Relatório - Cifra de Vigenère

INGRID LORRAINE RODRIGUES DA SILVA - 160125260
JOÃO VICTOR R. DA SILVA - 160127815

Dep. Ciência da Computação - Universidade de Brasília (UnB)

1. Introdução

O presente documento visa descrever a implementação dos processos de codificação, decodificação e um dos métodos de ataque para a Cifra de Vigenère. Abaixo, iremos descrever brevemente a Cifra de Vigenère, as decisões tomadas para implementação de cada algoritmo (codificação, decodificação e ataque) e suas particularidades.

2. Cifra de Vigenère

Criada por volta de 1465 por Leon Battista Alberti, a cifra de Vigenère se trata de uma evolução da cifra de César, pois usava uma palavra chave para cifrar a mensagem, o que dificultava o ataque com base na frequência das letras que era uma vulnerabilidade conhecida da cifra de César.

Para criptografar uma mensagem, cada letra do alfabeto é mapeada para um número de 0 a 25 de acordo com a ordem alfabética, ou seja, o “a” é mapeado para o 0, o “b” é mapeado para o 1, e assim sucessivamente. Em seguida é escolhida uma palavra chave, e a mesma tem suas letras repetidas até se igualar ao tamanho da mensagem. O valor de cada letra da mensagem é somado ao valor da letra da palavra chave, após isso é calculado o resto da divisão pelo tamanho do alfabeto.

Usando como exemplo a mensagem “frequencia” e a palavra chave “ola”, aplicando os passos descritos acima, iremos obter “tceefebnio” como resultado.

Passo 1:

- frequencia, (e.g. f -> 5, r -> 17, e -> 4, q -> 16, u -> 20, e -> 4, n -> 13, c -> 2, i -> 8, a -> 0);

Passo 2:

- olaolaolao, (e.g. o -> 14, l -> 11, a -> 0, o -> 14, …, o -> 14);

Passo 3:

- f -> 5 + 14 = 19, resto da divisão de 19 por 26 = 19, que corresponde à letra “t”.
- r -> 17 + 11 = 28, resto da divisão de 28 por 26 = 2, que corresponde à letra “c”.
- e -> 4 + 0 = 4, resto da divisão de 4 por 26 = 4, que corresponde à letra “e”.

Para decodificar uma mensagem cifrada utilizando a cifra de Vigenère, devemos repetir os passos 1 e 2 da codificação, porém utilizando a mensagem cifrada e palavra chave. Em seguida, devemos subtrair o valor da letra da palavra chave da mensagem cifrada, caso o valor seja maior ou igual a zero, o mesmo corresponde a letra da mensagem em texto claro. Caso o valor da subtração seja menor que zero, isso implica que durante a codificação, a soma foi superior a 25, sendo assim, foi pego o resto da divisão por 26, para

obter o valor da letra em texto claro, precisamos somar 26 ao resultado da subtração anterior.

Utilizando mensagem codificada “tceefebnio” do exemplo acima e aplicando os passos da decodificação descritos, temos “frequencia” como resultado, estando dentro do esperado.

Passo 1:

- tceefebnio, (e.g. t -> 19, c -> 2, e -> 4, e -> 4, f -> 5, e -> 4, b -> 1, n -> 13, i -> 8, o -> 14);

Passo 2:

- olaolaolao, (e.g. o -> 14, l -> 11, a -> 0, o -> 14, …, o -> 14);

Passo 3:

- t -> 19 - 14 = 5, 5 é maior que ou igual a zero, correspondendo à letra “f”.
- c -> 2 - 11 = -9, -9 é menor que zero, portanto temos de somar 26, -9 + 26 = 17, que corresponde à letra “r”.
- e -> 4 - 0 = 4, 4 é maior que ou igual a zero, correspondendo à letra “e”.

3. Codificação

Para a codificar a mensagem foi escrito um algoritmo em Python seguindo os passos descritos acima na Cifra de Vigenère. Foi escolhida a linguagem Python pela agilidade para prototipação dos algoritmos e familiaridade dos integrantes da equipe, iterando rapidamente sobre a codificação e decodificação das mensagens, possibilitando o enfoque no método de ataque.

Para a codificação da mensagem, primeiramente caracteres especiais da língua portuguesa , como letras com acentos ou o “ç”, foram mapeados para o caractere da tabela ASCII mais semelhante, ou seja, vogais com acento foram mapeadas para a vogal (e.g. “á” -> “a”) e o “ç” foi mapeado para a letra “c”. Essa decisão foi tomada pois a tabela de frequência de letras utilizada no método de ataque foi criada seguindo a mesma lógica, dessa forma a mensagem codificada pelo algoritmo possivelmente terá a frequência de letras mais próxima ao da tabela utilizada no método de ataque. Outro detalhe é que todas as letras maiúsculas foram substituídas por sua correspondente minúscula.

Em seguida, a mensagem informada é codificada seguindo os passos da cifra de Vigenère, com exceção de caracteres que não fazem parte do alfabeto, que foram ignorados durante a codificação e decodificação.

Por conta da linguagem escolhida para implementação, foi necessário converter os caracteres para o seus correspondentes na tabela ASCII para as manipulações, esse processo também ocorreu para o algoritmo de decodificação, sendo o principal ponto negativo observado pelos integrantes durante a implementação.

1. Decodificação

Para a decodificação da mensagem, não há necessidade de mapear caracteres especiais como na codificação, dessa forma a decodificação basicamente seguiu o algoritmo descrito na seção 2. Vale lembrar que assim como na codificação, pontuação e caracteres que não fazem parte do alfabeto foram ignorados.

2. Ataque

O método de ataque implementado se baseia no produto escalar de vetores e suas propriedades, tendo os seguintes passos:

1. Descobrir o tamanho da palavra chave;
2. Descobrir cada caractere da palavra chave com base no seu tamanho e frequência das letras do idioma;
3. Decodificar a mensagem utilizando a palavra chave descoberta no passo anterior.

Para descobrir o tamanho da palavra chave, foi utilizada a propriedade de produto escalar de vetores que tem valor máximo quando os vetores são paralelos [Mathematics 2013][Theoretically 2015]. Para aplicamos essa propriedade ao método de ataque, a mensagem codificada foi comparada com ela mesma deslocada uma posição para a direita, contando o número de caracteres iguais para o mesmo índice, como mostrado abaixo:

Figura 1

Chamamos de coincidência quando um caractere aparece na mesma posição em ambas mensagens. O produto escalar de vetores tem valor máximo quando dois vetores são paralelos, no caso do ataque a cifra de Vigenère, essa propriedade se aplica ao vetor de frequência das letras deslocadas pela codificação, sendo que quando as letras que foram codificadas com a mesma letras da palavra chave se alinham, a probabilidade de coincidências é maximizada [Stange 2020]. Dessa forma podemos observar onde as coincidências tiveram valores de pico locais e calcular a distância entre eles, sendo a distância um bom chute para o valor do tamanho da palavra chave [Cox 2022].

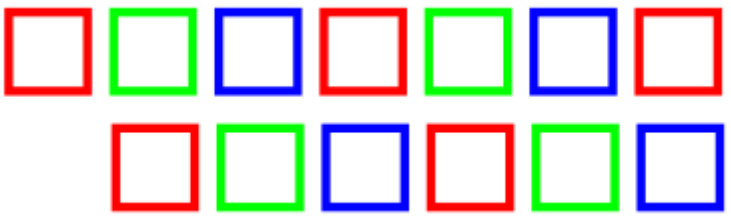

Figura 2. Exemplo para quando chance de coincidências é menor

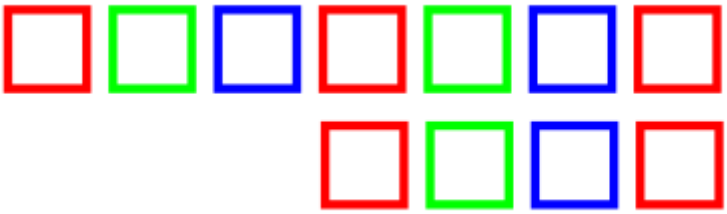

Figura 3. Exemplo para quando chance de coincidências é maior

De posse do tamanho da mensagem, a próxima etapa é descobrir os caracteres individuais da palavra chave, para isso iremos selecionar todos os caracteres da mensagem codificada com a mesma letra, contando a frequência de cada letra dentre os selecionados formando um vetor de frequências. Idealmente esse vetor de frequências será igual ao vetor de frequência de letras do idioma deslocado, sendo o deslocamento correspondente a letra da palavra chave. Para descobrir qual foi o deslocamento, podemos utilizar a mesma propriedade do produto escalar, sendo que o produto escalar dos vetores de frequência tem o valor máximo quando as frequências das letras estão alinhadas.

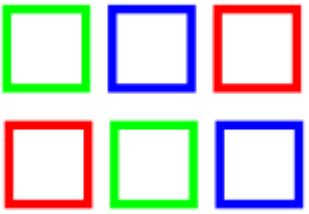

Figura 4. Exemplo para quando o produto escalar das frequências não é máximo

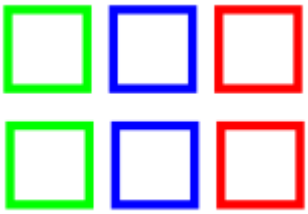

Figura 5. Exemplo para quando o produto escalar das frequências é máximo

Após descobrir cada caractere da palavra chave, basta aplicar os passos descritos na decodificação da cifra de Vigenère, como descrito na seção 2.

3. Resultados

Os algoritmos de codificação e decodificação tiveram resultados dentro do esperado, sendo que a mensagem codificada com um pode ser decodificada com o outro sem problemas, salvo os detalhes de implementação descritos nas respectivas seções dos mesmos (mapeamento de caracteres especiais e pontuação).

O método de ataque para a cifra de Vigenère teve eficiência limitada, por se tratar de um algoritmo probabilístico. Em nossos testes, ele foi capaz de deduzir corretamente o tamanho e boa parte dos caracteres das palavras chaves, caso a quantidade de letras que

foram deduzidas erroneamente pelo algoritmo fosse pequena, a mensagem decodificada com a mesma ainda era legível porém não idêntica à mensagem original.

Observamos que o algoritmo falhou principalmente quando o tamanho da chave foi deduzido de forma errada ou quando a chave apresenta caracteres repetidos, como “arara” por exemplo. Para esses casos, o número de caracteres decodificado de forma errada é muito grande, impossibilitando a leitura da mensagem.

O código fonte está disponível no repositório do github no seguinte link: https://github.com/joao-victor-silva/vigenere-cipher.

4. Conclusões

A cifra de Vigenère foi uma evolução comparada a cifra de César, porém ainda apresenta é vulnerável a ataques de baseados na frequência de letras. Para a época que foi criada, a cifra era eficiente porém ficou defasada com a passagem do tempo. Sua eficiência pode ser melhorada aumentando o número de caracteres na palavra chave, tendo como impacto negativo a maior complexidade para codificar e decodificar a mensagem, tornando o uso do método menos prático.

5. Referências

[Stange 2020] Stange, K. (2020). Cryptanalysis of Vigenere cipher: not just how, but why it works. https://www.youtube.com/watch?v=QgHnr8-h0xI.

[Theoretically 2015] Theoretically. (2020). Vigenere Cipher - Decryption (Unknown Key). https://www.youtube.com/watch?v=LaWp_Kq0cKs.

[URFJ 2021] Universidade Federal do Rio de Janeiro. (2021). Decifrando Textos em Português.
https://web.archive.org/web/20210430083032/https://www.gta.ufrj.br/grad/06_2/alexandre/criptoanalise.html

[Mathematics 2013] Encyclopedia of Mathematics. (2013). Inner product. http://encyclopediaofmath.org/index.php?title=Inner_product&oldid=29549

[Cox 2022] Cox, G. (2022). Peak Detection in a Measured Signal. https://www.baeldung.com/cs/signal-peak-detection.