# UESTCHN4009: Information Security

# Laboratory Session: Teaching block 1 & 2

Release date: 21st October 2026.

*[Note: In this lab, you will work in a group of two students. You should cooperate proactively and take part in all aspects of the laboratory session including design, planning and execution. The lab should be solved in Matlab. Since you have access to Matlab outside the lab sessions too, we would encourage you to work with your lab partner outside the lab sessions and demonstrate the results in the lab sessions.]*

## Introduction

In this lab you will develop a simple symmetrical encryption and decryption system using **Matlab**. Symmetric encryption is also referred to as conventional encryption or single-key encryption. One of the earliest symmetrical encryption systems is known as Caesar cipher where the plain text was replaced by alphabets 3 positions later. The later versions shifted away from 3 position and instead use any distant positions. So, the general Caesar algorithm can be expressed as:

$$C = \mathrm{E}(k, p) = (p + k) \bmod 26$$

The simplest way of encrypting the plain text is to assign a numerical value to each alphabet as shown in the Figure 1 below and then using the value of k to find the cipher text alphabet.

| a | b | c | d | e | f | g | h | i | j | k | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Figure 1: Numerical equivalent value assignment for each letter.

## Task 1: (20%)

Design a Caesar cipher that will take the given blue coloured italicised text at the end of this document as input. You should copy the text to a notepad/text file and load it as input. The output should be cipher text of the given plain text.

You will need a 'k' value for your encryption. You should use your GUIDs as input to decide 'k' value using following method:

- Step 1: Add GUIDs of both the group members
  (For example: 2429202+2429222= 4858424)
- Step 2: Calculate k which is sum of all the individual numbers and then calculate k_mod= Mod (k,26)
  For example: k= 4+8+5+8+4+2+4=35. So, K_mod=35 mod 26 = 09.

## Task 2: (15%)

The deciphering of Caesar cyphertext is generally done by using brute force method where you cyclically shift the text until you find a legible output (visual inspection). You should write a code for brute force cryptanalysis to get possible set of deciphered text and select the one that is meaningful by visual inspection. What was the k-value you found? Does it match with your value you used for encryption?

## Task 3 (25%)

Please write a MATLAB programme that counts the frequency of the alphabets in both plain text and ciphertext and plot them in the bar chart showing percentage of total frequency.

Frequency analysis is an interesting way of deciphering. In this method, the frequency of alphabet occurring in the ciphertext is used to match them with the frequency of alphabet occurrence in general English text. For example, in English text, alphabet 'e' and 't' appears most frequently. In ciphertext, the pattern will remain nearly the same, but it will be different letter that will occur most frequent. You can use the most occurred alphabet in cipher text and then find the distance of that alphabet from 'e' or 't'. Fortunately, in Caesar cipher the distance remains the same for whole text. So, you just need to find the most frequently occurred alphabet (not all the alphabets' frequencies). If the text

sample is limited, you may need to try second most frequently occurred alphabet too, but this is unlikely.

Does this method work? Compare the bar chart for this ciphertext with the general English language frequency bar chart (refer to the book for bar chart, you do not need to write code for this).

## Task 4 (40%)

This task is about the Playfair cipher. Playfair cipher requires a keyword to generate the Playfair matrix. Figure 2 shows an example of a Playfair matrix for the keyword 'monarchy'.

| | | | | |
|---|---|---|---|---|
| M | O | N | A | R |
| C | H | Y | B | D |
| E | F | G | I/J | K |
| L | P | Q | S | T |
| U | V | W | X | Z |

Figure 2: Playfair matrix for the keyword 'monarchy'.

Write a program that:

1. takes a keyword as input and builds the 5×5 Playfair matrix;
2. encrypts a given plaintext using that matrix;
3. decrypts a given ciphertext using that matrix.

**Your code must follow all of the rules below.**

**Building the matrix**

- Convert the keyword to upper case, remove spaces and any non-letter characters, and treat J as I.
- Remove repeated letters from the keyword, keeping only the first occurrence of each letter.
- Fill the matrix row by row with the reduced keyword, then with the remaining letters of the alphabet (without J) in alphabetical order.

**Preparing the plaintext**

- Convert to upper case, remove spaces, punctuation and digits, and replace J with I.
- Split the text into pairs of letters from left to right. If the two letters of a pair would be the same, insert X after the first letter and continue pairing from the second letter.
- If a single letter is left at the end, add X to complete the last pair.

**Encrypting each pair**

- Both letters in the same row: replace each letter with the letter to its right, wrapping around from the last column to the first.
- Both letters in the same column: replace each letter with the letter below it, wrapping around from the bottom row to the top.
- Otherwise: replace each letter with the letter in its own row and in the column of the other letter.

**Decrypting** reverses these rules (left instead of right, up instead of down). The decrypted output will contain the inserted X letters and I in place of J; you do not need to remove them.

Demonstrate your program by:

- showing the matrix for the keyword ‘monarchy’ and checking that it matches Figure 2;
- encrypting ‘balloon’ with the keyword ‘monarchy’ (you should obtain IB SU PM NA) and decrypting the result;
- showing the matrices for the keyword ‘exceeding’ and for your full name used as the keyword (without space between first name and last name).

**In the lab session, your TA will give you further keywords and plaintexts to test.** Your program must produce the correct result for these during the session, so make sure that it handles every rule above, in particular keywords with repeated letters and plaintexts with repeated letters, the letter J and an odd number of letters.

## Palin text to be encrypted for task 1-3:

*In this section and the next, we examine a sampling of what might be called classical encryption techniques. A study of these techniques enables us to illustrate the basic approaches to symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated. The two basic building blocks of all encryption techniques are substitution and transposition. We examine these in the next two sections. Finally, we discuss a system that combines both substitution and transposition. A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.*

## Marking Criteria:

| **Elements** | **F-E**<br>**20-39** | **D**<br>**40-49** | **C**<br>**50-59** | **B**<br>**60-69** | **A**<br>**70-100** |
|---|---|---|---|---|---|
| **Presentation, clarity and communication of ideas (30%)** | Poor presentation without meaning and lot of irrelevant material | Elementary presentation with some mistakes and some irrelevant material | Satisfactory presentation with meaning and almost no irrelevant material | Good presentation with clear meaning and no irrelevant material | Excellent presentation with very clear meaning and precisely relevant material |
| **Theoretical/ Mathematical /Technical Content (30%)** | Poor or no technical content and analysis without referencing | Elementary technical content and analysis with limited referencing | Satisfactory technical content and analysis with some referencing | Good technical content and analysis with relevant referencing | Excellent technical content and analysis with precisely relevant referencing |
| **Results (40%)** | Poor or no results | Elementary results with limited discussion and comparisons | Satisfactory results with some discussion and comparisons | Good results with critical discussion and comparisons | Excellent results with thorough critical discussion and comparisons |