

Relatório Trabalho 1 - Cifra de Vigenére

Aluno: Felipe Renato Alvarenga Batalha (19/0012862)Turma Noturna

2 de Outubro de 2023

1 Cifrador/Decifrador

A primeira parte desse trabalho consistiu na elaboração de um cifrador e um decifrador que operem de acordo com [2]. Conforme a descrição da cifra de Vigenére fornecida, para cifração é necessária uma mensagem e uma chave, e o mesmo vale para a decifração, com a diferença que a mensagem é cifrada e queremos efetuar o processo inverso. A codificação do trabalho foi realizada na linguagem C, que estava entre uma das linguagens indicadas para a realização do trabalho.

```
1
2 if (alphabet(key_string[i]) + alphabet(message[i])
   > 26)
3 {
4     cyphertext[i] = ascii(alphabet(key_string[i]) +
        alphabet(message[i]) - 26);
5 }
6 else
7 {
8     cyphertext[i] = ascii(alphabet(key_string[i]) +
        alphabet(message[i]));
9 }
```

O trecho de código acima corresponde ao processo de definição de cada letra da cifra com relação a chave, sendo message a mensagem a cifrar, o keystring a string com repetição da chave e o cyphertext o texto cifrado com

base na keystream. A função `alphabet` é uma função que posiciona o caractere no alfabeto entre 0 e 26, e a função `ascii` simplesmente converte o número obtido da operação em uma letra com base na tabela `ascii`. A condicional se deve ao fato de que na cifra de Vigenere, ao ultrapassar a letra "z", devemos retornar ao início do alfabeto, o que é feito com a subtração de 26.

```
1 if (alphabet(cyphertext[i]) -  
    alphabet(key_string[i]) < 0)  
2     {  
3         message[i] =  
            ascii(alphabet(cyphertext[i]) -  
                alphabet(key_string[i]) + 26);  
4     }  
5     else  
6     {  
7         message[i] =  
            ascii(alphabet(cyphertext[i]) -  
                alphabet(key_string[i]));  
8     }
```

O código acima é o referente a decifração, que utiliza também das funções `alphabet` e `ascii`, mas realiza o processo inverso, ao da cifração, e por isso, a preocupação na condicional é que o resultado da operação preceda a letra "a", e assim ao invés de subtrair 26, soma-se.

2 Ataque

O ataque realizado foi baseado em [3], que avalia a frequência de repetição de diferentes sequências de letras dentro do texto cifrado, para então deduzir o tamanho da chave. É importante ressaltar que diferente da etapa anterior, nesse cenário não há acesso a chave ou qualquer informação da mesma. Na implementação efetuada, o usuário deve visualizar na tela os trigramas e seus respectivos divisores e decidir manualmente um tamanho para a chave.

Uma vez que tenha sido selecionado o comprimento da chave, o usuário pode então verificar o comprimento da mesma utilizando o método indicado no [1] recomendado. Sendo assim, foi efetuado o mesmo proced-

imento visando equiparar as três letras mais frequentes da linguagem com as três letras mais frequentes em relação a cada posição da chave, conforme exemplo abaixo, de iteração com base na linguagem inglês:

```
1 for (int k = j; k < 26; k += key_size)
2     {
3         int a = k;
4         int e = k + 4;
5         int i = k + 8;
6
7         if (e > 25)
8         {
9             e -= 26;
10        }
11        if (i > 25)
12        {
13            i -= 26;
14        }
15        if (occurrences[a] + occurrences[e] +
            occurrences[i] > total)
16        {
17            total = occurrences[a] + occurrences[e]
                + occurrences[i];
18            topthree = a;
19        }
20    }
```

A implementação acima busca pela sequências de três letras com o mesmo espaçamento de "a", "e" e "i" com maior número de ocorrências total, equivalendo ao processo de shift adotado no vídeo, porém de forma automatizada.

3 Conclusão

A realização do trabalho permitiu verificar empiricamente como a cifra de Vigenère é vulnerável a ataques baseados em análises de frequência rela-

tivamente simples, que permitem deduzir a chave de cifração e consequentemente decifrar a mensagem.

Alguns fatores podem dificultar a realização do ataque, como por exemplo tamanhos de mensagem pequenos, que podem acabar não permitindo que tendências normalmente aplicáveis a linguagem da mensagem se manifestem, dificultando a análise de frequência. Outro ponto observado foi a questão do tamanho das chaves. Quanto maiores as chaves utilizadas para a cifração, mais difícil prever o tamanho da mesma, dado que a distância entre trigramas não indica diretamente o tamanho da chave de forma direta, e sim levanta um número de possibilidades com base nos divisores comuns entre os trigramas. Quanto maior a cifra, maior tende a ser o número de divisores comuns entre os trigramas dificultando a escolha de um tamanho de chave.

Por mais que algumas etapas do código desenvolvido no trabalho dependam da interpretação e interação humana, é possível vislumbrar alternativas automatizadas mais robustas e abrangentes. Em retrospecto, é possível substituir a constatação humana em algumas etapas do programa por tentativa e erro automatizadas em grande escala, indicando um grande problema quando se trata da utilização da cifra de Vigenère nos dias de hoje, dado que ataques completamente automatizados podem ser implementados.

References

- [1] Brian Veitch. Cryptography - breaking the vigenere cipher, Accessed on 2023-10-02. Online video.
- [2] Wikipedia. Cifra de vigenère, Accessed on 2023-10-02. Online; accessed on 2023-10-02.
- [3] Wikipedia. Método de kasiski, Accessed on 2023-10-02. Online; accessed on 2023-10-02.