Eric Vance

Homework 1  
Comp 424

**Problem**:

Decrypt the intercepted code:

DRPWPWXHDRDKDUBKIHQVQRIKPGWOVOESWPKPVOBBDVVVDXSURWRLUEBKOLVHIHBKHLHBLNDQRFLOQ

**Solution**:

When analyzing the problem, I decided there were two methods that I could use:

1. Blind brute force
2. Heuristics - Analysis combined with brute force.

I decided that the second choice was how I was going to approach the problem and as a backup use brute force.

Resources (written in Python 2.7):

The python library Pycipher was used for shift substitution and columnar transposition decryption. A method (see infer\_spaces.py) to find words from a dictionary in a string with no spaces and then separate with spaces was taken from Stacked Overflow (see comment in code for source link).

**Simple Shift Attack**:

Key information was given about the string that allowed me to analyze and correctly guess the key for the simple shift substitution. The given information is the message is in English and contains only letters. Knowing this information, I used frequency analysis to try and find a possible pattern for the simple shift substitution key. In python I wrote code to count how often a given letter appeared in the string. From that list I was able to identify that the characters d, v, b, h, k, and r were the top 6 characters that appeared the most in the string. I then wrote code to compare the character I chose to the top 5 most frequent letters in the English language; a, s, e, t, and o. I was able to find a pattern, ‘d’ was 3 positions from ‘a’, ‘v’ was 3 positions from ‘s’, ‘h’ was 3 positions from ‘e’, and ‘r’ was 3 positions from ‘o’. At this point I was sure the key for the simple shift was 3.

**Columnar Transposition Attack**:

This was the harder part of the problem. After finding a possible simple shift, I had to decide if I would try to brute force the possible key combinations or attempt more analytics to reduce the time cost. Knowing that the key length is less than or equal to 10 letters long, it was quite possible to brute force the 10! + 9! + … 1! combinations. However, it could be approached in a more manageable manner. By using frequency analysis with the top common digrams I could find more probable lengths of the key instead of trying them all. I wrote code to analyze the string, after the simple shift, for possible key lengths based on the digram frequencies. Using the top 30 digrams, I kept track of digram frequencies of only the distances from 1 to 10; most frequent key length to the least: 2, 6, 8, 5, 3, 4, 7, 9, 10, 1. My program would prompt me to try key lengths one by one. By using this method, I could stop if I found a solution quickly and at worst I would try all the keys.

**Final Solution:**

A major obstacle for my solution to work was to separate words that appeared in the string after shifting and columnar transposing. I used a method that I found on Stack Overflow find words that existed in a dictionary and separate them with spaces. On my first approach, I had issues with dictionaries that I found on the internet. They contained to many words and found “words” prematurely. By using the /usr/share/dict/words from my Ubuntu Linux system, I was able to provide an extensive list of words but avoid the obscure that would throw off the results. This process worked great but I was still flooded with many incorrect results. I decided it would be better to filter these using my program rather than to manually check after each key was tried. The false readings were leaving strings with one letter words. To discard these, I would not write plaintext results to the soultions.txt file if more than 10 one letter words were found. I felt it was safe that no solution contained more than 10 one letter words. Once I did this, and ran my program with the first 3 keys, 2, 6, and 8, I received no results. I was sure that the shift pattern of 3 was correct so I planned to try all of the keys before I gave up on the simple shift key. Once I chose the key length of 5, I received 3 possible decryptions in my solutions.txt:

1. e happy forth e moment this moment is your life by k hay yam oh and also this class is really fun b
2. be happy forth e moment this moment is your life by k hay yam oh and also this class is really fun
3. a p p y forth e moment this moment is your life by k hay yam oh and also this class is really fun b he

The second solution had to be correct, but something weird happening in the middle; “k hay yam” was showing up and I was not sure what that was. So putting the word fragments into google, I found out it was the name of a Persian mathematician, astronomer, philosopher and poet, Omar Khayyam. After adding khayyam to the dictionary and manually fixing the issue that was happening with “forth e”, to “for the”, the final solution and key was:

**Key**: 35214

**Plaintext**: be happy for the moment this moment is your life by khayyam oh and also this class is really fun