Network Security Assignment 1 - Monoalphabetic Substitution Cipher

Reshan Faraz: PhD19006

Koustuv Kanungo: PhD18007

Problem Statement

To implement a monoalphabetic substitution cipher with substitution of 2 symbols at a time, and then try to brute force the key from a given collection of ciphertexts by using some property of the plaintext ot verify the validity of obtained key.

Design

The code has been written in C++, and all of encryption, decryption, and brute force attacking are done through reading the plaintext and ciphertext 2 characters at a time.

The key is generated by giving the number of symbols to a key generator, which then outputs a sequence to use for substitution 2-symbol tuples.

The encryption works by using the key to scramble the tuples generated from the set of symbols, and then substitute 2-symbol pairs in the plaintext with their scrambled counterparts.

The decryption follows the opposite order by switching the key order with the indices of the tuple, and use this new order to substitute the tuples in the encrypted text to get back the original text.

The brute force attack is made by permuting the tuple order and testing the decryption algorithm in a loop, until the resulting text satisfies the property of being in non-decreasing order. This is checked by assigning a value to each tuple.

Code running example

```
./keygen 3
cat monokey
8 0 5 2 3 7 4 1 6 %

./encrypt symbols plaintext monokey
cat encoded_text
ca ca ca ac aa cc cc cc
ab bc cc cc cc cc cc cc
ca ca bb ac aa cc cc cc
```

```
cb cb ba cc cc cc cc cc
ca ba bc aa cc cc cc cc
ca ca ac aa aa aa cc cc
ca ca ac bc bc aa cc cc
ca ca ca cb cc cc cc cc
ca ac aa aa cc cc cc cc
cb aa cc cc cc cc cc cc

./decrypt symbols encoded_text monokey
cat decoded_text
cc cc cc ba ab aa aa aa
cb ac aa aa aa aa aa aa
cc cc ca ba ab aa aa aa
bc bc bb aa aa aa aa aa
cc bb ac ab aa aa aa aa
cc cc ba ab ab ab aa aa
cc cc ba ac ac ab aa aa
cc cc cc bc aa aa aa aa
cc ba ab ab aa aa aa aa
bc ab aa aa aa aa aa aa

./codebreaker symbols encoded_text
Key is:
1 3 4 5 6 2 8 7 0
```