Lab 01: Breaking the Byte-wise Vigenère (XOR) Cipher
CSE-130-01 **Points: 100**

Overview

In this assignment, you will analyze a ciphertext encrypted using a byte-wise Vigenère (XOR) cipher with an unknown key. Your objective is to determine the key length, recover the key, and decrypt the message using statistical cryptanalysis techniques.

Refer to the lecture slides for:

- The explanation of how the byte-wise Vigenère cipher works.
- The strategy for determining the key length.
- The technique for recovering the key and decrypting the ciphertext.

Getting Started

The given ciphertext is provided in hexadecimal format. You must convert it to raw bytes before performing any analysis.

You may use C (highly preferred), Python , C++ or Java.

Tasks

Step 1: Determine the Key Length

- Use statistical analysis to estimate the key length.
- Refer to the lecture slides for the method to determine the key length.

Step 2: Recover the Key

- Once the key length is determined, use frequency analysis to recover the key bytes.
- Use the approach provided in the lecture slides to deduce the most likely key values.

Step 3: Decrypt the Ciphertext

- Use the recovered key to XOR-decrypt the ciphertext and extract the original plaintext.

Collaboration

You must credit anyone you worked with in any of the following three different ways:

1. Given help to
2. Gotten help from
3. Collaborated with and worked together

Please review the syllabus for details on the collaboration policy. In summary, you may discuss general conceptual approaches with your peers; however, **all code and report must be written independently**. You are strictly prohibited from sharing or receiving code from peers, online sources, or AI-based tools such as Large Language Models (LLMs).

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

1. A well-written short report (1–2 pages max) covering:
   - Key-Length Discovery: Explanation of the method used.
   - Key Recovery Process: How the key bytes were determined.
   - Decrypted Message and Key: Provide the recovered plaintext and key.
   - Reflections: Discuss the weaknesses of repeated-key XOR encryption.
2. A well-documented code implementation for:
   - Key-Length Discovery

- Key Recovery
- Decryption process

Before you hit submit, make sure you have done the following:

- Check that your code compiles and runs on a Linux machine (i.e., without the need for special libraries).
- Attached your code files and report.
- Filled in your collaborator's name (if any) in the "Comments…" text-box at the submission page.

Also, remember to **demonstrate your code to your TA** before the Available Until date (listed in the assignment page) to receive credit for this assignment**. Submissions without a demo will receive a grade of 0**.

Scoring:

- Correctness of Key-Length Discovery: 30pts
- Accuracy of Key Recovery: 30pts
- Code Quality & Documentation: 20pts
- Report Clarity & Explanation: 20pts