Segurança Computacional - Trabalho 1

Alice da Silva de Lima - 18/0112601,
Carlos Eduardo de Oliveira Ribeiro - 18/0099094

Universidade de Brasília

1 Introdução

A criptografia consiste em tornar inelegível algum texto, de forma que apenas aquele que o criptografou saiba quais são os passos para recuperar o texto original. Esse tipo de técnica vem sendo utilizada desde muito tempo, por exemplo no contexto do Império Romano onde mensagens de cunho militar eram cifradas por espartanos utilizando um método específico denominado Cifra de César, em homenagem ao militar e político Júlio César que fazia o uso desta[1].

Nesse contexto, há diversas maneiras de cifrar mensagens. Além da famosa citada anteriormente, uma outra é a Cifra de Vegenère. Esta é composta por variações da Cifra de César e se baseia em uma palavra chave para transformar o texto.

Com o objetivo de por em prática os conhecimentos adquiridos na disciplina Segurança Computacional, foi realizada uma implementação da Cifra de Vegenère para cifrar, decifrar e encontrar a chave de um texto cifrado. Foi utilizada a linguagem C++ para o desenvolvimento do programa. A seção 2 explica o algoritmo desenvolvido.

2 Descrição do Algoritmo Desenvolvido

O programa possui um menu interativo que dispõe de três funcionalidades: a primeira é cifrar um texto regular especificando a chave; a segunda é decifrar um texto cifrado informando também a chave; a terceira é encontrar a chave de um texto cifrado.

Cifrador

Como dito antes, o método de Vegenère é uma composição de variações de cifra de César, enquanto numa cifra de César cada letra do alfabeto é deslocada por uma posição fixa, a de Vegenère utiliza uma tabela com todas as letras do alfabeto, cada uma escrita 26 vezes com um deslocamento de uma posição a esquerda, como é possível ver na Figura 1, essa tabela é conhecida como *tabula recta*.

Para realizar a cifração, uma palavra chave é escolhida, então faz-se uma cobertura de todo o texto que será cifrado. Por exemplo, se deseja-se cifrar a frase "seguranca computacional", com a chave "teste", o texto coberto ficaria

"testetest etestestetest", logo é feita uma comparação letra a letra dos dois textos, buscando na tabela o caractere correspondente e assim cifrando o texto. A tabela 1 mostra o processo de cifragem do exemplo:

Tabela 1: Processo de cifragem

Texto	s e g u r a n c a c o m p u t a c i o n a l
Chave	t e s t e t e s t e t e s t e s t e t e s t
Cifra	l i y n v t r u t g h q h n x t g a h r t p

A cifra de Vegenère pode ser representada por um cálculo, se mapearmos as letras de A-Z em números de 0 a 25, para cada caractere é possível escrever a criptografia como na fórmula a seguir:

$$Texto = (chave + cifra) \bmod 26 \tag{1}$$

Decifrador

Para decifrar, é realizado o processo inverso da cifragem: o texto recebido é analisado caractere a caractere juntamente com um caractere de sua chave. Com essas duas informações é realizada uma consulta na *tabula recta*, para encontrar o caractere original daquela posição do texto. É possível escrever a descriptografia como na fórmula a seguir:

$$Texto = ((cifra - chave) + 26) \bmod 26 \tag{2}$$

Voltando ao exemplo anterior, onde "liynvtrut ghqhnxtgahrtp"é o texto cifrado ao final do processo e queremos recuperar o texto original.

Sabe-se que a chave é "teste", dessa forma compara-se o texto cifrado com "testetest etestetestete", que exatamente o corpo do texto coberto com a palavra chave. Dessa forma realiza-se uma consulta na tabela com dois caracteres por vez. A primeira consulta seria percorrer a linha onde está localizada a letra T até encontrar a letra L, a partir da posição de L observar na primeira linha qual é a letra que está na mesma coluna, como exposto na 2.

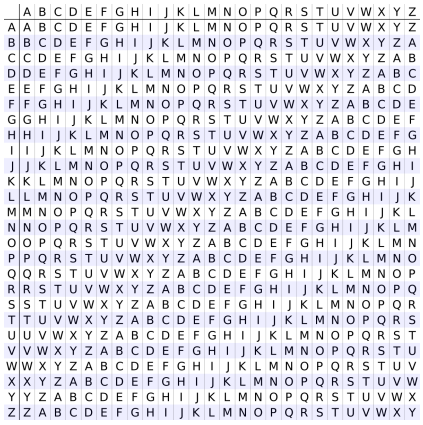

Figura 1: Tabela de Vigenère.

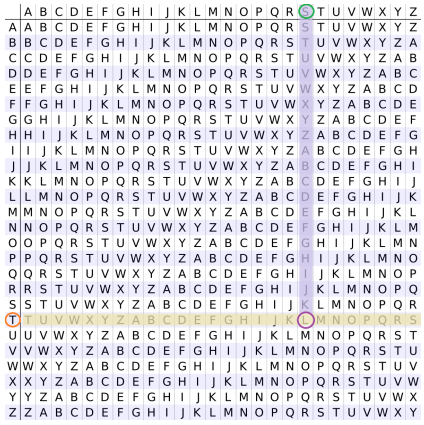

Figura 2: Substituição de caractere.

Quebra Chave

Para encontrar a chave de um texto cifrado foram implementadas cinco funções. Nos procedimentos de descobrir o tamanho de chave e encontrá-la foram utilizadas fórmulas estatísticas para tal, essas serão explicadas ao decorrer dessa seção. O primeiro processo é descobrir o tamanho da chave, para isso o texto cifrado foi dividido em conjuntos de letras separadas por um intervalo, sendo esse intervalo o suposto tamanho da chave. Para descobrir se esse intervalo é o correto, precisamos fazer um cálculo nesse conjunto de letras no intuito de descobrir se esse conjunto de letras é o mesmo caractere. Essa técnica é chamada de Índice de Coincidência [3]. Sua fórmula é a seguinte:

$$IC = \frac{\sum F_i(F_i - 1)}{N(N-1)} \tag{3}$$

Sendo F_i a frequência de uma determinada letra do alfabeto no conjunto escolhido e N o tamanho do conjunto.Após ter calculado em todas as letras do conjunto, é tirado a média do valor em relação ao intervalo. E por fim é feito o mesmo processo em todos os intervalos possíveis. O maior valor encontrado é o provável tamanho da chave.

```
double i_coincidencia(string texto){
    double soma_freq, i_coincidencia;
    double tamanho = texto.length();

    for(int i = 0; i < 26; i++){
        int contador = count(texto.begin(), texto.end(),
    alfabeto[i]); // Formula do indice de coincidencia
        soma_freq += (contador * (contador - 1));
    }

    i_coincidencia = soma_freq/(tamanho*(tamanho-1));

    return i_coincidencia;
}
```

Listing 1.1: Função que calcula o índice de coincidência.

Uma vez descoberto o tamanho da chave, é realizado um procedimento para descobrir os caracteres que a compõem. Considerando que o valor da chave é N, analisamos todas as fatias do texto de tamanho N que foram geradas no processo de descobrir o tamanho. Uma série de deslocamentos é realizado com cada uma das fatias e a cada deslocamento, é verificada a frequência de cada letra do alfabeto na nova fatia. Para descobrir cada caractere da chave, é aplicado o método estatístico Qui-Quadrado [2]:

$$x^2 = \frac{\sum(f_i - F_i)^2}{F_i} \tag{4}$$

Onde f_i é a média da frequência de uma determinada letra na fatia e F_i é a frequência da letra no alfabeto. O programa apresenta um vetor de frequências

das letras em inglês e um vetor para português. Ao final temos o Qui-Quadrado para cada letra, e a letra cujo resultado possui o menor valor será considerada a letra da determinada posição da chave. Esse processo é realizado N vezes.

```
char frequencia(string fatia, int idioma){
    vector<double> X2(26), aux2, lingua;

    ...

    // Quantidade de deslocamento
    for(int i = 0; i < 26; i++){

        (...)

        // Desloca i vezes
        for(int j = 0; j < fatia.size(); j++){
            char seq = (((fatia[j]-97-i+26) % 26)+97);
            sequencia.push_back(seq);
        }

        (...)

        // Formula X2
        for(int j = 0; j < 26; j++){
            double aux1 = (cont_freq[j] - (double)(lingua[j])
   );
            qaux += (aux1*aux1/(double)(lingua[j]));
        }

        X2[i] = qaux;
    }

    (...)
}
```

Listing 1.2: Deslocamentos e cálculo Qui-Quadrado implementados.

Após a descoberta da chave, ela e o texto são levados à função de decifração e o texto original é mostrado na tela.

Referências

1. Slides Criptografia, GTA-UFRJ.
   Disponível em: https://www.gta.ufrj.br/grad/00_2/firewall/criptografia
2. Keyword Recovery with the x2 Method, Dr.C.-K. Shene.
   Disponível em: https://pages.mtu.edu/ shene/NSF-4/Tutorial/VIG/Vig-Recover.html
3. Index of Coincidence, Dr.C.-K. Shene.
   Disponível em: https://pages.mtu.edu/ shene/NSF-4/Tutorial/VIG/Vig-IOC.html