

# Segurança Computacional

## Implementação da Cifra de *Vigenère*

Oseias Romeiro Magalhães / 211036123  
Prof. João Gondim

11 de dezembro de 2022

### Resumo

Neste trabalho de Segurança Computacional, foi explorado a cifra de *Vigenère* e o desenvolvimento de um programa composto por duas partes: o cifrado/decifrador e o ataque de recuperação de senha por análise de frequência.

## 1 Introdução

A cifra de *Vigenère* é um método de criptografia que usa uma série de diferentes cifras de César baseadas em letras de uma senha.

- Cifra de César: Consiste em incrementar ou decrementar cada letra por um número fixo, movimentando em N (chave) posições.[1]
- Cifra de *Vigenère*: Já na Cifra de *Vigenère*, a cada posição é atribuído um incremento diferente, dependendo da posição do *keystream*, sendo o *keystream* a repetição das letras de uma chave até alcançar o tamanho da mensagem a ser cifrada.[2]

## 2 Ambiente

Para o desenvolvimento desse projeto, foi utilizado a linguagem *Python* na versão 3.10.6, utilizando apenas os módulos padrão da linguagem.

## 3 Descrição

### 3.1 Cifrador/Decifrador

O cifrador recebe uma senha e uma mensagem que é cifrada segundo a cifra de *Vigenère*, gerando um criptograma, enquanto o decifrador recebe uma senha e um criptograma que é decifrado segundo a cifra de *Vigenère*, recuperando uma mensagem.[2]

```

class Cipher(Help):
    SYMBOLS: str
    SYMBOLS_SIZE: int

    def __init__(self, symbols) -> None:
        self.SYMBOLS = symbols
        self.SYMBOLS_SIZE = len(symbols)

    def encode(self, letter:str, key_index:int) -> str:
        return self.SYMBOLS[((self.SYMBOLS.find(letter) + key_index) % self.SYMBOLS_SIZE)]

    def decode(self, letter:str, key_index:int) -> str:
        return self.SYMBOLS[((self.SYMBOLS.find(letter) - key_index) % self.SYMBOLS_SIZE)]

```

```

class Viginere(Cipher):

    def __init__(self, symbols:str=string.ascii_lowercase) -> None:
        super().__init__(symbols)

    def gen(self, key:str, message:str) -> str:
        key_size = len(key)

        return "".join(
            key[i % key_size]
            for i in range(0, len(message))
        )

    def enc(self, keystream:str, message:str) -> str:
        self.vrfy_key(message, keystream)

        return "".join(
            self.encode(l, self.SYMBOLS.find(keystream[i]))
            if (l in self.SYMBOLS) else l
            for i,l in self.enumerate(message, self.SYMBOLS)
        )

    def dec(self, keystream:str, ciphertext:str) -> str:
        self.vrfy_key(ciphertext, keystream)

        return "".join(
            self.decode(l, self.SYMBOLS.find(keystream[i]))
            if (l in self.SYMBOLS) else l
            for i,l in self.enumerate(ciphertext, self.SYMBOLS)
        )

```


## 3.2 Recuperação de senha

O ataque de recuperação de senha por análise de frequência: recebe uma mensagens cifrada que é utilizada para recuperar a senha geradora do *keystream*, usado na cifração e então decifradas.[3]

```

unb@desktop:~/vigenere_cipher$ python3 main.py ideia -e
saved: data/encrpted.txt
unb@desktop:~/vigenere_cipher$ python3 main.py ideia -d
saved: data/decripted.txt
unb@desktop:~/vigenere_cipher$ python3 main.py -b pt
possible key sizes: [2, 3, 5, 6, 4]
choose: 5
key: ideia

```



```

data > encrypted.txt
1 i viouvge oumuvi mcqhqt isq uu fsvftlxw
mqombaz jpwbio uce lxvwu lh 1939 e 1945,
mndrpdevgs i milszii gea nifsms lr
qcnlr mvctxmvdw wslaa dw oriqhms
xrxmnklea ozjevihdhis mp hcaa dpqavfea
mqombazhw wpwvxis: wv etiigsa e w hmfo.
nrm i gchvza udma ajuevgmqxm di
kmatwumi, cwp qiia gi 100 uitksms lh
qqqlwezea psjitldidwv. iu eawelo lh
kcezue bobdp, ws xumvcseqs mqzwdlhws
lhhqciueu twge aui fexaklhdm hgwnwpmka,
qqhcsbumil m fmmnbljqci d wmrldgw dwv
iafwugws lh kcezue, leqaevdw gi talr e
liawmvcir ivtzh vmccuwws klzqs m
pmtibdvms. udvkalr twr cp rcmmus
aioqmnikdrbe lh ebayxia cwqzxa klzqs,
qqgtuqqhw o prpwcixwbo m d yvikd zmz mp
uce iuqis vxgteiua fwueu ublpqzigea eu
fsubiwi, noq r gwnnombo udma lmwet di
kmatwumi di kyuavlhdm, uiautwevdw
hrbrm d qiia gi uitksms lh qwrbbw.

data > decripted.txt
1 a segunda guerra mundial foi um conflito
militar global que durou de 1939 a 1945,
envolvendo a maioria das nacoes do
mundo incluindo todas as grandes
potencias organizadas em duas aliancas
militares opostas: os aliados e o eixo.
foi a guerra mais abrangente da
historia, com mais de 100 milhoes de
militares mobilizados. em estado de
guerra total, os principais envolvidos
dedicaram toda sua capacidade economica,
industrial e cientifica a servico dos
esforços de guerra, deixando de lado a
distincao entre recursos civis e
militares. marcado por um numero
significante de ataques contra civis,
incluindo o holocausto e a unica vez em
que armas nucleares foram utilizadas em
combate, foi o conflito mais letal da
historia da humanidade, resultando
entre a mais de milhoes de mortes.

```

## 4 Conclusão

Assim, foi implementado o cifrador e decifrador utilizando a cifra de *Vigenère* e foi mostrado uma de suas vulnerabilidades ao descobrir a chave utilizada a partir de uma mensagem cifrada.

## Referências

- [1] Daniel Adornes. *Quebrando a Cifra de Vigenère*. <https://informatabrasileiro.blogspot.com/2013/04/quebrando-cifra-de-vigenere.html>.
- [2] Udacity. *Vigenere Cipher*. <https://youtu.be/SkJcmCaHqS0>.
- [3] Wikipédia. *Frequência de letras*. [https://pt.wikipedia.org/wiki/Frequência\\_de\\_letras](https://pt.wikipedia.org/wiki/Frequência_de_letras).