Segurança Computacional - Trabalho 1

Aluno: João Pedro Assunção Coutinho
Matrícula: 18/0019813

1. Introdução

O trabalho é uma atividade em torno da cifra de Vigenere, e é composto por duas partes:

- Na primeira, devem ser implementadas cifração e decifração utilizando o algoritmo, com chave conhecida.

- Na segunda, deve ser implementado um algoritmo de ataque à cifra com acesso somente ao texto cifrado, utilizando análise de frequências.

Em ambas partes foi utilizada a linguagem Python 3.10, e o sistema operacional Ubuntu 22.04.

2. Desenvolvimento e abordagens utilizadas

a. Primeira parte

O procedimento para ambos cifração e decifração é muito semelhante, portanto aqui, foi utilizado somente uma função, que recebe o texto a ser cifrado/decifrado, a key que deve ser utilizada, e o modo escolhido (cifração ou decifração).

A função itera no texto de entrada, de linha a linha, e cada letra de cada linha é processada. Para cada letra:

- Verifica-se se a letra é válida e pode ser cifrada utilizando vigenere, ou seja, pertence ao alfabeto. Se não for, por exemplo, se for um número ou sinal, a letra é simplesmente adicionada à string resultante. Caso contrário, se for válida, a letra é passada para lowercase, é calculado seu valor ascii, e subtrai-se o valor da letra “a”, dessa forma teremos números entre 0 e 25 (0 sendo “a” e 25 sendo “z”).

- Se a letra for válida, também é necessário calcular em quanto ela será deslocada, e isso é feito utilizando a key. Aqui a key também é processada como lowercase, e seleciona-se o caractere que será utilizado, a partir de um key index, que é incrementado sempre que uma letra válida é encontrada, e é aplicada uma operação de módulo ao resultado desse incremento, pelo tamanho da key, dessa forma, sempre que key index chega ao fim da key, o próximo incremento o fará retornar ao início. Pelo mesmo motivo do passo anterior, aqui também se subtrai o valor da letra “a”.

- Tendo em mãos um valor para a letra do texto a ser cifrado/decifrado (entre 0 e 25) e o valor do deslocamento a ser realizado (entre 0 e 25), pode-se realizar a operação

principal, que é a soma de ambos aplicando-se um módulo 26, no caso da cifração, e uma subtração no caso da decifração.

- Ao final, temos um valor resultante, entre 0 e 25, que representa uma letra deslocada de acordo com a key. Agora, podemos somar o valor de “a” de volta, a fim de ser possível transformar esse valor em um caractere novamente. E finalmente, antes de adicionar o caractere resultante à string resultante, checa-se se a letra original era maiúscula, e em caso positivo, passa-se a letra resultante para maiúscula.

b. Segunda parte

Para a segunda parte, primeiramente o texto foi processado removendo-se caracteres não processáveis pela cifra de Vigenere. O algoritmo basicamente objetiva encontrar a chave de cifração para que seja possível decifrar utilizando a primeira parte:

- O texto foi separado em trigramas, e as distâncias entre as ocorrências dos trigramas foram computadas e fatoradas. Os fatores são adicionados a um dicionário que torna possível contar a ocorrência de cada fator.

- Ordenando-se reversamente (maiores primeiro) o dicionário de fatores de acordo com o número de ocorrência, obtém-se, com os maiores valores, os tamanhos mais prováveis de chave.

- Aqui, assume-se que o fator que mais ocorre é o tamanho da chave, o que faz com que o algoritmo nem sempre funcione. (Para o desafio 2, é necessário fazer uma mudança, escolhendo o segundo fator mais comum, e não o primeiro).

- Em posse do tamanho do chave, dividimos o texto cifrado em “t” partes (sendo “t” o tamanho da chave), de forma a sempre escolher, para uma parte “i”, uma letra a “t” de distância de si. Dessa forma, caso o tamanho escolhido de chave esteja correto, tem-se que cada parte “i” foi sempre cifrada por um mesmo caractere da chave (como em uma cifra de César) e a partir daqui, é possível conseguir a chave utilizando análise de frequência.