linear algebra helps define both sides of this asymmetry.

In classical ciphers the asymmetry is too weak because the linear map can be learned. In modern block ciphers the asymmetry is strengthened by nonlinear layers and repeated mixing. In post-quantum systems it is strengthened by high dimension, modular structure, and noise.

6.4 Limitations of the Study

The dissertation is theoretical and explanatory. It does not include software implementation, hardware timing tests, or empirical security evaluation. It also does not provide formal reduction proofs for LWE or detailed parameter analysis for standardized schemes. AES is discussed mainly through its linear and finite-field components, not through a full cryptanalytic treatment of all rounds. Side-channel attacks are mentioned as practical concerns but not analyzed in depth.

6.5 Recommendations for Future Research

1. Implement the Hill cipher attack in Python or SageMath to demonstrate complete key recovery from known plaintext.
2. Develop small AES finite-field demonstrations showing MixColumns, inverse MixColumns, and the effect of byte changes.
3. Extend the coding theory chapter with Reed-Solomon and LDPC decoding examples.
4. Study LWE, Ring-LWE, and Module-LWE parameters more rigorously, including error distributions and decryption failure probability.
5. Investigate side-channel resistance in matrix and lattice-based implementations.
6. Compare lattice-based and code-based post-quantum schemes to emphasize mathematical diversity.

6.6 Concluding Remarks

Linear algebra is not merely a background topic in cryptography. It is one of the principal languages through which secure transformation, reliable communication, and post-quantum hardness are expressed. The same ideas that explain a simple classroom cipher also appear, in more refined form, in finite-field diffusion, syndrome decoding, and lattice-based key establishment.

The development of cryptography shows that linear algebra must be used carefully. Pure linearity is usually not enough for secrecy, but structured linear operations remain essential for efficiency, diffusion, correction, and hardness. The future of secure communication will therefore not abandon linear algebra; it will use it in richer algebraic environments, with stronger assumptions and more careful implementation.

REFERENCES

[1] Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.

[2] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

[3] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[4] Hill, L. S. (1929). Cryptography in an algebraic alphabet. The American Mathematical Monthly, 36(6), 306-312.

[5] Lidl, R., & Niederreiter, H. (1997). Finite Fields. Cambridge University Press.

[6] MacWilliams, F. J., & Sloane, N. J. A. (1977). The Theory of Error-Correcting Codes. North-Holland.

[7] Hoffstein, J., Pipher, J., & Silverman, J. H. (2008). An Introduction to Mathematical Cryptography. Springer.

[8] Bernstein, D. J., Buchmann, J., & Dahmen, E. (Eds.). (2009). Post-Quantum Cryptography. Springer.

[9] Lay, D. C., Lay, S. R., & McDonald, J. J. (2016). Linear Algebra and Its Applications. Pearson.

[10] Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography. CRC Press.

[11] Stinson, D. R., & Paterson, M. (2019). Cryptography: Theory and Practice. CRC Press.

[12] Shannon, C. E. (1949). Communication theory of secrecy systems. Bell System Technical Journal, 28(4), 656-715.

[13] Friedberg, S. H., Insel, A. J., & Spence, L. E. (2003). Linear Algebra. Prentice Hall.

[14] Lin, S., & Costello, D. J. (2004). Error Control Coding. Pearson.

[15] Peterson, W. W., & Weldon, E. J. (1972). Error-Correcting Codes. MIT Press.

[16] Goldreich, O. (2004). Foundations of Cryptography, Volume 2. Cambridge University Press.

[17] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES - The Advanced Encryption Standard. Springer.

[18] Dummit, D. S., & Foote, R. M. (2004). Abstract Algebra. Wiley.

[19] Rueppel, R. A. (1986). Analysis and Design of Stream Ciphers. Springer.

[20] Paar, C., & Pelzl, J. (2010). Understanding Cryptography. Springer.

[21] Boneh, D., & Shoup, V. (2020). A Graduate Course in Applied Cryptography.

[22] Gallager, R. G. (1962). Low-density parity-check codes. IRE Transactions on Information Theory, 8(1), 21-28.

[23] Golub, G. H., & Van Loan, C. F. (2013). Matrix Computations. Johns Hopkins University Press.

[24] Hansen, P. C. (1987). The truncated SVD as a method for regularization. BIT Numerical Mathematics, 27, 534-553.

[25] Strang, G. (2016). Introduction to Linear Algebra. Wellesley-Cambridge Press.

[26] Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. Proceedings of FOCS.

[27] Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6), 1-40.

[28] McEliece, R. J. (1978). A public-key cryptosystem based on algebraic coding theory. JPL DSN Progress Report.

[29] Beullens, W. (2020). On the security of multivariate cryptography. Journal of Cryptology, 33, 1-47.

[30] Ducas, L., et al. (2021). CRYSTALS-Kyber: Algorithm specifications and supporting documentation.

[31] Bindel, N., et al. (2019). Hybrid key exchange in TLS 1.3. ACM CCS Workshop on Quantum-Safe Cryptography.

[32] Alagic, G., et al. (2022). Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. NIST IR 8413.

[33] National Institute of Standards and Technology. (2024). FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard.

[34] National Institute of Standards and Technology. (2024). FIPS 204: Module-Lattice-Based Digital Signature Standard.

[35] National Institute of Standards and Technology. (2024). FIPS 205: Stateless Hash-Based Digital Signature Standard.

[36] Peikert, C. (2016). A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science, 10(4), 283-424.

[37] Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In Post-Quantum Cryptography. Springer.

[38] Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. Nature, 549, 188-194.

[39] Bos, J. W., Costello, C., Naehrig, M., & Stebila, D. (2015). Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. IEEE Symposium on Security and Privacy.

MSc Dissertation

ORIGINALITY REPORT

12%	8%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Delhi Technological University Student Paper	4%
2	Submitted to IIT Delhi Student Paper	1%
3	dokumen.pub Internet Source	<1%
4	dspace.dtu.ac.in:8080 Internet Source	<1%
5	www.irjet.net Internet Source	<1%
6	www.numberanalytics.com Internet Source	<1%
7	Submitted to City University of Hong Kong (Dongguan) Student Paper	<1%
8	www.mdpi.com Internet Source	<1%
9	Douglas R. Stinson, Maura B. Paterson. "Cryptography - Theory and Practice", CRC Press, 2018 Publication	<1%
10	Submitted to University of Greenwich Student Paper	<1%
11	Submitted to Southern New Hampshire University - Continuing Education	<1%

Student Paper

12	www.openzeppelin.com Internet Source	<1%
13	Submitted to Hibernia College Student Paper	<1%
14	Junfeng Jia, Yanxun Chang. "Cardinality-consistent flag codes with larger cardinality", Finite Fields and Their Applications, 2026 Publication	<1%
15	weber.itn.liu.se Internet Source	<1%
16	Submitted to Arab Open University Student Paper	<1%
17	mafiadoc.com Internet Source	<1%
18	www.ijraset.com Internet Source	<1%
19	Cyprian Omukhwaya Sakwa, Fagen Li. "Survey on post-quantum cryptography implementations and deployment challenges", Computer Science Review, 2026 Publication	<1%
20	Submitted to University of Sheffield Student Paper	<1%
21	W. Cary Huffman, Jon-Lark Kim, Patrick Solé. "Concise Encyclopedia of Coding Theory", CRC Press, 2021 Publication	<1%
22	"Informatics Engineering and Information Science", Springer Science and Business Media LLC, 2011 Publication	<1%

23	Al-Mahrooqiyah, Sharifa Abdullah Hilal. "Coding Theory with Special Emphasis on Bch Codes", Sultan Qaboos University (Oman), 2025 Publication	<1 %
24	Submitted to University of Lancaster Student Paper	<1 %
25	Ton Duc Thang University Publication	<1 %
26	Submitted to University of Leeds Student Paper	<1 %
27	brightideas.houstontx.gov Internet Source	<1 %
28	Submitted to De La Salle University - Manila Student Paper	<1 %
29	Submitted to University of Southampton Student Paper	<1 %
30	Submitted to Vaagdevi College of Engineering Student Paper	<1 %
31	researchwap.com Internet Source	<1 %
32	security.fudan.edu.cn Internet Source	<1 %
33	www.trademarkelite.com Internet Source	<1 %
34	Mozhgan Mokhtari. "Analysis and Design of Affine and Hill Cipher", Journal of Mathematics Research, 01/30/2012 Publication	<1 %
35	vdoc.pub Internet Source	<1 %

36	www.edfenergy.com Internet Source	<1%
37	Chipman, Damyn. "An Adaptive and Parallel Direct Solver for Elliptic Partial Differential Equations", Boise State University Publication	<1%
38	networkdls.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

MSc Dissertation

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40