Assignment 2

CS-UH-3210: Computer Security

Due: Wednesday, Feb 8, by 11:59pm GST

Upload your solution as one PDF file to Brightspace with your name on the first page (only). The file name must contain your full name: `Ass-2_<FirstNLastN>.pdf`. Code needs to be submitted as a separate file (you may include short code snippets in your PDF). If your code consists of multiple files, zip all files and upload one zip archive file only, but use directory/file names that clearly indicate the task tackled.

Expectation:

- **You are expected to provide your own, individual solution unless a task explicitly allows or encourages group work. If you use any external resources, name them.**
- **In case of a task are still asked to submit the (possibly identical) solution as part of your own hand-in, but all group members need to specify who they collaborated with in their solution. Failure to do so will make us assume that solutions were copied instead of being collaboratively developed.**
- **The tasks are of different difficulty level. You may be struggling to complete the challenging one(s) – in that case explain until where you got, what you tried, and what were the difficulties you were facing.**

Task 1. Identify Ciphers [5 points]

The task of this exercise is to identify which type of encryption was used for encrypting messages. The possible types are: Shift cipher, Caesar's Cipher, Substitution Cipher, and Vigenère Cipher. For each pair of plaintext and ciphertext (a) determine which method of encryption was used, (b) explain why, and (c) derive the key that was used for this method. Multiple correct answers may be possible.

- **Plaintext**: `ILOVECOMPUTERSECURITYCOURSE`
  Ciphertext: `VYBIRPBZCHGREFRPHEVGLPBHEFR`
- **Plaintext**: `THISISNOTASECRETMESSAGE`
  Ciphertext: `GSRHRHMLGZHVXIVGNVHHZTV`
- **Plaintext**: `ITWASTHEBESTOFTIMES`
  Ciphertext: `DXYTGKCIDYGKJJVBAWN`

Task 2. Substitution Ciphers [5 points]

The ciphertext below was encrypted using a substitution cipher. What was the plaintext? Decrypt the ciphertext without knowledge of the key.

```
HQT VJQDUCPKU QH AGCTU BOPIU SDGGPU CPK IGPGTCNU JCXG
TGNOGK QP GHHOEOGPV EQFFDPOECVOQP OP QTKGT VQ IQXGTP VJGOT
EQDPVTOGU CPK EQFFCPK VJGOT CTFOGU CV VJG UCFG VOFG VJGA
JCXG CNN WGGP CYCTG QH VJG EQPUGSDGPEGU QH VJGOT FGUUCIGU
HCNNOPI OPVQ VJG YTQPI JCPKU TGXGCNOPI RTGEOQDU UGETGVU VQ
TOXCN PCVOQPU CPK WGVTCAOPI XOVCN OPHQTFCVOQP VQ QRRQUOPI
```

```
HQTEGU OV YCU VJG VJTGCV QH GPGFA OPVGTEGRVOQP VJCV
FQVOXCVGK VJG KGXGNQRFGPV QH EQKGU CPK EORJGTU VGEJPOSDGU
HQT KOUIDOUOPI C FGUUCIG UQ VJCV QPNA VJG OPVGPKGK
TGEOROGPV ECP TGCK OV
```

You can find a `sub-cipher.txt`-file with the ciphertext provided on Brightspace. Note that the text is relatively short and the letter frequencies in it might not perfectly align with that of general English language from a letter frequency table. Do not use an existing tool/webpage to solve the problem but instead describe your approach for decrypting the ciphertext. If you write code (recommended), append it to your solution. Answer the following questions:

1. How did you approach and solve this task? Be sufficiently detailed.
2. What does the plaintext say?
3. Who wrote the text?

Task 3. One-time Pad [5 points]

You may collaborate in teams of 2 for this task. Self-organize your collaboration, but do reach out if you have problems finding a partner. You may also use the Brightspace Forum Function to find a collaboration partner.

It is well known that re-using a one-time pad can be insecure. This task explores this issue. In this task, all characters are represented as 8-bit bytes with the usual US-ASCII encoding (e.g., “A” is encoded as `0x41`). The bitwise exclusive-or of two bytes x and y is denoted $x \oplus y$. Let $M = (m_1, m_2, \ldots, m_n)$ be a message to be encrypted, consisting of a sequence of n message bytes. Let $K = (k_1, k_2, \ldots, k_n)$ denote the key, consisting of a corresponding sequence of (randomly chosen) key bytes. In the usual one-time pad, the sequence $C = (c_1, c_2, \ldots, c_n)$ of ciphertext bytes is obtained by xor-ing each message byte with the corresponding key byte:

$$c_i = m_i \oplus k_i, \text{ for } i = 1 \ldots n.$$

To refer to more than one message, we will denote the messages as M_1, M_2, $\ldots$, M_k and the bytes of message M_j as m_{ji}, namely $M_j = (m_{j1}, \ldots, m_{jn})$; we will also use similar notation for the corresponding ciphertexts.

(a) Here are two 8-character English words M_1 and M_2 encrypted with the same “one-time pad”. What are the words?

```
e9 3a e9 c5 fc 73 55 d5
f4 3a fe c7 e1 68 4a df
```

First analyze the problem analytically. Then you may want to write code to help your investigation, using a word dictionary may be useful. Give a detailed explanation how you arrived at your solution.

(b) *[Challenging: You may get full points even if you do not fully solve this task – Try to get as far as you can]*
Ash Supersafe decided to tackle the above problem by ensuring that one cannot simply cancel key bytes in the one-time pad by xor-ing the ciphertext bytes. In their scheme the key is still as long as the ciphertext. If we define $c_0 = 0$ for notational convenience, then the ciphertext bytes $c_1, c_2, \ldots, c_n$ are obtained as follows:

$$c_i = m_i \oplus ((k_i + c_{i-1}) \mod 256).$$

That is, each ciphertext byte is added to the next key byte and the addition result (modulo 256) is used to encrypt to the next plaintext byte.

Ash is now confident they can reuse their pad, since $(k_i + c_{i-1}) \mod 256$ will be different for different messages, so nobody would be able to cancel the k_i's output.

You are provided with `otp-feedback.py`, which contains an implementation of Ash's algorithm. You are also given the file `tenciphers.txt`, containing ten ciphertexts $C_1, C_2, \ldots, C_{10}$ produced by Ash, using the same one-time pad. All messages consist of valid English text.

Your task is to provide the plaintext messages and the pad (the key), along with a careful explanation of how you found them, and any code you used to help find the messages. The most important part is the explanation.