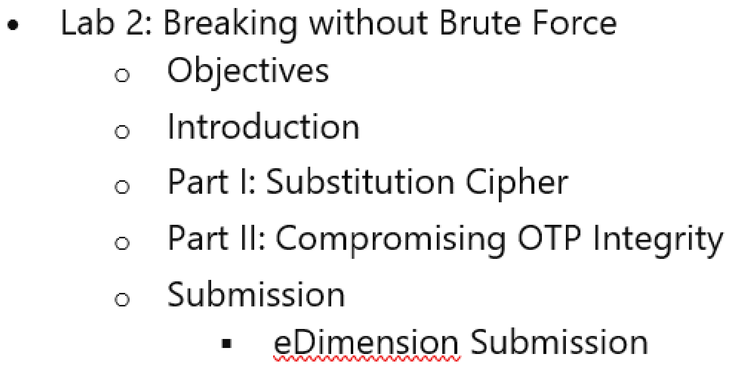
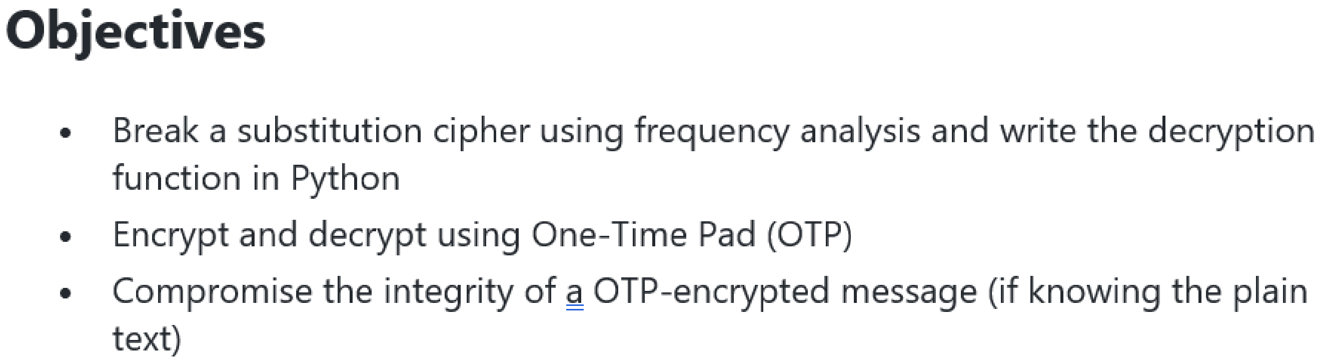
Lab 2: Substitution Cipher (Breaking without Brute Force)

Deadline: 15 June 2025 11:59PM

(From **Lab2_2025.pdf)**



- Lab 2: Breaking without Brute Force
    - Objectives
    - Introduction
    - Part I: Substitution Cipher
    - Part II: Compromising OTP Integrity
    - Submission
        - eDimension Submission



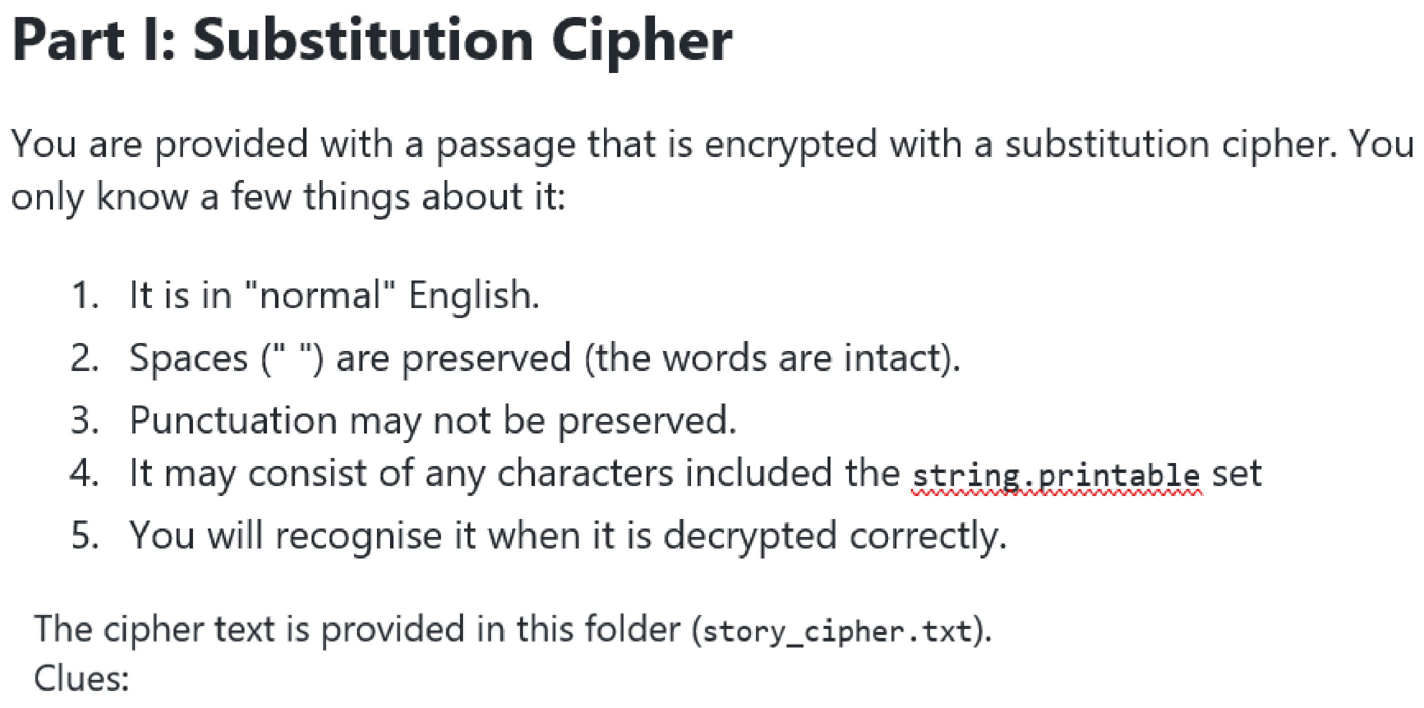
Objectives

- Break a substitution cipher using frequency analysis and write the decryption function in Python
- Encrypt and decrypt using One-Time Pad (OTP)
- Compromise the integrity of a OTP-encrypted message (if knowing the plain text)

Introduction

In this lab, you will be

- breaking a **substitution cipher** via frequency analysis. You are **not** allowed to use other methods to break the cipher.
- manipulating an OTP encrypted message so that it decrypts to a message of your choosing



Part I: Substitution Cipher

You are provided with a passage that is encrypted with a substitution cipher. You only know a few things about it:

1. It is in "normal" English.
2. Spaces (" ") are preserved (the words are intact).
3. Punctuation may not be preserved.
4. It may consist of any characters included the `string.printable` set
5. You will recognise it when it is decrypted correctly.

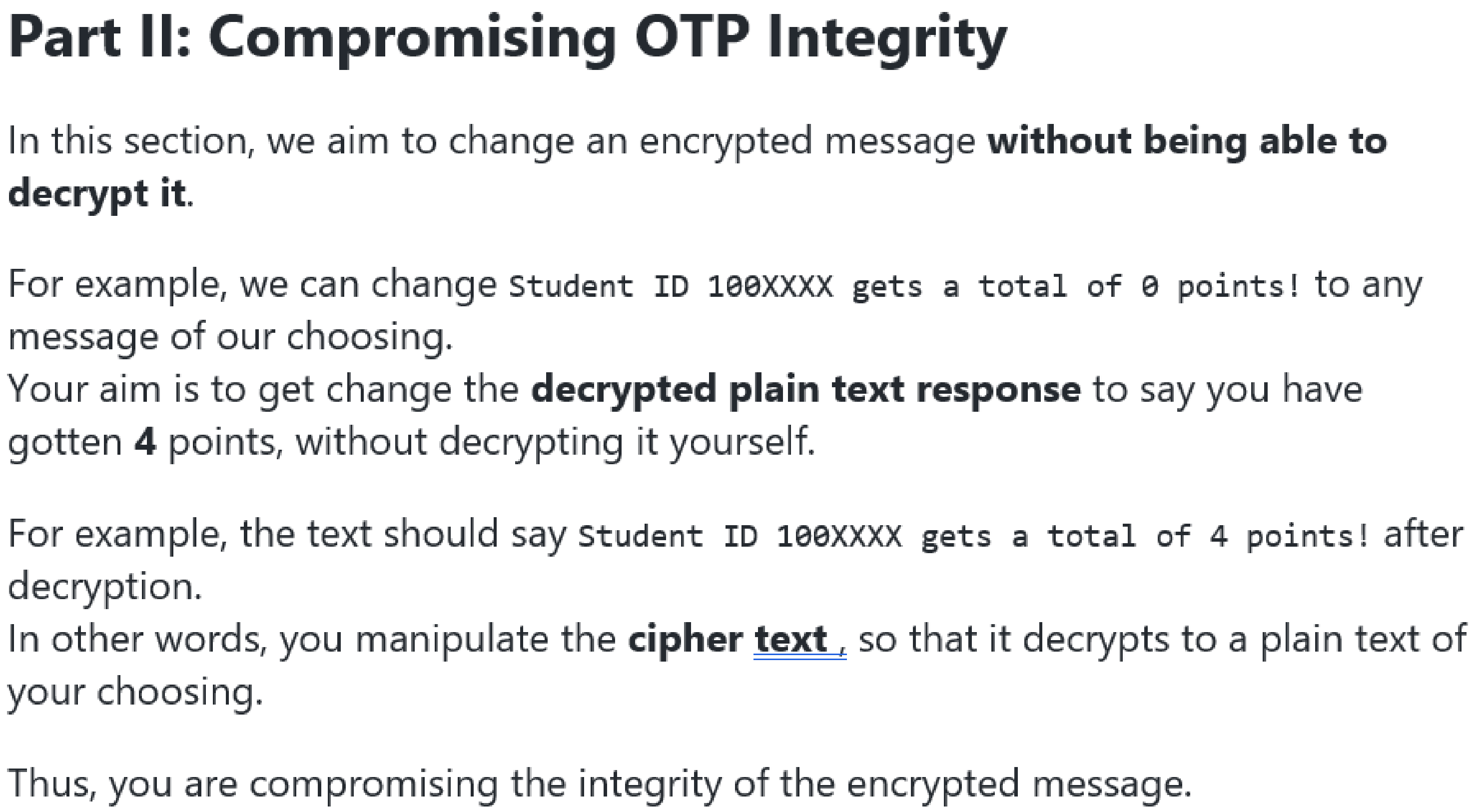
The cipher text is provided in this folder (`story_cipher.txt`).
Clues:

- Wikipedia: Frequency analysis
- Hints for Frequency Analysis
- The frequency of the letters of the alphabet in English Dictionary
- SAS: The frequency of letters in an English corpus

Practical hints:

- You can use Python's string `replace` function.
- If you are stuck halfway, visually inspect your current cipher, and see if you recognise any words that are only partially decrypted.
- Make sure you keep track of the replacements to ensure you do not "double replace". All the characters in the cipher are upper-case by design to make it easier for you. You can gradually replace them with your hypothesis of the correct lower-case characters and visually inspect the result.

Write a Python script to decrypt the cipher text, and submit it together with your decrypted plain text.



Part II: Compromising OTP Integrity

In this section, we aim to change an encrypted message **without being able to decrypt it**.

For example, we can change `Student ID 100XXXX gets a total of 0 points!` to any message of our choosing.
Your aim is to get change the **decrypted plain text response** to say you have gotten **4** points, without decrypting it yourself.

For example, the text should say `Student ID 100XXXX gets a total of 4 points!` after decryption.
In other words, you manipulate the **cipher text ,** so that it decrypts to a plain text of your choosing.

Thus, you are compromising the integrity of the encrypted message.

Hints:

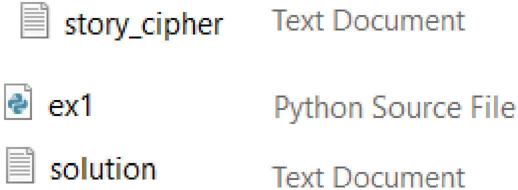
- The ciphertext is encrypted with a OTP. You do not know what the OTP is, it is randomly generated.
- **You do not need to know anything about the OTP for this exercise.**

You are provided with `ex2.py` in this folder. Complete it to show that you can change the encrypted message without knowledge of the OTP.

Part I: Substitution Cipher

Essential file: story_cipher.txt
Expected result files: ex1.py

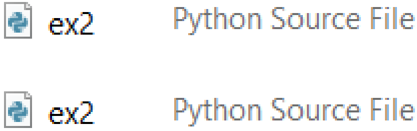
solution.txt



story_cipher Text Document
ex1 Python Source File
solution Text Document

Part II: Compromising OTP Integrity

Essential file: ex2.py
Expected result file: ex2.py



ex2 Python Source File
ex2 Python Source File

Submission

eDimension Submission

Submission rules:

- Please rename the file to: **lab2_name_studentid**

Lab 2 submission:

Upload a **zip file** with the following:

- **`ex1.py`** `(No skeleton code provided)`, your python script to perform decryption of the substitution cipher (`story_cipher.txt`)
- Decrypted plain text for Part I as **`solution.txt`**
- **`ex2.py`**, your python script to change the OTP message, you can base it on `ex2.py`
- Jupyter Notebook report (with the outputs saved) in (.ipynb) or (.pdf)
- Please do not change the names. The names must be as listed above.

Deadline: 15 June 2025 11:59PM