Cifra de Vigenère e Ataque por Análise de Frequência

Autores:
Pedro Brum Tristão de Castro – 202067470
Mateus Oliveira Santos – 221029150

Link para o repositório:

https://github.com/PedroBrumTC/Cifra_Vigenere.git

1. Introdução

A cifra de Vigenère é uma cifra polialfabética clássica, uma variante da cifra de César. A cifra de César consiste em substituir cada letra do texto pela letra que está a **k** letras após ela no alfabeto. Já a cifra de Vigenère usa utiliza uma palavra-chave para definir esse valor k para cada posição no texto. Por exemplo, usando a chave "**dia**" teremos **k = 3** para a primeira letra do alfabeto, **k = 8** para a segunda, **k = 0** para a terceira, **k = 3** para a quarta e assim segue.

Essa cifra altera dinamicamente o alfabeto de cifragem, dificultando ataques por análise de frequência simples. Porém não é impossível decifrar esse algoritmo. Esse algoritmo utiliza uma chave cíclica, ou seja, dado uma chave de tamanho **n**, toda letra com posição múltipla de **n** no texto cifrado possuem o mesmo deslocamento. Com isso, um ataque pode verificar o tamanho provável da chave testando vários valores possíveis e escolhendo aqueles com maior número de coincidências.

Agora para descobrir cada caractere da chave, é preciso escolher um possível caractere da chave, checar a frequência de cada letra e comparar as frequências no texto com as frequências na própria língua. Possibilitando descobrir o caractere mais provável para cada posição na chave.

Com isso em mente, esse trabalho visa implementar dois componentes: (i) o cifrador/decifrador da cifra de Vigenère, e (ii) um ataque por análise de frequência, visando recuperar a senha a partir de mensagens cifradas, utilizando conceitos como Índice de Coincidência e análise estatística de frequências de letras.

2. Implementação do Cifrado/Decifrador

A cifra de Vigenère foi implementada em python. Ela considera cada símbolo como um byte, e não como uma letra no alfabeto. Então o programa lê um arquivo como um conjunto bytes (e não string) e aplica o codificador/decodificador. O algoritmo permite escolher se irá

implementar a cifra como uma soma, o padrao dessa cifra, ou se vai utilizar a operação XOR, que tende a ser mais rápida.

Cifrar: SOMA: C = (ord(M) + ord(K)) mod 256 **XOR: C = (ord(M) ^ ord(K))**

Decifrar: SOMA M = (ord(C) - ord(K)) mod 256 **XOR: C = (ord(M) ^ ord(K))**

3. Ataque por Análise de Frequência

3.1 Determinação do Tamanho da Senha

Para a determinação do tamanho da chave foi utilizado o método de Kasiski, que conciste em analisar sequências repetidas no texto. O procedimento é o seguinte:

1. Procuramos por sequências repetidas de 3 ou mais caracteres no texto cifrado.

2. Calculamos as distâncias entre as posições dessas sequências repetidas.

3. Determinamos os divisores comuns dessas distâncias (limitado a um t_max) e calculamos a frequência desses divisores.

4. O tamanho de senha mais provável é o divisor que ocorre com maior frequência entre as distâncias analisadas.

Este método é eficaz especialmente em textos longos, onde há maior probabilidade de ocorrência de sequências repetidas significativas.

3.3 Recuperação das Letras da Senha

O processo de recuperação da senha é o seguinte:

1. Cada letra possível, em cada posição da chave, é usada para cifrar seu respectivo subgrupo.
2. Em cada subgrupo cifrado, são calculadas as frequências de cada letra.
3. A lista de frequências é comparada com as frequências na língua utilizada, a diferença entre elas é calculada usando “chi square”.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

O -> Observado; E -> Esperado

4. A letra com menor χ^2 é a mais provável de ser a correta.
5. As letras mais prováveis em cada posição são encadeadas para descobrir a chave.

4. Resultados Obtidos

Para analisar os resultados obtidos foi utilizado o livro Dom Casmurro. A partir do qual forma pegos trechos de tamanhos 10, 25, 50, 100, 200, 500 e 1000, além do livro completo, utilizando a operação “add”.

Chave utilizada: qualquercoisavale

4.1 Cifra

Para o processo de criptografar e descriptografar foram gerados os seguintes valores:

Tamanho do texto	Tempo de cifra (ms)	Tempo de decifra (ms)	Precisão
10	1.88	1.07	100%
25	1.51	1.70	100%
50	0.88	1.24	100%
100	0.89	0.98	100%
200	1.72	1.46	100%
500	1.07	1.01	100%
1000	1.24	1.21	100%
389670	27.56	40.31	100%

Nessa tabela, percebe-se que esse algoritmo é muito rápido. Para todos os valores até 1000 caracteres foram realizados em questão de 1 ou 2 ms. Mesmo para o livro completo, com quase 390 000 caracteres ele levou apenas 40,31 ms.

4.2 Ataque

Para o ataque foram utilizados os mesmos arquivos e chave. Nele foram observados os seguintes valores:

Tamanho do texto	Tempo de ataque (ms)	Precisão da chave	Precisão do texto
10	0.03	0%	—
25	0.02	0%	—
50	0.03	0%	—
100	2.87	0%	2.94%
200	12.77	17.65%	17.33%
500	21.28	88.24%	88.19%
1000	25.40	100%	100%
389670	6093.95	100%	100%

Na tabela, percebe-se que abaixo de 100 caracteres o algoritmo não consegue nem gerar uma chave. A partir de 100 ele gera uma chave, porém ela não é confiável, já que não teve nenhuma coincidência com a chave original. Com 200, percebe-se que as chaves começaram

a coincidir. Porém só a partir de 500 a chave gerada é mais confiável e em 1000 o ataque já descobre a chave.

Quanto a questão do tempo ele é relativamente rápido, já que mesmo para o livro inteiro o código só precisou de 6 segundos.

5. Considerações Finais

Nesse trabalho foi demonstrado a implementação da cifra de Vigenère. Alem disso foi apresentado um dos principais ataques que podem ser implementados contra essa criptografia.

Esse ataque mostrou que, apesar de sua maior complexidade em relação a cifras monoalfabeticas, ainda há grande vulnerabilidade nesse algoritmo. Foi mostrada sua vulnerabilidade à ataques estatísticos, já que com um trecho relativamente pequeno (1000 caracteres ~ 1 a 3 paragrafos) é possivel descobrir a chave utilizada rapidamente.

6. Referências

- Wikipedia. 'Frequência de letras' — https://pt.wikipedia.org/wiki/Frequ%C3%AAncia_de_letras
- Wikipedia. 'Cifra de Vigenère' — https://pt.wikipedia.org/wiki/Cifra_de_Vigen%C3%A8re