Instructions for Authors of SBC Conferences Papers and Abstracts

Tobias Trein

[1]Escola Politécnica - PUCRS
Av. Ipiranga, Prédio 30 - Bloco F, 6681 - Partenon, Porto Alegre - RS, 90619-900

Abstract. *This article describes the cryptanalysis of a ciphertext using the Index of Coincidence method to determine the key length of the Vigenère cipher and then decrypt the text. The technique exploits letter frequencies in Portuguese and English to identify patterns in the ciphertext. The developed algorithm automates the key length calculation and text decryption, providing an effective solution for breaking the cipher. The work includes the source code and a descriptive text according to its functionality.*

Resumo. *Este artigo descreve a criptoanálise de um texto cifrado utilizando o método de Índice de Coincidência para determinar o tamanho da chave do cifrador Vigenère e, em seguida, decifrar o texto. A técnica explora as frequências de letras em Português e Inglês para identificar padrões no texto cifrado. O algoritmo desenvolvido automatiza o cálculo do tamanho da chave e a decifração do texto, proporcionando uma solução eficaz para a quebra do cifrador. O trabalho apresenta o código fonte e um texto descritivo conforme a sua funcionalidade*

1. Introdução

A cifra de Vigenère é um método de criptografia polialfabético, ou seja, utiliza várias cifras de César diferentes, baseadas nas letras de uma palavra-chave. O processo de cifragem consiste em repetir a palavra-chave até alcançar o tamanho do texto a ser cifrado e, então, deslocar cada letra do texto original de acordo com a letra correspondente na palavra-chave. Dessa forma, a cifra de Vigenère torna mais difícil a análise estatística do texto cifrado, pois cada letra pode ser cifrada de várias maneiras diferentes ao longo do texto, aumentando a segurança do sistema.

Apesar de ter sido criada no século XVI e decifrada no ano 1863, permanecendo cerca de 300 anos sem solução, a cifra de Vigenère ainda é um desafio interessante para a criptoanálise. A sua resistência à quebra se deve à sua natureza polialfabética, que dificulta a aplicação de métodos simples de análise estatística. O método do Índice de Coincidência, utilizado neste trabalho, é uma das abordagens eficazes para a quebra da cifra, pois explora padrões de repetição no texto cifrado para determinar o tamanho da chave utilizada.

2. Análise de abordagens

Dos métodos de soluções conhecidas se descatam duas abordagens:

2.1. Teste de Kasiski

O Teste de Kasiski é uma técnica de criptoanálise usada para quebrar cifras polialfabéticas, como a cifra de Vigenère. A abordagem se baseia na identificação de padrões de repetição no texto cifrado, que podem revelar o tamanho da chave utilizada. O processo envolve encontrar sequências de caracteres repetidos no texto cifrado e calcular as distâncias entre elas. Ao identificar padrões de repetição que ocorrem com frequência, é possível inferir o comprimento da chave, uma vez que esses padrões podem ser causados por múltiplas cifragens usando a mesma parte da chave.

2.2. Índice de Coincidência

Por outro lado, o Índice de Coincidência é uma técnica que se baseia na análise estatística das frequências das letras no texto cifrado. Em idiomas como o Português e o Inglês, algumas letras são mais frequentes do que outras. No caso do Índice de Coincidência, calcula-se a probabilidade de que duas letras escolhidas aleatoriamente em um texto sejam iguais. Em textos não cifrados, esse índice tende a ser próximo de um valor específico, que varia de acordo com o idioma. No entanto, em textos cifrados com a cifra de Vigenère, o índice de coincidência tende a se aproximar do índice de coincidência do idioma original, ajudando a identificar o tamanho da chave.

3. Solução

A solução proposta se baseia na solução via Indice de coincidência. A solução pode ser vista no vídeo `https://youtu.be/mrITq7lvOe4` Ela é iniciada pelo descobrimento do tamanho da chave utilizada. Para isso, criamos slices da string original e calculamos o IC para cada uma das slices. No código criado, são geradas slices de 1 a 10 pulos de caractere.

A fórmula do Índice de Coincidência (IC) é dada por:

$$IC = \frac{\sum_{i=0}^{c} n_i(n_i - 1)}{N(N-1)}$$

Onde:

- f_i é a frequência absoluta da letra i no texto;

- N é o total de letras no texto;

Representada no código como:

```python
def indice_coincidencia(texto):
    freq = {}
    for c in texto:
        if c.isalpha():
            if c in freq:
                freq[c] += 1
            else:
                freq[c] = 1
    total = sum(freq.values())
    ic = sum((freq[c] * (freq[c] - 1)) / (total * (total - 1)) for c in
    freq)
    return ic
```

Com a lista de todas os ICs dos slices, verifica o slice com maior proximidade com algum dos idiomas possíveis, português ou inglês. O tamanho do slice é o tamanho da chave encontrada. O trecho do slicing e a descoberta do tamanho da chave é definida no código como:

```
    for tamanho_chave in range(1, 11):  # Itera sobre chaves de 1 a 10 caracteres
        textoselec = texto_cifrado[::tamanho_chave]  # Gera as chaves de acordo com o tamanho
        frequencia_textos[tamanho_chave] = calcular_frequencia_letras(textoselec)
        similaridades = {}
        ic_texto = indice_coincidencia(textoselec)
        for lingua, ic_lingua in linguagens.items():
            similaridade = similaridade_ic(ic_texto, ic_lingua)
            similaridades[lingua] = similaridade
        lingua_mais_proxima = min(similaridades, key=similaridades.get)
        resultados.append((tamanho_chave, ic_texto, lingua_mais_proxima))

    # Encontra o tamanho de chave com o  ndice  de coincid ncia mais pr ximo de alguma das l nguas
    tamanho_chave_mais_proximo = min(resultados, key=lambda x: min(abs(x[1] - v) for v in linguagens.values()))
```

Ao encontrar o tamanho da chave, Para cada slice calcula o percentual da frequência das letras e compara com as frequências das letras da língua. Assim, conseguimos descobrir o deslocamento para cada slice e seu caractere correspondente.

```
frequencias_portugues = {
    'a': 14.634, 'b': 1.043, 'c': 3.882, 'd': 4.992, 'e': 12.570,
    'f': 1.023, 'g': 1.303, 'h': 1.281, 'i': 6.186, 'j': 0.879,
    'k': 0.015, 'l': 2.779, 'm': 4.738, 'n': 4.446, 'o': 9.735,
    'p': 2.523, 'q': 1.204, 'r': 6.530, 's': 6.805, 't': 4.336,
    'u': 3.639, 'v': 1.575, 'w': 0.037, 'x': 0.453, 'y': 0.006, 'z': 0.470
}

frequencias_ingles = {
    'a': 8.167, 'b': 1.492, 'c': 2.782, 'd': 4.253, 'e': 12.702,
    'f': 2.228, 'g': 2.015, 'h': 6.094, 'i': 6.966, 'j': 0.153,
    'k': 0.772, 'l': 4.025, 'm': 2.406, 'n': 6.749, 'o': 7.507,
    'p': 1.929, 'q': 0.095, 'r': 5.987, 's': 6.327, 't': 9.056,
    'u': 2.758, 'v': 0.978, 'w': 2.360, 'x': 0.250, 'y': 1.974, 'z': 0.074
}
```

```
    chave= ''

    for n in range(nmbchave):
        texto = texto_cifrado[n::nmbchave]
        freq_texto = calcular_frequencia_letras(texto)
        frequencias_texto_ordenadas = ordenar_frequencias(freq_texto)
        deslocamento = ord(next(iter(frequencias_texto_ordenadas))) - ord(next(iter(frequencias_lingua_ordenadas)))

```

```
        # Transforma o deslocamento em um caractere
        chave += chr((deslocamento % 26) + ord('a'))

    print(f"chave encontrada: {chave}")
    return chave
```

Agora com a chave, conseguimos decifrar o texto, basta apenas percorrermos cada letra do texto e aplicar o deslocamento correspondente da letra da chave para decifrá-la.

```
def decrypt_vigenere(ciphertext, key):
    key_length = len(key)
    decrypted = []
    for i, char in enumerate(ciphertext):
        key_char = key[i % key_length]
        if char.isalpha():
            decrypted_char = chr(((ord(char) - ord(key_char)) % 26) +
   ord('A'))
        else:
            decrypted_char = char
        decrypted.append(decrypted_char)
    return ''.join(decrypted)
```

4. Considerações Finais

Podemos afirmar que a abordagem aplicada no algoritmo de criptoanálise do Índice de Coincidência para a cifra de Vigenère é uma das mais eficazes e amplamente aplicadas na quebra desse tipo de cifra. A técnica se mostrou eficiente na determinação do tamanho da chave, permitindo a decifração automática do texto cifrado. A automatização desse processo é especialmente relevante em contextos onde a análise manual seria inviável devido ao tamanho do texto ou à necessidade de rapidez na obtenção do resultado. Portanto, a utilização do Índice de Coincidência representa uma contribuição significativa para a criptoanálise de cifras polialfabéticas, como a cifra de Vigenère, ao permitir a quebra desses códigos de forma mais rápida e eficiente.