Introduction

Bookmark this page

Encryption is the process of obscuring information to make it unreadable without special knowledge. For centuries, people have devised schemes to encrypt messages - some better than others - but the advent of the computer and the Internet revolutionized the field. These days, it's hard not to encounter some sort of encryption, whether you are buying something online or logging into a shared computer system. Encryption lets you share information with other trusted people, without fear of disclosure.

A cipher is an algorithm for performing encryption (and the reverse, decryption). The original information is called plaintext. After it is encrypted, it is called ciphertext. The ciphertext message contains all the information of the plaintext message, but it is not in a format readable by a human or computer without the proper mechanism to decrypt it; it should resemble random gibberish to those for whom it is not intended.

A cipher usually depends on a piece of auxiliary information, called a key. The key is incorporated into the encryption process; the same plaintext encrypted with two different keys should have two different ciphertexts. Without the key, it should be difficult to decrypt the resulting ciphertext into readable plaintext.

This assignment will deal with a well-known (though not very secure) encryption method called the Caesar cipher. Some vocabulary to get you started on this problem:

- *Encryption* - the process of obscuring or encoding messages to make them unreadable until they are decrypted
- *Decryption* - making encrypted messages readable again by decoding them
- *Cipher* - algorithm for performing encryption and decryption

- *Plaintext* - the original message
- *Ciphertext* - the encrypted message. Note: a ciphertext still contains all of the original message information, even if it looks like gibberish.

The Caesar Cipher

The idea of the Caesar Cipher is to pick an integer and shift every letter of your message by that integer. In other words, suppose the shift is k . Then, all instances of the i-th letter of the alphabet that appear in the plaintext should become the (i+k)-th letter of the alphabet in the ciphertext. You will need to be careful with the case in which i + k > 26 (the length of the alphabet). Here is what the whole alphabet looks like shifted three spots to the right:

```
Original:  a b c d e f g h i j k l m n o p q r s t u v w x y z
 3-shift:  d e f g h i j k l m n o p q r s t u v w x y z a b c
```

Using the above key, we can quickly translate the message "happy" to "kdssb" (note how the 3-shifted alphabet wraps around at the end, so x -> a, y -> b, and z -> c).

Note!! We are using the English alphabet for this problem - that is, the following letters in the following order:

```
>>> import string
>>> print string.ascii_lowercase
abcdefghijklmnopqrstuvwxyz
```

We will treat uppercase and lowercase letters individually, so that uppercase letters are always mapped to an uppercase letter, and lowercase letters are always mapped to a lowercase letter. If an uppercase letter maps to "A", then the same lowercase letter should map to "a". Punctuation and spaces should be retained and not changed. For example, a plaintext message with a comma should have a corresponding ciphertext with a comma in the same position.

```
|    plaintext     |  shift     |  ciphertext       |
| -----------------|------------|-------------------|
| 'abcdef'         |    2       |  'cdefgh'         |
| 'Hello, World!'  |    5       |  'Mjqqt, Btwqi!'  |
| ''               | any value  |  ''               |
```

We implemented for you two helper functions: `load_words` and `is_word`. You may use these in your solution and you do not need to understand them completely, but should read the associated comments. You should read and understand the helper code in the rest of the file and use it to guide your solutions.

Getting Started

To get started, download the ps6.zip file. Extract it to your working directory. The files inside are:

- ps6.py - a file containing three classes that you will have to implement.
- words.txt - a file containing valid English words (should be in the same folder as your ps6..py file).
- story.txt - a file containing an encrypted message that you will have to decode (should be in the same folder as your ps6..py file).

This will be your first experience coding with classes! We will have a `Message` class with two subclasses `PlaintextMessage` and `CiphertextMessage` .

Introduction

Topic: Problem Set 5 / Topic-Level Student-Visible Label

Show Discussion