

COMP 7402 Assignment 2 Report

Dimitry Rakhlei

Goal

Our goal was to break the transposition cipher by continuously comparing decoded text to a dictionary for each possible key and testing if it matches a high enough threshold of words. This should give us the ability to break the cipher rather easily.

Design

The program was rather simple. Modules for decryption and encryption were provided as well as the module used for detecting languages in text. Using these three modules a new module called *"break_transposition.py"* was constructed. This module has three functions:

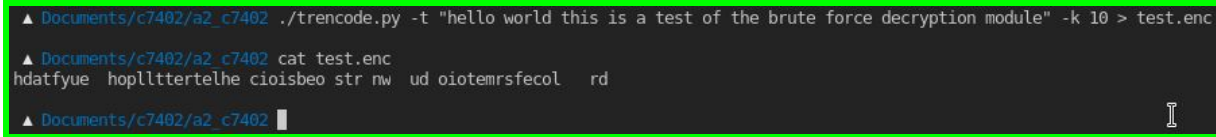
1. `main()` - Its purpose is to parse all of the arguments and facilitate the correct execution of the application.
2. `attempt_decode()` - This function attempts to decode the ciphertext with one key and returns True or False based on if it thinks that the plaintext is in English or not.
3. `break_cipher()` - This function calls `attempt_decode()` anywhere from 1 to `len(ciphertext)` times unless a `maxsize` argument is passed. Then it stops at `min(len(ciphertext), maxsize)`.

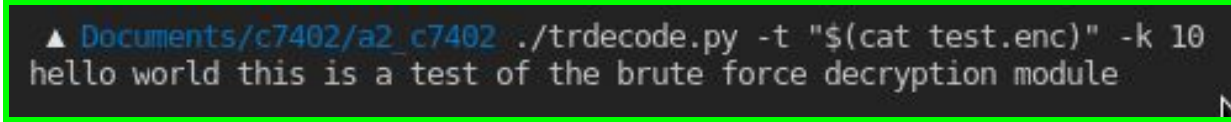
Usage

The module recognized `-h` as a command and will print usage instruction if needed. It is very simple to operate overall. A user can simply pass arguments such as:

- `-c` for ciphertext to be decoded
- `-m` for maximum key size
- `-t <lowthresh>:<highthresh>` for the detection thresholds
- `-i` for quick interactive mode which prompts for inputs instead

Testing

Test # 1	
Command	Expected output
<code>./tencode.py -t "hello world this is a test of the brute force decryption module" -k 10 > test.enc</code>	Test.enc is created with encrypted text
 <pre>▲ Documents/c7402/a2_c7402 ./tencode.py -t "hello world this is a test of the brute force decryption module" -k 10 > test.enc ▲ Documents/c7402/a2_c7402 cat test.enc hdatfyue hoplltttertelhe cioisbeo str nw ud oiotemrsfecol rd ▲ Documents/c7402/a2_c7402</pre>	

Test # 2	
Command	Expected output
<code>./trdecode.py -t "\$(cat test.enc)" -k 10</code>	The original text is output back to us. This is important because we want to know that the module can correctly decode an encoded string given the right key.
 <pre>▲ Documents/c7402/a2_c7402 ./trdecode.py -t "\$(cat test.enc)" -k 10 hello world this is a test of the brute force decryption module</pre>	

Test # 3	
Command	Expected output
<pre>./break_transposition.py -c "\$(cat test.enc)" -m 100 -t 10:90</pre>	<p>The program attempts to decode the message and prompts the user for confirmation if it finds something that fits under the threshold specified. If the user confirms the decoded message then the program exits.</p>
<pre> ▲ Documents/c7402/a2_c7402 ./break_transposition.py -c "\$(cat test.enc)" -m 100 -t 10:90 (10, 90) Is this the decoded message? ["hellitueod thsrdmla teb r the enos forcowif ypti oeruleos tcd"] with a key size of: 9 [y/n]>n n Trying Again Is this the decoded message? ["hello world this is a test of the brute force decryption module"] with a key size of: 10 [y/n]>y y </pre>	

Test # 4	
Command	Expected output
<code>./break_transposition.py -i</code>	The program prompts the user for the max key size and ciphertext and attempts to break it as before. This is just the interactive mode.
<pre>▲ Documents/c7402/a2_c7402 ./break_transposition.py -i Enter max size: 128 Enter cipher text: hdatfyue hoplltttertelhe cioisbeo str nw ud oiotemrsfecol rd Is this the decoded message? ["hellitueod thsrdmla teb r the enos forcowif ypti oeruleos tcd"] with a key size of: 9 [y/n]>n n Trying Again Is this the decoded message? ["hello world this is a test of the brute force decryption module"] with a key size of: 10 [y/n]>y y</pre>	