Relatório Segurança Computacional - Cifra de Vigenère

João Victor de Souza Calassio - 180033808 William Coelho da Silva - 180029274

March 14, 2022

Universidade de Brasília - Instituto de Ciências Exatas
Departamento de Ciência da Computação - CIC
CIC0201 - Segurança Computacional - Turma B
Professor: João José Costa Gondim
Prédio CIC/EST - Campus Universitário Darcy Ribeiro
Asa Norte 70919-970 Brasília, DF

1 Introdução

Este projeto tem como objetivo explorar a Cifra de Vigenère, e suas vulnerabilidades.

A Cifra de Vigenère é um método para criptografar textos alfabéticos, realizando uma série de substituições letra-a-letra, baseado em uma chave escolhida.

Essa cifra foi inicialmente descrita em 1553, e era chamada de "a cifra indecifrável" até 1863, quando Friedrich Kasiski publicou um método geral para decifrar textos criptografados utilizando a Cifra de Vigènere, que foi posteriormente aprimorado através do método de Kerckhoff.

Neste projeto, implementaremos um cifrador e decifrador para a Cifra de Vigènere, que consiga criptografar/descriptografar qualquer texto dado a chave correta, e exploraremos as vulnerabilidades dessa cifra utilizando o método de Kerckhoff para encontrar qual a possível chave utilizada para criptografar dois textos pré estabelecidos, um em português e outro em inglês.

Foi escolhido Python para implementação deste projeto por ser uma linguagem simples, tornando fácil a manipulação de texots (*strings*), que é nosso principal objeto de estudo neste trabalho.

2 Desenvolvimento

2.1 Cifrador e decifrador

Começamos o desenvolvimento fazendo o cifrador, é uma função que recebe o texto a ser cifrado e uma chave, o primeiro passo é o de "consertar" a chave, uma vez que ela precisa ter o tamanho do texto, por exemplo, se a chave "TI" for usada para cifrar o texto "ATACARBASENORTE", a chave precisa ficar da seguinte maneira: "TITITITITITITIT".

Depois percorremos o texto a ser cifrado e verificamos se o char de cada posição é válido, isto é, se ele é uma letra maiúscula de A a Z, neste passo ignoramos espaços, números e caracteres especiais, e simplesmente os colocamos no resultado cifrado, as condições sendo satisfeitas, somamos o valor da tabela ASCII do char[i] do texto e char[i] da chave e depois o resto da divisão por 26, dessa forma garantimos que o resultado sempre será maior ou igual a A e menor ou igual a Z e o colocamos na string resultante.

Já a implementação do decifrador é praticamente idêntica a do cifrador, a única diferença é que em vez de somarmos o valor ASCII do char[i] do texto cifrado e da chave, os subtraímos.

2.2 Ataque de recuperação de chave

A Cifra de Vigenère, assim como outras cifras poli-alfabéticas, está suscetível à análise de frequências. Isto é, se soubermos o idioma em que o texto original foi escrito, podemos verificar a correlação entre a frequência de cada uma das letras no texto cifrado e no alfabeto desse idioma. Por exemplo, se em um texto em inglês for cifrado, e a letra "V" for a mais frequente, há uma boa chance da letra "V" corresponder à letra "E" no texto original.

No entanto, na Cifra de Vigenère não podemos fazer uma simples análise em todo o texto cifrado, pois uma mesma letra do texto original pode resultar em diferentes letras em cada ponto da mensagem cifrada.

Podemos resolver esse problema se soubermos o tamanho da chave utilizada, já que na cifração a chave é repetida várias vezes até ter o tamanho do texto original. Sendo assim, se soubermos que a chave tem tamanho n, e dividirmos o texto cifrado em n trechos (colunas), temos certeza que em uma mesma coluna foi utilizada a mesma letra da chave para cifrar. Por exemplo, para o texto original "ATACARBASENORTE" e a chave "TI": como o tamanho da chave é 2, sabemos que se dividirmos o texto em trechos de 2 letras, a primeira letra sempre vai ser cifrada com "T" e a segunda letra sempre vai ser cifrada com "I":

```
AT AC AR BA SE NO RT E
TI TI TI TI TI TI TI T
```

Podemos então realizar a análise de frequência em cada uma das n colunas do texto cifrado, e estimar qual a possível letra utilizada para cifrar cada uma das colunas. Temos então a chave.

Mas como podemos encontrar o tamanho da chave?

2.2.1 Encontrando o tamanho da chave

O método de Kasiski para encontrar o tamanho da chave consiste em medir as distâncias entre bigramas/trigramas repetidos no texto cifrado. Se uma mensagem for longa o suficiente, haverão vários bigramas/trigramas repetidos no texto cifrado que correspondam a um mesmo bigrama/trigrama no texto original. Se medirmos e fatorarmos as distâncias entre esses bigramas/trigramas, podemos estimar o tamanho da chave utilizada para cifrar esse texto.

Para o projeto, escolhemos verificar por trigramas, por dar resultados aparentemente melhores nos testes realizados.

Na implementação, esse processo é feito em três etapas:

1. Verificação de trigramas repetidos
    - Nesta etapa, varremos o texto cifrado e guardamos todas as sequências de três letras que aparecem mais de uma vez.
2. Cálculos das distâncias entre os trigramas repetidos
    - Agora que temos os trigramas que se repetem, calculamos as distâncias entre todas as ocorrências de um mesmo trigrama.
3. Fatoração das distâncias e estimativa do tamanho da chave
    - Agora, fatoramos todas as distâncias encontradas na etapa anterior e verificamos quais fatores mais aparecem. O fator mais frequente provavelmente é o tamanho da chave (mas nem sempre é o caso). Reduzimos muito nossas possibilidades, e podemos então fazer a análise de frequência para encontrar a chave.

2.2.2 Encontrando o valor da chave

Agora que sabemos o tamanho da chave, podemos dividir o texto cifrado em trechos de n letras (colunas), e realizar a análise de frequência em cada uma dessas colunas.

Nesta implementação, apenas relacionamos as três letras mais frequentes na coluna com a letra mais frequente no alfabeto escolhido, e contamos quantos deslocamentos à direita foram feitos para saber qual a letra correspondente na chave (por exemplo, se "E" for a letra mais frequente no alfabeto, e "G" a mais frequente na coluna, foram feitos três deslocamentos então a letra correspondente na chave deve ser "C").

2.3 Decifrando o desafio1.txt e desafio2.txt

Para decifrar os desafios, executamos o algoritmo de ataque duas vezes para cada texto cifrado, uma vez tentando o ataque em português, e outra tentando o ataque em inglês. Isso foi necessário pois não sabemos qual texto estaria em qual idioma.

Por tentativa e erro, descobrimos que o desafio1 está em português, e a chave é **ARARA**, e o desafio2 está em inglês, e a chave é **TEMPORAL**.

O ataque realizado no desafio1, assumindo que o texto estava em inglês estimou que o tamanho da chave era 5 letras, com uma boa margem (1025 ocorrências para o fator 5, contra 831 ocorrências do fator 2 que seria o próximo candidato).

Realizando a análise de frequência, obtivemos as seguintes possibilidades (cada coluna corresponde a um caractere da chave, e cada linha representa uma letra candidata para aquela posição, sendo primeira mais provável e última menos provável):

```
P R A R A
A U E A E
N V P B W
E N J G N
```

O ataque realizado no desafio2, assumindo que o texto estava em português, estimou que o tamanho da chave seria 2, 4 ou 8. A ocorrência de todos os três fatores estava bem próxima (388 vs 365 vs 339), então começamos os testes pelo 8, por ser maior.

Realizando a análise de frequência, obtivemos

```
T E M T O V E P
X I A P S F A Z
H W Q D C R O C
K S E G G J I L
```

3 Conclusão

Neste projeto fomos capazes de aprender o funcionamento de um cifra polialfabética, por um lado ela pode ser boa e simples para quem faz o uso dela e complexa para se decifrar a depender da chave para o *hacker/cracker*, mas apenas no caso de textos pequenos como senhas de contas digitais alfabéticas porque não é viável fazer análise de frequência com textos pequenos.

Porém, ao cifrar um parágrafo, por exemplo, a análise de frequência de caracteres se torna bem mais fácil, e qualquer criptoanalista seria capaz de decifrar, por exemplo, em uma guerra, o QG não poderia essa cifra para dar ordens grandes (com muitas letras) aos seus comandantes porque o inimigo poderia interceptar e facilmente quebrar a cifra.

E ainda levando em conta o poder computacional atual, podendo fazer bilhões de cálculo por segundo, ainda mais com inteligência artificial, que poderia aprender a dizer aos humanos quando a cifra está correta e tirar

mais uma função de análise das pessoas tornou a cifra de Vigenère usável apenas no contexto pessoal para senhas e afins.

4 Referenciais bibliográficas

Página da cifra na Wikipedia

Cifrador/Decifrador online

Métodos para quebrar a cifra

Vigenere Cipher - Department of Information Technology Uppsala Universitet

Vigenere Frequency Analysis Tool