TRABALHO DE IMPLEMENTAÇÃO 1: CIFRA DE VIGENÈRE E CRIPTOANÁLISE

Eduardo Marques - 211021004 Yan Tavares - 202014323

UNIVERSIDADE DE BRASÍLIA
DISCIPLINA: CIC0201 - Segurança Computacional
PERÍODO: 2025/1
PROF[a]. PRISCILA SOLIS

1 INTRODUÇÃO

A cifra de Vigenère, polialfabética do século XVI, usa uma palavra-chave para alternar alfabetos de cifragem, dificultando a análise de frequência. Este trabalho implementa um sistema de cifragem/decifragem Vigenère e um módulo de criptoanálise para quebra da cifra por análise de frequência, usando a linguagem C e bibliotecas padrão.

2 PARTE I: CIFRADOR/DECIFRADOR DE VIGENÈRE

2.1 Fundamentação Teórica

A cifra opera no alfabeto de 26 letras. A chave, repetida se necessário, determina o deslocamento para cada letra do texto. Fórmulas:

- Cifragem: $C_i = (P_i + K_i) \mod 26$
- Decifragem: $P_i = (C_i - K_i + 26) \mod 26$

(A=0, ..., Z=25). Não alfabéticos são mantidos.

2.2 Descrição da Implementação

O arquivo `main.c` contém um menu interativo para:

Cifragem (`vigenere_encrypt`, `encrypt_menu`):

- Entrada de texto claro (manual/arquivo) e chave (alfabética, não vazia).
- A função `vigenere_encrypt` aplica a fórmula, preservando caixa e não alfabéticos.
- Exibe e opcionalmente salva o criptograma.

Decifragem (`vigenere_decrypt`, `decrypt_menu`):

- Entrada de criptograma (manual/arquivo) e chave.
- A função `vigenere_decrypt` aplica a fórmula inversa.

- Exibe e opcionalmente salva o texto claro.

2.3 Preservação de Caracteres Não Alfabéticos

A implementação mantém caracteres não alfabéticos (espaços, pontuação, números, etc.) sem alteração durante o processo de cifragem/decifragem. Esta abordagem é válida por várias razões:

1. **Compatibilidade com a definição original**: A cifra de Vigenère foi originalmente concebida para operar apenas sobre caracteres alfabéticos. Caracteres não alfabéticos estão fora do escopo da transformação matemática da cifra [4].
2. **Preservação da estrutura do texto**: Manter sinais de pontuação e espaçamento facilita a leitura do texto decifrado e mantém sua estrutura sintática intacta [5].
3. **Utilidade prática**: Em aplicações reais, a preservação de caracteres especiais como números, símbolos e formatação é frequentemente necessária para manter a funcionalidade e contexto da mensagem [6].
4. **Consistência com implementações padrão**: Implementações modernas da cifra de Vigenère tradicionalmente mantêm caracteres não alfabéticos inalterados, estabelecendo uma prática padrão.

3 PARTE II: ATAQUE DE RECUPERAÇÃO DE SENHA

3.1 Fundamentação Teórica

A repetição da chave produz periodicidade. O Índice de Coincidência (IC) global indica se o texto é cifrado. Para descobrir o tamanho m da chave, divide-se o criptograma em m colunas e calcula-se o IC médio de cada coluna; o m cujo valor mais se aproxima do IC do idioma é o mais provável.

Com m conhecido, cada coluna forma uma subturma monalfabética. Oferecemos dois métodos para estimar o deslocamento de cada subturma, escolhidos pelo usuário no menu:

1. **Teste Qui-Quadrado**: calcula $\chi^2 = \sum(O_i - E_i)^2/E_i$ entre as frequências observadas O e esperadas E do idioma.
2. **Índice de Coincidência Mútua (ICM)**: calcula o IC entre a subturma, rotacionada por cada um dos 26 possíveis deslocamentos, e a distribuição de frequência do idioma; o deslocamento que maximiza o ICM é escolhido.

3.1.1 Determinação do Tamanho da Chave

O Índice de Coincidência (IC) mede a probabilidade de duas letras aleatórias serem iguais.

$$\mathrm{IC} = \frac{\sum_i f_i(f_i - 1)}{N(N-1)} \tag{1}$$

Valores esperados: ~ 0.0761 (PT), ~ 0.0667 (EN), ~ 0.0385 (aleatório). Dividindo o criptograma em m colunas (tamanho da chave), o IC médio das colunas deve aproximar-

se do IC do idioma. O m que melhor satisfaz essa condição é o tamanho provável da chave.

3.1.2 Determinação das Letras da Chave

Com o tamanho m, o criptograma normalizado é dividido em m subsequências. Para cada uma:

- Calcula-se a frequência das letras.
- Testam-se 26 deslocamentos (letras da chave 'a'-'z').
- O deslocamento que torna a distribuição de frequência da subsequência mais similar à do idioma, medido pelo Teste do Qui-Quadrado, $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$ ou o ICM, indica a letra da chave.

3.2 Descrição da Implementação do Ataque

O módulo de ataque (`attack_menu` e funções auxiliares) em `main.c`:

Entrada: Criptograma (manual/arquivo) e idioma (PT/EN). Normalização do texto (`clean_text_to_lower`).

IC Global: Cálculo para avaliação inicial.

Tamanho da Chave (`find_key_length`):

- Testa tamanhos de 1 a `MAX_KEY_LENGTH_TO_TRY` (padrão 20).
- Para cada m, calcula o IC médio das subsequências (`average_ic_for_key_length`).
- O m com IC médio mais próximo do `target_ic` do idioma (0.0761384 PT, 0.066699 EN) é escolhido. Opção de entrada manual do tamanho.

Letras da Chave (`recover_key`):

- Para cada posição da chave, extrai a subsequência.
- `find_likely_shift_chi_squared` testa os 26 deslocamentos, retornando o que minimiza χ^2.

Decifragem: O criptograma original é decifrado com a chave recuperada (`vigenere_decrypt`). Exibe e opcionalmente salva texto e relatório. Frequências de letras (PT/EN) são de fontes da Wikipedia, conforme especificado.

3.3 Resultados e Discussão dos Testes da Parte II

Testamos três arquivos disponíveis em `https://github.com/yantavares/segcomp/tree/main/vigenere`. Para cada um, executamos ambos os métodos.

ptbr.txt (168 palavras, PT, chave "literatura"): Qui-Quadrado e ICM recuperaram automaticamente o tamanho da chave (10) e a chave correta, produzindo texto claro perfeitamente legível.

ptbr2.txt (26 palavras, PT, chave "teste"): O Qui-Quadrado falhou em identificar a chave mesmo após fixarmos manualmente o tamanho (5); o ICM, com o mesmo tamanho manual, recuperou "teste" e decifrou o texto.

en.txt (84 palavras, EN, chave "literatura"): Ambos os métodos determinaram automaticamente tamanho 10 e recuperaram a chave, revelando o texto original.

Síntese: Para textos médios ou longos, Qui-Quadrado e ICM têm desempenho idêntico; em textos curtos, o ICM mostrou-se mais robusto.

4 COMPILAÇÃO E USO

4.1 Requisitos

- Compilador C (GCC recomendado)
- Bibliotecas Padrão C (stdio, stdlib, string, ctype, math)

4.2 Compilação

`gcc -o vigenere main.c -lm` (A flag -lm linka a biblioteca matemática)

4.3 Uso

`./vigenere`
Menu: 1.Cifrar, 2.Decifrar, 3.Ataque, 0.Sair. Siga as instruções.

5 CONCLUSÃO

A implementação da cifra de Vigenère e do ataque por análise de frequência demonstrou conceitos de criptografia clássica. O cifrador/decifrador e o módulo de ataque (IC para tamanho da chave, Qui-Quadrado para letras) foram eficazes, com precisão dependente do comprimento e características do criptograma. Melhorias no código (tratamento de entrada, validações, Qui-Quadrado) resultaram em um programa robusto e preciso. A geração de relatórios é útil para análise. O trabalho reforça a importância da estatística na quebra de cifras clássicas.

Referências

[1] SOLÍS, P. (2023). CIC0201-Segurança Computacional - 2023/1: Trabalho de Implementação 1 - Cifra de Vigenère.

[2] WIKIPEDIA. Frequência de letras. Disponível em: `https://pt.wikipedia.org/wiki/Frequ%C3%AAncia_de_letras`. Acesso em: 17/05/2025

[3] WIKIPEDIA. Letter frequency. Disponível em: `https://en.wikipedia.org/wiki/Letter_frequency`. Acesso em: 17/05/2025.

[4] KAHN, D. (1996). The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner.

[5] STAMP, M. (2011). Information Security: Principles and Practice. Wiley.

[6] PAAR, C., PELZL, J. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer.