COMP6370 – CA 01: How to break this cipher?

Song Gao <szg0031>

September 21, 2014

This cipher is a substitution cipher. And it's essentially a modified Vigenere Cipher. The difference is that, rather than simply shifting the cleartext character by the corresponding key value, this cipher shifts the character by the corresponding key value plus previous encrypted character's index. This adds another layer of security, but this added layer is reversible. Since the extra shift is from previous ciphertext character rather than cleartext character, the extra value being shifted is directly available in the ciphertext. After reversing the extra shift, the cipher can be broken the same way as Vigenere Cipher.

To convert this cipher to Vigenere Cipher, we can follow a simple algorithm. Suppose the length of ciphertext is `n`. `m` is length of the character set. `C(i)` is $\texttt{i}^{th}$ character in cipher text (index starts from 1). `I(c)` is the index of `c` in character set. `A(i)` is the character at index `i` in the character set. Following algorithm converts the cipher to Vigenere Cipher:

```
let i = n
while i > 1 do
  let ind = I(C(i))
  let ind_p = I(C(i-1))
  let ind_new = ind - ind_p
  if ind_new < 1 then
    ind_new += m
  end
  C(i) = A(ind_new)
end
```

After converting to Vigenere Cipher, we can adopt techniques for breaking Vigenere Cipher. When there are repetitions in the cleartext, e.g., common words like "the", and it happens that they are shifted using the same key characters, repetitions can occur in ciphertext. As a result, the number of characters between the repetition in ciphertext is multiple of the period of the key. In other words, the period of the key is likely to be a factor of the gap (the number of characters between repetitions) of the ciphertext.

Based on this fact, we can start by looking for repetitions in the ciphertext. The longer the repetition characters are, the better it is because it's less likely a coincidence. After finding the repetitions, we take their gaps and list all the factors. Then we start with the longest repetition characters, find common factors of their gaps, and assume that to be our key period.

To verify a key period guess, we re-arrange the ciphertext into N columns, where N is equal to the being guessed key period. Suppose our guess is correct, each column represents the result of a key character, i.e., all characters in a column are enciphered with the same key character. Then we calculate the ICs (indices of coincidence) of all columns, and compare with our statistical data to see if they are all likely encrypted by single alphabet. If so, we can continue with the guessed key length.

Then we count the frequencies of each character's appearing in each column. This gives us a list of high-frequency characters and low-frequency characters. By comparing this to a statistical frequency list for plain English, we can guess the substitutions for each column. Since we have multiple columns, they can serve as complementation of each other. If at some point we get a paragraph that reads as English and makes sense, that's the cleartext and we know the key.

This method attacks the cipher by looking at repetitions and by using statistical of alphabets used in the language. It needs some luck, and needs a lot of time of guessing. Also, in order to use it, we need to know the natual language being used in the cleartext. The language also need to have statistical characteristics (e.g. some characters are used more than others).