IMT4113 Assignment 1

Ole André Hauge

September 13, 2020

The assignment requested an implementation of a switch-case, with additional information of the look and feel. As a result, not wanting to *"miss"* the design target, the code in this assignment utilizes a fall through design of the switch-case. Meaning that the user cannot go back and forth between the different levels. They have to start the program again each time. In addition, a try-catch statement has been implemented to surround the switch-case in order to catch invalid values entered by the user at the various stages.

Furthermore, every implemented cipher assumes that the result of encryption\decryption should not contain spaces. This can be implemented, however since the assignment examples did not use spaces, this code does not.

No error explanation is available for the person decrypting the messages. This is due to security. Giving an adversary information about what we are looking for while executing the code is viewed as bad practice.

Furthermore, all ciphers are based on the English alphabet. Implementing other alphabets is possible.

Route Cipher Inputs

Takes *plaintext (str)*, *route key (str)* and *grid dimesions (str)* as inputs. Dimensions need to be seperated by "*".
Two routes are implemented: *Horisontal route from top-left* and *Vertical route from top-right*.

Tested using *"We are discovered flee at once"* as plaintext, *3*9* as matrix dimensions, with both of the keys. The decrypted ciphertext contains the padding used during the encryption *(X's)*. This is intentional and by design.

Vigenère cipher Inputs

Takes *plaintext (str)* and *key (str)* as input.

Tested with *"vigene"* as plaintext and *"crypto"* as key, in addition to several Lorem Ipsum (`www.lipsum.com`) texts to test the performance on longer texts. It was tested for a key longer than the plaintext. If this happens the code will tell the user to use a smaller key. However, a smaller key is accepted.

Four Square Cipher Inputs

Takes *plaintext (str)* and a *key (str)* containing two keys divided by *","* as input.

Tested with *"test plaintext"* as plaintext and *"EXAMPL, KEYWORD"* as keys. The keys are run through a *unique* function to ensure that the ciphertext matrices do not contain more than one instance of a letter.

Breaking Caesar Cipher Using Frequency Analysis Inputs

Takes a *text (str)* as primary input. Secondary input (users estimate) is the users *key (int)* estimate.

The code then decodes the ciphertext using the estimated key.

Tested with several texts form Lorem Ipsum (`www.lipsum.com`), which are not included in this document due to the length limitation of this document. The texts were encrypted using an online caesar cipher encrypter, before used as inputs for the frequency analysis code.