Cifra de Vigenère

Rodrigo Mello da Rosa

[1]Escola Politécnica – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) - Porto Alegre - RS - Brasil

Resumo. *O objetivo deste trabalho é descrever, através de uma abordagem simplificada, o funcionamento da implementação do algoritmo apresentado para o primeiro trabalho da disciplina de segurança de sistemas.*

1. Introdução

A cifra de Vigenère é uma cifra clássica e razoavelmente simples de entender, implementar e até mesmo quebrar, mas mesmo assim três séculos se passaram até que alguém conseguisse quebrá-la. Chegou a ser chamada de ”A cifra indecifrável”. Um método geral de decifrá-la foi publicado pelo criptógrafo e arqueólogo alemão Friedrich Kasiski em 1863. Essa cifra consiste em várias cifras de César utilizadas em sequência, com valores de deslocamento diferentes e obtidos a partir de uma palavra-chave previamente estabelecida.

2. Funcionamento

Então tudo parte de uma chave *K* previamente definida. Esta chave é sobreposta ao texto a ser cifrado *M* e repetida o numero de vezes necessário, quanto maior a chave mais difícil de termos repetições como é mostrado na figura 1.

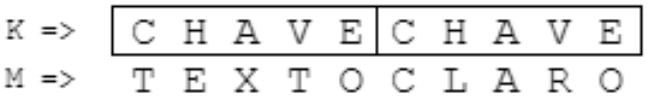

Figura 1. Chave sobreposta

Agora, usando aritmética modular é feito a soma dos caracteres da chave com os do texto base. Com isso observamos que uma letra nem sempre vai ser cifrada da mesma forma.(Figura 2)

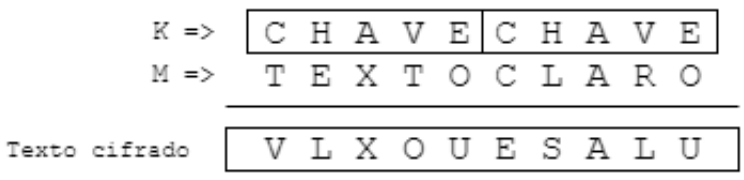

Figura 2. Soma da chave e texto

3. Solução para o trabalho

A ideia aqui é desenvolver uma forma mais automatizada possível para tentar quebrar um texto cifrado com a cifra de Vigenère. O método escolhido foi o calculo do índice de coincidência, baseado na ideia proposta por William Friedman. O Índice de coincidência de um texto qualquer é definido como a probabilidade de dois elementos aleatórios serem idênticos. A seguir teremos os passos que foram seguidos para decifrar o texto proposto.

3.1. Tamanho da chave

O primeiro passo é descobrir o tamanho da chave utilizada para cifrar o texto que queremos decifrar. Para isso o texto é subdividido em *C* partes de forma a testar vários tamanhos de chave *m*. Por exemplo para *m = 2* teremos o texto dividido em duas partes *C*, das quais *C1* com todos caracteres nas posições *0, 2, 4, 6 ...* do texto cifrado e *C2* tera as posições *1, 3, 5, 7 ...* . Então para cada tentativa de valor de *m*, é calculado o índice de coincidência para cada parte *C* e feito uma média destes valores. O Tamanho testado que mais se aproximou do índice de coincidência da língua corrente do texto é eleito como tamanho da chave *m*.(Figura 3)

$$C_1 = C_1 C_{m+1} C_{2m+1}$$

$$C_2 = C_2 C_{m+2} C_{m+2}$$

$$C_m = C_m C_{2m} C_{3m}$$

Figura 3. Soma da chave e texto

4. Encontrar letras da chave

Já temos o tamanho da chave que foi utilizado para cifrar o texto. Agora precisamos desvendar o conteúdo da chave. Para isso novamente o texto é cifrado e divido, mas agora com o tamanho da chave certo. Com isso presumimos que todo o texto em cada uma destas partes foi cifrado com o mesmo caractere. Então para cada uma destas partes é levantado os três caracteres mais incidentes e um deles é eleito como o caractere utilizado na encriptação. De forma a dar uma certa semântica em como é feita a escolha dentre estes três caracteres mais frequentes, isto é feito pelo usuário e é repetido para todas as partes do texto. Ao fim teremos todas letras que foram utilizadas na chave.

Cada caractere escolhido para a chave é assumido como representante do caractere mais frequente na linguagem. Tendo todas estas informações podemos decifrar o texto utilizando a possível chave encontrada anteriormente.

5. Executando o script e resultados

Para executar o script, basta usar o seguinte comando;

```
node main.js
```

Para cada caractere da chave sera solicitado a escolha de uma das três mais prováveis letras para aquela posição da chave. Neste momento pode ser escolhido tanto o numero da opção *1, 2, 3* quanto a letra proposta *m, q, a*, segue um exemplo;

```
Select one of this possible most incident character for this set:

    1 - [letter: "m" incidence: 267]
    2 - [letter: "q" incidence: 262]
    3 - [letter: "a" incidence: 224]
```

Por final é exibido o resumo da execução, temos o tamanho da chave, o índice de coincidencia obtido para aquele tamanho de chave, e também a chave que foi utilizada. O texto claro é salvo em um arquivo chamado *out.txt*:

```
    Key Length: 7   ||    IC: 0.07686708922340826

    Possible key: meunome

    Result is saved on out.txt file.
```