Trabalho 01 - Implementação da Cifra de Vegenère
Segurança Computacional - Turma 02
2º/2023

Mirella Gomes Silva Nascimento
Universidade de Brasília
202033525
mirella.nascimento@aluno.unb.br

October 2, 2023

1 Objetivos

O objetivo deste trabalho é compreender como funciona o algoritmo de criptografia Cifra de Vigenère. Isso é feito por meio de um cifrador e decifrador com uma palavra chave de um texto. Além disso, é solicitado também a realização de um ataque de recuperação de senha por análise de frequência.

2 Implementação

O código se ínicia com a leitura de um arquivo txt que contém uma palavra chave e uma **mensagem**, caso a opção desejada seja o cifrador, e um **criptograma** e uma palavra chave, caso a ação escolhida seja o decifrador. Em seguida, é escolhida a ação a ser realizada. Neste trabalho apenas as funções de cifrar e decifrar foram implementadas.

Inicialmente foi definido um dicionário contendo as letras do alfabeto como chave e a sua posição como valor. Com isso, a posição de cada letra da chave e da mensagem ou criptograma recebidos será encontrada por meio do dicionário.

Para definir as posições, é necessário iterar sobre o texto lido e sobre a chave recebida. Dessa forma, é possivel resgatar pelo dicionário a respectiva posição da letra indicada. Após a definição das posições, serão calculados os seguintes valores, dependendo da ação escolhida [KL21].

Para o processo de cifrar uma mensagem, temos que:

$$C_i = (P_i + K_i) mod 26 \tag{1}$$

Para decifrar é realizado a função inversa:

$$P_i = (C_i - K_i) mod 26 \tag{2}$$

Após o cálculo, deve-se retornar o letra correspondente ao número encontrado com o objetivo de formar a mensagem que desejada.

No código existe a presença de um contador i. Ele é responsável por repetir a palavra chave caso ela seja menor que o texto inserido. Isso é feito quando o valor do contador atinge o tamanho máximo da palavra chave e, ao zerá-lo, ele retorn ela para sua posição inicial.

2.1 Cifrador

```
def cipher(text, key):
    cipher_text = ""
    i = 0
    for value in text:
        value = value.lower()
        if value in alfabeto:
            if i == len(key):
                i = 0
            t = alfabeto[value]
            k = alfabeto[key[i]]

            c = (t + k) % 26
            ct = list(alfabeto)[list(alfabeto.values()).index(c)]

            cipher_text+= ct
            i+=1
        else:
            cipher_text+=value

    return cipher_text
```

2.2 Decifrador

```
def decipher(text, key):
    decipher_text = ""
    i = 0
    for value in text:
        value = value.lower()
        if value in alfabeto:
            if i == len(key):
                i = 0
            c = alfabeto[value]
```

```
            k = alfabeto[key[i]]

            d = (c - k) % 26
            dt = list(alfabeto)[list(alfabeto.values()).index(d)]

            decipher_text+= dt
            i+=1
        else:
            decipher_text+=value

    return decipher_text
```

References

[KL21] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2021.