



# University of Engineering and Technology, Lahore

Department of Computer Science

---

## Project Report

### Team Members

Name	Roll Number	Email
Shahram Ali Butt	2025-CS-20	shahramalibutt@gmail.com

Session: Fall'25 Morning/Evening      Section: A      Semester: 1<sup>st</sup>

---

Course: CSC101 - Discrete Mathematics

Teacher: Mr. Waqas Ali

January 6, 2026

# Table of Contents

Table of Contents.....	2
Project Report .....	3
1. Project Title .....	3
2. Chosen Track .....	3
3. Overview .....	3
4. Summary .....	3
5. Technology Stack & Libraries Utilized .....	3
6. Problem Statement .....	3
7. Objectives .....	4
8. Mathematical Foundations .....	4
9. Examination of Key Functions/Methods .....	4
10. System Design and Implementation.....	6
11. Business Application Integration .....	6
12. System Features .....	6
13. Results and Testing.....	7
14. Limitations.....	7
16. References.....	8
17. Links.....	8

# Project Report

*Submission Deadline: January 6, 2026*

## 1. Project Title

**Password Security Mechanism, Design and Implementation using Discrete Mathematics**

## 2. Chosen Track

Track 5: Number Theory Applications

## 3. Overview

This project shows the practical application of discrete mathematics concepts, specifically cryptography and number theory, to solve a real-world security problem. The system implements multiple classic encryption algorithms including Caesar cipher, Vigenère cipher, Affine cipher, and a custom hash function to secure user credentials.

The motivation for this project was a business application where passwords were initially stored in plain text format, which was a security vulnerability. By applying cryptographic techniques. The system now encrypts all sensitive data before storage, demonstrating the practical use of Discrete mathematics concepts in computer science.

## 4. Summary

The motivation for this project was a business application where passwords were initially stored in plain text format, which was a security vulnerability. By applying cryptographic techniques. The system now encrypts all sensitive data before storage, demonstrating the practical use of Discrete mathematics concepts in computer science.

## 5. Technology Stack & Libraries Utilized

- **Programming Languages:** C++ 23
- **Libraries:** iostream, fstream, sstream, limits, cmath
- **Development Environment:** gitbash, VSCode

## 6. Problem Statement

In the initial version of a business inventory management application, user credentials were stored insecurely:

- Admin password was hardcoded directly in the source code
- User passwords were stored in plain text in external files
- Anyone with access to the code or files could view all passwords
- No encryption or security measures were implemented

## 7. Objectives

### Primary Objectives:

- Implement multiple classical cryptographic algorithms using discrete mathematics
- Secure password storage through encryption techniques
- Demonstrate practical application of modular arithmetic and number theory
- Create an interactive system for testing different encryption methods

### Secondary Objectives:

- Provide educational demonstrations of each cryptographic method
- Implement password verification without storing plain text
- Integrate encryption into existing business application

## 8. Mathematical Foundations

The project is built upon many concepts from discrete mathematics, specifically from Chapters 4 and 5 of Rosen's textbook:

## 9. Examination of Key Functions/Methods

### 9.1 Caesar Cipher

#### Mathematical Concept:

The Caesar cipher is a substitution cipher that shifts each letter by a fixed number of positions in the alphabet using modular arithmetic.

#### Encryption Formula:

$$E(x) = (x + \text{shift}) \bmod 26$$

#### Decryption Formula:

$$D(x) = (x - \text{shift}) \bmod 26$$

#### Example:

Plain text: HELLO

Shift: 3

Cipher text: KHOOR

### 9.2 Vigenère Cipher

#### Mathematical Concept:

A polyalphabetic substitution cipher that uses a keyword to determine different shifts for each character position.

**Encryption Formula:**

$$E(x) = (x + k[i \bmod |k|]) \bmod 26$$

**Decryption Formula:**

$$D(x) = (x - k[i \bmod |k|]) \bmod 26$$

**Example:**

Plain text: HELLO

Key: KEY

Cipher text: RIJVS

### 9.3 Affine Cipher

**Mathematical Concept:**

Uses two keys (a, b) with modular arithmetic. Requires understanding of coprime numbers and modular multiplicative inverse.

**Encryption Formula:**

$$E(x) = (a \times x + b) \bmod 26$$

**Decryption Formula:**

$$D(x) = a^{-1} \times (x - b) \bmod 26$$

**Requirement:**

$\gcd(a, 26) = 1$  (a must be coprime with 26)

**Valid values for a:**

1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25

### 9.4 Hash Function

**Mathematical Concept:**

A one-way function that maps data of arbitrary size to fixed-size values using modular arithmetic and prime numbers.

**Formula:**

$$\text{hash} = \sum (\text{char}[i] \times \text{prime}^i) \bmod \text{large\_prime}$$

**Properties:**

- Deterministic: Same input always produces same output
- One-way: Cannot reverse the hash to get original password
- Fixed output size regardless of input length

### 9.5 Supporting Mathematical Concepts

**Greatest Common Divisor (GCD):**

Used to verify coprimality in Affine cipher. Implemented using Euclidean algorithm.

**Modular Multiplicative Inverse:**

Essential for Affine cipher decryption. For a and m coprime, find x where  $(a \times x) \bmod m = 1$ .

**Modular Arithmetic:**

Foundation of all cipher operations. Ensures values wrap around within a fixed range (0-25 for alphabet).

## 10. System Design and Implementation

The system consists of three main components:

- **crypto\_functions.h/cpp**: Core cryptographic algorithms and helper functions
- **main.cpp**: Interactive menu system for demonstrations and testing
- **passwordManagement.cpp**: Integration with business application

## 11. Business Application Integration

The passwordManagement.cpp file implements several critical functions for the business application:

- **storeUserDataEnc()**: Encrypts usernames using Affine cipher ( $a=5$ ,  $b=8$ ) and stores with hashed passwords
- **verifyUser()**: Decrypts stored usernames and compares hashed passwords for authentication
- **verifyAdmin()**: Double-hashes admin password for additional security
- **storeSignUpDataEnc()**: Encrypts signup requests using different Affine keys ( $a=9$ ,  $b=15$ )
- **loadSignUpDataDec()**: Decrypts signup data for admin review
- **checkUserExists()**: Verifies if username already exists without storing plain text

## 12. System Features

### 12.1 Interactive Menu System

The main program provides an intuitive menu-driven interface with the following options:

- Individual encryption for each cipher type
- Password hashing functionality
- Cipher combination feature (joining two encryption methods)
- Comprehensive demonstrations of all methods
- Built-in explanations for each cryptographic technique

### 12.2 Cipher Combination Feature

Unique feature allowing users to combine two encryption methods for enhanced security:

- Caesar + Vigenère
- Caesar + Affine
- Vigenère + Affine

This demonstrates how multiple mathematical transformations can be composed for stronger encryption.

### 12.3 Educational Explanations

Each encryption method includes:

- Step-by-step explanation of how it works
- Mathematical formulas
- Practical examples
- Known limitations and vulnerabilities

## 13. Results and Testing

### 13.1 Test Cases and Validation

Comprehensive testing was performed on all cryptographic functions:

Method	Input	Encrypted	Status
Caesar (shift=3)	HelloWorld	KhoorZruog	✓ Pass
Vigenère (key=KEY)	HelloWorld	RijvsUyvjn	✓ Pass
Affine (a=5, b=8)	HelloWorld	RcllaOaplx	✓ Pass
Hash Function	MyPassword123	972352	✓ Pass

All encryption and decryption operations successfully reversed to original text, confirming mathematical correctness.

### 13.2 Business Application Integration Results

The password management system was successfully integrated into the business application with the following results:

- All usernames encrypted with Affine cipher before storage
- All passwords hashed before storage
- Admin password receives double-hashing for extra security
- Successful authentication without storing plain text
- Signup requests encrypted with different keys to segregate data

## 14 Limitations

As an educational project, the system has certain limitations:

- Caesar cipher vulnerable to frequency analysis
- Vigenère cipher can be broken with sufficient ciphertext
- Affine cipher has limited key space
- Hash function not cryptographically secure

## 15. Conclusion

This project demonstrates the role of discrete mathematics in computer security. By implementing multiple cryptography techniques, I transformed an insecure system storing plain text passwords into a secure application.

The system integrates into a business application, proving that theoretical mathematical knowledge directly translates to practical problem-solving in software development. The implementations are educational only and not production-ready.

## 16. References

- Rosen, K. H. (2019). Discrete Mathematics and Its Applications (8th ed.). McGraw-Hill Education. Chapter 4: Number Theory and Cryptography.
- Rosen, K. H. (2019). Discrete Mathematics and Its Applications (8th ed.). McGraw-Hill Education. Chapter 5: Induction and Recursion.
- Tutorial: <https://youtu.be/0ZxqLybIINQ?si=DYgdY62lqGmirxEB>
- Tutorial: <https://www.youtube.com/watch?v=0oP6XLTl2tY>

## 17. Links

GitHub Repository:

[https://github.com/Shahram-ali-butt/Password-Security-Mechanism\\_DM\\_CS101](https://github.com/Shahram-ali-butt/Password-Security-Mechanism_DM_CS101)

Demonstration Video:

[https://www.linkedin.com/posts/shahram-ali-butt-a3253138a\\_encryption-computerscience-mathematics-activity-7415175468787679232-aZXS?utm\\_source=share&utm\\_medium=member\\_desktop&rcm=ACoAAF\\_Um4YBjC\\_GnhMPARbZoBrhVLQkZLVwjz0](https://www.linkedin.com/posts/shahram-ali-butt-a3253138a_encryption-computerscience-mathematics-activity-7415175468787679232-aZXS?utm_source=share&utm_medium=member_desktop&rcm=ACoAAF_Um4YBjC_GnhMPARbZoBrhVLQkZLVwjz0)