# Segurança computacional - Estudos na cifra de Vigenère

João Tito do Nascimento Silva (18/0123301)

Agosto de 2022

## 1 Objetivo

Este trabalho tem o objetivo de estudar a implementação e a criptoanálise da cifra de Vigenère. Almejamos não somente implementar, mas entender os melhores princípios de implementação e design de ferramentas de criptografia.

O presente documento tem como objetivo detalhar a metodologia e as decisões realizadas na implementação, que tem duas funções básicas: cifrar/decifrar textos com a cifra de Vigenère e recuperar a senha (chave) utilizada em um texto cifrado.

## 2 Teconologias utilizadas

A implementação foi realizada em Python 3.8.10 com uma simples interface gráfica baseada em terminal utilizando o pacote *pynput*. Os componentes foram desenvolvidos no código, por meio de manipulação de strings, e as cores e detalhes especiais (cores e efeitos) foram implementados por meio de *ANSI Escape Sequences*.

## 3 Implementação da cifração

A implementação da cifração levou em conta dois guias principais:

1. A cifra gerada não deve dar informações desnecessárias sobre o texto aberto original.
2. A cifra gerada pela ferramenta desenvolvida deve ser o mais "padrão"o possível, de modo que ferramentas similares possam decifrá-la sem maiores problemas sem que seja necessário o conhecimento da implementação.

As decisões causadas por esses guias são apresentadas a seguir em mais detalhes.

### 3.1 Evitar vazamento de informações desnecessárias

Algumas informações não são de fato necessárias para a decifração de uma cifra e a compreensão do texto decifrado mas que podem facilitar o trabalho de atacantes.

Um exemplo muito claro disso são os espaços em branco. Ao manter espaços em branco no texto cifrado, revelaríamos o tamanho das palavras no texto original. Com isso, um atacante poderia, por exemplo, descobrir mais facilmente a linguagem utilizada. Além disso, facilitaríamos o reconhecimento de padrões de repetição de digramas e trigramas (método de Kasiski), revelando mais informações sobre a chave utilizada. De forma similar, manter a diferenciação de letras maiúsculas e minúsculas pode se tornar um problema.

Outro exemplo é a acentuação e uso de caracteres especiais nas palavras. Em português, várias palavras incluem acentos ou cedilhas, e sua presença em um texto já auxiliaria um atacante a determinar que o texto original está nessa linguagem. O atacante também teria mais facilidade de descobrir algumas partes da chave utilizada consultando um dicionário ou banco de palavras em claro procurando por ocorrências dos acentos ou cedilhas. O mesmo acontece em outras linguagens, como o alemão, com caracteres diferentes ou acentuações diferentes. Portanto, remover acentuação e caracteres especiais é essencial para evitar o vazamento de informações.

Por fim, a presença de números no texto cifrado também poderia auxilixar um atacante, uma vez que o contexto poderia revelar a que palavras claras as palavras em texto cifrado correspondem. Um atacante que conhece, por exemplo, que um texto se refere a uma data na forma "XX de agosto", ao ver o padrão "29adcefg..."poderia entender que o número 29 marca o início da informação de data, possivelmente auxiliando no ataque. Há ainda outras desvantagens em manter números no texto cifrado a serem tratadas na seguinte subseção.

Portanto, a implementação realizada retira todos os caracteres que não estão listados no alfabeto considerado para a linguagem, espaços em branco e números. Para evitar que essa extração dificulte excessivamente a leitura dos textos após a decifração, quando há acentos é utilizada uma técnica de normalização chamada *Normalization Form Canonical Decomposition (NFD)* em combinação com um simples tratamento que mantem as letras originais sem os acentos (por exemplo, "á"vira "a"). O texto também é transformado para caixa alta.

## 3.2 Manter um "padrão"na implementação

Na cifração, é importante evitar que a decifração do texto resultante dependa de detalhes da implementação da ferramenta, de forma que outras ferramentas similares possam decifrar o texto. Isso é uma boa premissa e sua formalização está inclusa nos argumentos a favor dos princípios de Kerckhoff: "[...]for large-scale deployment it is significantly easier for users to all rely on the same encryption algorithm/software (with different keys) than for everyone to use their own custom algorithm"[1].

Levando isso em conta, evitou-se criar alfabetos incomuns (o alfabeto utilizado é o alfabeto comum e outros caracteres são ignorados). Assim, números realmente são ignorados e não são tratados, e o usuário deve utilizar outras formas de representar números com letras (por exemplo, escrever por extenso ou com números romanos).

# 4 Implementação da decifração

Para a decifração, a premissa considerada mais importante é facilitar a utilização pelo usuário. Nesse sentido quando há espaços no texto cifrado, os espaços são mantidos no texto decifrado (mesmo que talvez fosse melhor que eles não estivessem no texto cifrado, como comentado na seção anterior). Também é preservado o *case* das palavras, de modo que letras maiúsculas no texto cifrado continuam maiúsculas no texto decifrado e vice-versa.

Dessa forma, o texto decifrado pode ser mais claro para o usuário, mesmo que essas informações provavelmente não devessem ter sido mantidas no texto cifrado durante a cifração.

Também assumimos que o método de cifração utilizado é "padrão", na forma comentada na seção anterior, e que o texto cifrado pode ser decifrado de forma simples e direta, sem ter que realizar análises no texto cifrado antes da decifração e sem necessidade de conhecimento de outras implementações.

# 5 Implementação da criptoanálise

A criptoanálise foi realizada com base em um ataque descrito no livro de Katz e Lindell[1], que não é baseado em observar se o texto decifrado "faz sentido"com um dicionário de palavras, mas em analisar estatisticamente quais chaves são mais prováveis.

Inicialmente, são decididas a linguagem e o tamanho da chave. A partir daí, a chave é considerada em cada caractere individualmente, como uma composição de várias cifras de deslocamento, ao invés de ser analisada como um todo em um ataque de força bruta. Para cada caractere da chave são estimadas quais chaves de deslocamento são mais prováveis.

Primeiramente, são testados vários tamanhos de chaves diferentes para deduzir a linguagem e o tamanho da chave utilizada. Para cada tamanho de chave testado, divide-se o texto cifrado em "fatias"que são cifradas cada uma com um dos caracteres da chave. Assim, se a chave tem tamanho $k$ e o i-ésimo caractere do texto cifrado for $c_i$, a primeira fatia é composta dos caracteres $c_0, c_{0+k}, c_{0+2k}, ...$, a segunda fatia pelos caracteres $c_1, c_{1+k}, c_{1+2k}, ...$ e assim por diante, até que todas as fatias possíveis tenham sido listadas. Após isto, cada fatia é analisada estatisticamente nas frequências dos caracteres. As frequências são calculadas e é construída uma medida que será chamada neste trabalho de *característica*. A característica do texto é definida da seguinte forma: se $f_i$ é a frequência do i-ésimo caractere do alfabeto, $N$ o tamanho do alfabeto e $C$ é a característica, então

$$C = \sum_{i=0}^{N} f_i^2$$

A mesma medida é calculada para o alfabeto, a partir das frequências conhecidas das letras em textos comuns da linguagem analisada. Por fim, os resultados de cada fatia de texto e da linguagem são comparados e agregados. Desse teste, ao se utilizar várias linguagens diferentes, obtemos uma estimativa de qual é a linguagem mais provável e também obtemos uma estimativa de quais tamanhos de chave são mais prováveis observando quais diferenças de características são menores.

As opções geradas são então apresentadas em uma interface para o usuário, na qual ele pode manipular as chaves mais prováveis ou alterar a linguagem de forma a rapidamente fazer testes e observar o resultado de suas alterações sobre o texto quebrado (decifrado com a chave escolhida).

Cada vez que o usuário escolhe um tamanho de chave, são recalculadas as chaves mais prováveis. O método com que estas são calculadas é similar ao teste para descobrir o tamanho de chave, exceto que dessa vez, ao invés de testar vários tamanhos de chave e várias linguagens, testamos vários possíveis caracteres da chave para cada fatia do texto e calculamos uma "característica combinada"com as frequências da linguagem, definida como:

$$C_{comb} = \sum_{i=0}^{N} f_i * a_i$$

Onde $f_i$ é a frequência do i-ésimo caractere do alfabeto na fatia e $a_i$ é a frequência do i-ésimo caractere do alfabeto na linguagem utilizada. Ao comparar a característica combinada para a fatia com a característica da linguagem, obtemos uma visão de qual é a chave mais provável para cada fatia, sem necessidade de testar todas as chaves. Realizamos $N * k$ testes, onde $N$ é o tamanho do alfabeto e $k$ é o tamanho da chave, ao invés de $N^k$ testes possíveis para um ataque de força bruta puro.

# Referências

[1] Jonathan Katz e Yehuda Lindell. *Introduction to Modern Cryptography. Second edition.* CRC Press, 2015.