Patrick Whitsell, Casey McGinley, Fernando Maymi

Cryptanalysis

We are tasked with decrypting an alphabetic substitution cipher. The encryption scheme maps each letter to a set of numbers. The more frequent a letter is on average, the more numbers it will map to, so that the cipher is resistant to frequency analysis. We do have an advantage in the fact that white space is not encrypted which allows us to determine the length of each word in the original plaintext. We will use this advantage in each of the tests to find a possible key value.

For the first test, we are given all the possible plaintext values. We use this information, along with our advantage with respect to the ciphertext, to find similar word length patterns between our ciphertext and plaintext candidates in the following way. Given some plaintext candidates each we can iterate through the encrypted words in our ciphertext, comparing its length to the lengths of the corresponding plaintext words (or put another way, we compare the positions of the spaces). When the lengths do not match, we can safely eliminate that plaintext candidate from the list. Even assuming a very large number of candidates this strategy could eliminate enough candidates to make even a brute force approach effective.

A little bit of quick statistics will show us that the most likely length (8) consists of about 1/5 of words in the provided word list. Assuming the plaintext is 5 words (L = 40, much less than the L=500 in the actual problem) we find that we have a 1/3125 chance of getting a plaintext that consists of five eight-letter words. (It is important to note that this is the most likely result since eight-letter words are independently the most likely word length and the plaintext is randomly selected.) This means that the chance of any random, five word ciphertext matching an unrelated plaintext in word length pattern is no more than 0.032%. The longer the ciphertext, the larger the plaintext space needs to be to render this approach insufficient on its own.

Fortunately, we did not have a large number of plaintext candidates, only five, each with a first word of a different length. In this case the attack can be as simple as a few if statements comparing first word lengths. The decryption scheme for the second test can also be used successfully to complete this test

In the second test the encryption scheme is the same but the plaintext is generated by randomly selecting words from a large list of words. We have the list of words and we know the length of the words in the ciphertext. Our approach was to separate the list of words by length, and then use that smaller space to try to guess the value of words in the ciphertext. With each guess we add values to the key and reduce the space for the next guess. By guessing values for the longest words in the ciphertext first, the number of guesses to be made is reduced while filling up the key faster. We pick our guesses by scoring the candidate plaintext words by letter frequency and then picking the lowest scoring words first. This potentially allows us to focus our unguessed values into few letters, further reducing the space for guesses.

All theory was a collaborative effort. Specific implementation of part 1 by Patrick Whitsell. Implementation of part 2 by Casey McGinley with help from Fernando Maymi. Report by Fernando Maymi.