MAT 331 Fall 2017, Project 9
Breaking a re-used one time pad

This is a challenging project; it would be impressive to do this without a hint (even with the hint).

Suppose we encrypt a message by shifting the nth letter by a random amount s_n (the nth shift):

$$c_n = (p_n + s_n) \mod 26,$$

where p_n is the nth plain text letter and c_n is the nth code letter. This is very secure if the sequence $\{s_n\}$ is really random and is never reused; this is called a one-time pad.

However, this is breakable if we encrypt two messages using the same sequence $\{s_n\}$, say

$$d_n = (q_n + s_n) \mod 26,$$

where q_n is the nth plain text letter in the second message and and d_n is $n$the letter of the second coded message. Then

$$(c_n - d_n) \mod 26 = (p_n + s_n - q_n - s_n) \mod 26 = (p_n - q_n) \mod 26,$$

and we can compute this from the two coded messages.

(1) I have encoded two messages using the program `rs2.m` but the same seed for the random number generator. They are called `plain11.rs2` and `plain2.rs2` and are stored in the directory
`http://www.math.stonybrook.edu/~bishop/classes/math331.F17/Crypt/`
Retrieve these files and do a letter count on them. Do they look random?

(2) Read in the two files and compute the difference $t_n = c_n - d_n \mod 26 = p_n - q_n \mod 26$. Does this look random? Explain.

(3) Choose a word and convert it numbers modulo 26. Add it to t_n at different locations. If that word occurs in that position in the second message, then adding it cancels the $-q_n$ and leaves just p_n, which should look like a English word (or part of a word, or a combination of parts of adjacent words). Longer words or proper names will reveal a long stretch of code. We can test how much this looks like English using the `count_pairs.m` program and then sort, to find the most likely possibilities, which we can then examine by eye. If we see one that looks like English, then we keep it and repeat the procedure. Some code that I wrote to do this is `testword2.m`. This code prompts you to enter words that you want to test, but you may wish to modify it to use words chosen from a file, such as `wordlist.txt`, that contains over 50,000 English words.

(4) Decrypt the two encrypted files `plain1.rs2` and `plain2.rs2`. If you want a hint, email me and I will tell you the books from which each passage comes.