Cifra de Vigenère

Edgar Sampaio de Barros[1]

[1]Departamento de Ciência da Computação 160005213@aluno.unb.br

ABSTRACT

O trabalho apresenta um programa em Python capaz de realizar as operações de cifragem, decifragem e ataque para descobrir a chave da cifra de Vigenère. A cifra de Vigenère é uma técnica de criptografia polialfabética que utiliza uma chave de tamanho variável para cifrar uma mensagem. O programa utiliza o conhecimento da frequência de letras em diferentes idiomas para quebrar a criptografia e descobrir a chave utilizada na cifragem.

Keywords: Cifra de Vigenère, Criptografia, Decifrador, Ataque criptográfico, Análise de frequência

1 INTRODUÇÃO

Criptografia é uma área da criptologia que se dedica ao estudo de técnicas e princípios para comunicação segura em situações onde existem terceiros, chamados de ”adversários”. O objetivo da criptografia é construir protocolos que impeçam estes adversários de lerem mensagens privadas. A criptografia moderna é essencial para a segurança da informação e engloba aspectos como confidencialidade, integridade de dados, autenticação e não-repúdio. Ela é resultado da interseção entre diversas disciplinas, tais como matemática, ciência da computação, engenharia elétrica, ciência da comunicação e física. As aplicações da criptografia são amplas e vão desde o comércio eletrônico até comunicações militares [2].

A cifra de Vigenère é um método de criptografia que utiliza diferentes cifras de César baseadas em letras de uma senha. É uma versão simplificada de uma cifra de substituição polialfabética inventada por Leon Battista Alberti. A cifra é fácil de ser implementada e gera um resultado bastante satisfatório a primeira vista, caso o leitor não esteja habituado com criptologia pode achar que são letras aleatórias [1].

O objetivo deste trabalho é relatar minha experiência na criação de um programa em python3 que implmenta a cifra de Vigenère. A motivação surgiu como resultado do trabalho da disciplina de Segurança Computacional (CIC0201) ministrada pelo Professor João Gondim na Universidade de Brasília (UnB) durante o primeiro semestre de 2023.

2 METODOLOGIA

2.1 A cifra de Vigenère

A cifra de César é um método simples de criptografia que consiste em substituir cada letra do alfabeto por outra letra, que se encontra um número fixo de posições à frente no alfabeto. Esse número fixo é conhecido como ”chave” da cifra. Por exemplo, com uma chave de 3, a letra ”A” seria substituída pela letra ”D”, a letra ”B” pela letra ”E” e assim por diante. A cifra de Vigenère consiste no uso de várias cifras de César em sequência, com diferentes valores de deslocamento ditados por uma ”palavra-chave” [1].

A cifra de Vigenère pode ser representada por meio de operações algébricas. Através do mapeamento das letras A-Z para os números inteiros 0-25 e da aplicação da operação de adição módulo 26, é possível escrever a operação de criptografia da seguinte forma:

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

e decriptação assim:

$$P_i \equiv P_i - K_i + 26 (mod 26) \tag{2}$$

Onde C_i é a i-ésima letra da mensagem criptografada, P_i é a i-ésima letra da mensagem e K_i é a i-ésima letra da chave. Caso a chave tenha o comprimento menor que a mensagem, é feito a concatenação da mesma palavra chave ate que seu tamanho seja maior ou igual ao da mensagem.

Para quebrar a cifra de Vigenère, utilizou-se um método baseado na análise de frequência de letras. Esse método compara a frequência de cada letra no texto cifrado com a probabilidade de ocorrência de cada letra em um determinado idioma, permitindo a identificação da chave de criptografia utilizada.

2.2 Implementação

O processo de leitura e escrita de dados no código foi realizado por meio de arquivos, onde a entrada de dados foi feita por meio do arquivo *”input.txt”* e a saída dos resultados foi escrita no arquivo *”output.txt”*. Isso permite que o usuário possa inserir uma mensagem longa para sere cifradas ou decifradas de uma só vez, e posteriormente obter os resultados no arquivo de saída. Além disso, o uso de arquivos facilita o processo de testes e de comparação entre diferentes resultados obtidos a partir de diferentes entradas de dados. Dessa forma, a implementação do código permite que o processo de criptografia e descriptografia seja automatizado e facilite a realização de operações em grande escala.

Para a implementação do código foi criado uma classe *Vigenere* que é encarregada de realizar todas as três operações (cifrar, decifrar e ataque).

O fragmento de código a seguir mostra como ficou a implementação da funcionalidade cifrar (listing 1) e decifrar(listing 2), respectivamente. Onde o atributo `self.alfabeto` representa as letras do alfabeto em ordem. O método `self.tratar_texto` recebe como argumento a mensagem em que será realizado a operação e retorna ela sem acentos, caracteres especiais, espaços e quebra de linha.

```
def cifrar(self, mensagem, key):
    key = self.tratar_texto(key)
    mensagem = self.tratar_texto(mensagem)
    mensagem_cifrada = ""
    for i, letra in enumerate(mensagem):
        k = key[(i)%len(key)]
        ki = self.alfabeto.index(k)

        li = self.alfabeto.index(letra)
        mensagem_cifrada += self.alfabeto[(li+ki)
    % 26]

    return mensagem_cifrada
```

Listing 1: Método para cifrar

```python
def decifrar(self, mensagem, key):
    key = self._tratar_texto(key)
    mensagem = self._tratar_texto(mensagem)

    mensagem_decifrada = ""

    for i, letra in enumerate(mensagem):
        k = key[(i)%len(key)]
        ki = self.alfabeto.index(k)

        li = self.alfabeto.index(letra)
        mensagem_decifrada += self.alfabeto[(li-ki) % 26]

    return mensagem_decifrada
```

Listing 2: Método para decifrar

A implementação da cifra de Vigenère para cifrar e decifrar é bastante semelhante, e é derivada diretamente da interpretação algébrica da cifra (1 e 2).

Para a implementação do método que tenta descobrir a chave utilizada na cifração foi necessário definir o atributo `self.ling_props` que é um dicionario que contem dois vetores com a probabilidade de ocorrência de cada letra do alfabeto em um determinado idioma (português ou inglês), os valores foram retirados dos links (português e inglês), também foi definido um atributo que limita o tamanho máximo possível para a chave, tal valor foi fixado em 10.

A funcionalidade de ataque é a parte mais complexa do código. O fragmento de código a seguir apresenta sua implementação (listing 3), que recebe como entrada a mensagem cifrada e realiza o processo de encontrar a chave para os idiomas português e inglês, mas não antes de tratar o texto para o deixar em uma forma que é possível realizar as operações.

```python
def ataque(self, mensagem):
    mensagem = self._tratar_texto(mensagem)
    chaves = []

    chaves.append(self._encontrar_chave(mensagem, "PT"))
    chaves.append(self._encontrar_chave(mensagem, "EN"))

    return chaves
```

Listing 3: Método ataque

```python
def _encontrar_chave(self, text, language):
    tamanho = self._tamanho_chave(text)
    chave = ""

    for i in range(tamanho):
        freq_letra_bloco = {}

        # Divide o texto em blocos do tamanho da chave
        # e calcula a quantidade de vezes em que cada letra aparece na posicao i do bloco
        for j in range(i, len(text), tamanho):
            freq_letra_bloco[text[j]] = freq_letra_bloco.get(text[j], 0) + 1

        prop_alfabeto = []

        # Calcula a probabilidade de cada letra do alfabeto nos blocos
        for letra in self.alfabeto:
            prop_alfabeto.append(100*freq_letra_bloco.get(letra, 0)/sum(freq_letra_bloco.values()))

        chave += self._encontra_letra(prop_alfabeto, language)
    return chave
```

Listing 4: Método encontrar_chave

O método `self._encontrar_chave` (listing 4) é responsável por encontrar a chave utilizada para cifrar a mensagem, recebendo como argumento a mensagem tratada e o idioma para o qual se deseja realizar a quebra. Para realizar essa tarefa, o método precisa primeiramente encontrar o tamanho provável da chave. Tal valor é determinado utilizando o algoritmo do índice de coincidência, que identifica o tamanho mais provável da chave com base nas similaridades entre blocos da mensagem (listing 5).

```python
def _tamanho_chave(self, mensagem):
    tamanhos_possiveis = []

    for tamanho in range(2, len(mensagem)-2):
        dist = []
        for i in range(len(mensagem)-2):
            for j in range(i+tamanho, len(mensagem)-tamanho, tamanho):
                if mensagem[i:i+3] == mensagem[j:j+3]:
                    dist.append(j-i)
        if dist:
            gcd = self._calcula_mdc(dist)
            if gcd > 1 and tamanho < self.tamanho_max_chave:
                tamanhos_possiveis.append((tamanho, len(dist)/(len(mensagem)/gcd)))
    return [x[0] for x in sorted(tamanhos_possiveis, key=lambda x: x[1], reverse=True)][0]

def _calcula_mdc(self, dist):
    gcd = dist[0]
    for i in dist[1:]:
        gcd = math.gcd(gcd, i)
    return gcd
```

Listing 5: Método tamanho_chave

Depois de encontrar o possível tamanho para a chave, o método realiza a divisão da mensagem em blocos do tamanho da chave e o cálculo da frequência de ocorrência de cada letra em cada bloco. Em seguida, é calculada a probabilidade de cada letra do alfabeto nos blocos, e a diferença entre a probabilidade encontrada e a probabilidade esperada em cada idioma é calculada (listing 6). A letra que apresenta a menor diferença é adicionada à chave. A função realiza está operação para os idiomas português e inglês e retorna um vetor com a chave encontrada para cada idioma.

```python

def _encontra_letra(self, probability, language):
    diferencas =[]

    #calcula a diferenca entre a probabilidade da letra atual na lingua nativa do texto
    for i in range(26):
        soma = 0
        for j in range(26):
            soma += abs(probability[(i+j) % 26] - self.ling_props[language][j])

        diferencas.append(soma)

```

```
13      #retorna a que tem a menor diferenca
14      return self.alfabeto[diferencas.index(min(
        diferencas))]
```

Listing 6: Método encontra_letra

3 Conclusão

Este relatório apresentou a implementação da cifra de Vigenère em Python e sua quebra utilizando métodos de análise de frequência de letras. A cifra de Vigenère é um método de criptografia clássico que usa uma chave para cifrar e decifrar uma mensagem. A quebra da cifra de Vigenère pode ser realizada facilmente graças ao atual poder computacional por meio da análise de frequência de letras da mensagem cifrada, o que pode ser facilitado pela utilização de técnicas como a análise de índice de coincidência e a comparação com a frequência de letras em um idioma específico.

Em geral, a implementação do código realiza todas as operações propostas com sucesso. A operação de ataque retorna duas possíveis chaves para os idiomas português e inglês, porém é importante ressaltar que a quebra da cifra de Vigenère de forma totalmente automática é uma tarefa bastante complexa e pode apresentar mais de uma possível chave correta. Além disso, é importante considerar que a eficiência da operação de ataque depende da qualidade da mensagem cifrada e da quantidade de texto disponível para análise. No entanto, o código implementado apresenta uma solução viável e eficaz para a cifra proposta.

References

[1] Wikipedia. Cifra de vigenère.
[2] Wikipedia. Criptografia.