CIC0201 - Segurança Computacional - Turma 01

Saulo Oliveira de Freitas - 211000176

Cifra de Vigenère - codificador e decodificador

Nesse projeto serão abordados conceitos de criptografia, especificamente a cifra de Vigenère, aplicando-os para realizar a contrução de um codificador e decodificador de mensagens textuais.

Cifra de Vigenere - conceito

A cifra de Vigenère é um método de criptografia de substituição polialfabética. Em uma cifra de substituição polialfabética, cada letra da mensagem é substituída por outra letra, dependendo de uma chave. A cifra de Vigenère usa uma chave que é um alfabeto deslocado ciclicamente. A cifra de Vigenère é mais segura do que as cifras de César simples porque possui maior entropia, o que torna difícil adivinhar a senha. No entanto, ainda é possível quebrar a cifra se a senha for curta ou se o texto cifrado for repetido várias vezes.

Codificador

O processo de codificação consiste em duas funções: spawn_key e cipher, que são usadas para cifrar uma mensagem usando a cifra de Vigenère.

```
def spawn_key(msg,key):
   key_len = len(key)
   keystream = ''.join(key[i % key_len].upper() if char.isalpha() else ' '
   for i, char in enumerate(msg))
   return keystream

def cipher(msg, key):
   if not isinstance(msg, str) or not isinstance(key, str):
       raise TypeError('message and key must be strings')
   if not msg or not key:
       raise ValueError('message and key must not be empty')


   message = unidecode(msg)
   keystream = spawn_key(msg, key)
   ciphered_msg = ''.join(
       chr(((ord(char.upper()) + ord(key)) % 26) + ord('A'))
       if char.isalpha()
       else char
       for char, key in zip(message, keystream)
   )
   return ciphered_msg
```

A função spawn_key é responsável por gerar a keystream, que é uma sequência de caracteres repetidos da chave fornecida, com o mesmo tamanho da mensagem. A função itera sobre cada caractere da mensagem e, se o caractere for uma letra, obtém o caractere correspondente na chave (usando o operador de módulo % para repetir os caracteres da chave) e converte-o para letra maiúscula. Se o caractere não for uma letra, é substituído por um espaço em branco. A função retorna a keystream resultante.

A função cipher verifica se a mensagem e a chave são strings e não são vazias. Se não forem, são lançadas exceções de tipo e valor. Em seguida, a função chama a função unidecode para remover qualquer caractere acentuado da mensagem, convertendo-os em caracteres não acentuados. Em seguida, chama a função spawn_key para gerar a keystream. A mensagem e a keystream são então percorridas em pares usando a função zip. Para cada par de caracteres, se o caractere for uma letra, ele é convertido para letra maiúscula e é aplicada a fórmula da cifra de Vigenère para obter o caractere cifrado correspondente. Se o caractere não for uma letra, ele é mantido inalterado. Todos os caracteres cifrados são concatenados em uma única string e retornados como resultado da função.

Decodificador

A função decipher é implementada para decifrar uma mensagem cifrada usando a cifra de Vigenère, dada uma mensagem cifrada e a chave correspondente.

```
def decipher(cypher_msg, key):
  keystream = spawn_key(cypher_msg, key)
  decipher_msg = ''.join([chr(((ord(char.upper()) - ord(key.upper()) + 26) % 26)
   + ord('A')) if char.isalpha() else char
   for char, key in zip(cypher_msg, keystream)])
  return decipher_msg
```

A função começa chamando a função spawn_key para gerar a keystream, que é uma sequência repetida da chave, do mesmo tamanho da mensagem cifrada.

Em seguida, a função percorre cada caractere da mensagem cifrada e da keystream usando a função zip. Para cada par de caracteres, se o caractere for uma letra, ele é convertido para letra maiúscula. Em seguida, a fórmula de decifragem da cifra de Vigenère é aplicada para obter o caractere decifrado correspondente. A fórmula subtrai o valor do caractere da chave do valor do caractere cifrado, adiciona 26 e então aplica o operador de módulo % para garantir que o resultado esteja no intervalo de 0 a 25 (representando as 26 letras do alfabeto). O resultado é então convertido novamente para um caractere ASCII, adicionando o valor de 'A' (o código ASCII da primeira letra maiúscula).

Se o caractere não for uma letra, ele é mantido inalterado.

Todos os caracteres decifrados são concatenados em uma única string e retornados como resultado da função.

Quebra de Cifra

O método de quebra de cifra utilizado na função break_cipher é conhecido como análise de frequência. O intuito é explorar a distribuição de frequência das letras em um texto para tentar determinar a chave utilizada na cifra.

A metodologia utilizada na função break_cipher é baseada na coincidência de índice. O índice de coincidência é uma medida estatística que indica a probabilidade de duas letras escolhidas aleatoriamente em um texto serem iguais. Em um texto em que as letras estão distribuídas uniformemente, o índice de coincidência é próximo de 0.0385.

A estratégia utilizada na função é a seguinte:

- Determinar o tamanho da chave:
  - A função percorre diferentes comprimentos possíveis da chave e calcula o índice de coincidência médio para cada um. Se o índice de coincidência médio for maior que um determinado limiar (0.06 neste caso), considera-se que esse comprimento pode ser o tamanho da chave.
- Quebrar a cifra para cada tamanho da chave possível
  - Para cada tamanho de chave possível, a função divide o texto cifrado em substrings, onde cada substring contém os caracteres cifrados que estão a uma distância equivalente ao tamanho da chave. Em seguida, a função analisa a distribuição de frequência dos caracteres em cada substring e tenta encontrar o caractere da chave correspondente à letra mais frequente em cada substring.
- Descriptografar a mensagem
  - Com a chave possível, a função utiliza a função decipher para descriptografar a mensagem cifrada.