

”

UNIDADE CURRICULAR: Segurança em Redes e Computadores

CÓDIGO: 21181

DOCENTE: Professor Henrique São Mamede

A preencher pelo estudante

NOME: Francisco José Pinto de Amaral

N.º DE ESTUDANTE: 1802876

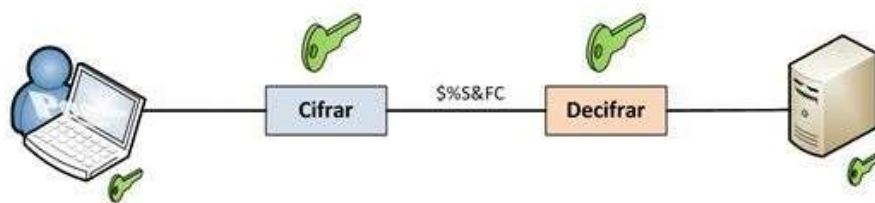
CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 23-11-2021

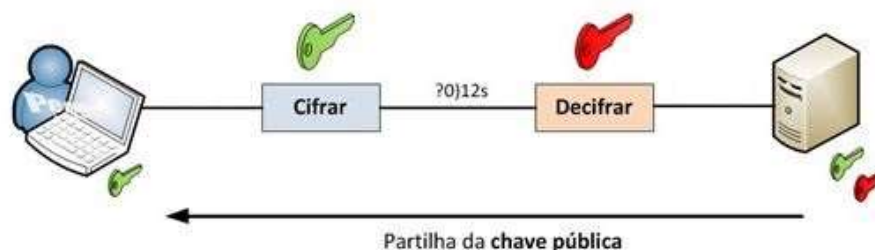
TRABALHO / RESOLUÇÃO:

RELATÓRIO:

Designa-se por cifra simétrica um sistema convencional de chave única (ou chave secreta) usada por ambos interlocutores e na premissa que esta seja conhecida apenas por eles de forma a privacidade e a autenticidade. A desvantagem para este método é a troca da chave entre os dois interlocutores, o processo deve ser feito em segurança de forma que uma terceira pessoa não tenha conhecimento da mesma, toda a segurança é refletida na chave.



Designa-se por cifra assimétrica aquela que utiliza duas chaves: uma chave privada e uma chave pública. Esta é a cifra mais utilizada por ser mais prática, a chave pública é distribuída livremente para todos os correspondentes via email ou outras formas enquanto a chave privada é para uso exclusivo do criador dos pares de chaves para cifrar decifrar mensagens de forma segura. O uso desta cifra surge como uma parte fundamental da segurança da internet (p.e. Blockchain) permitindo que haja sigilo de informação e que a mensagem/conteúdo não foi alterada.



Para implantação das cifras solicitadas neste trabalho escolhi a linguagem de programação **C++** no qual o código foi desenvolvido em um **IDE Web** – www.replit.com que tem como Sistema Operativo base o **ubuntu0.18.04.1 ~ clang version 7.0.0-3**.

Para a **Cifra Simétrica** escolhi a mais simples e mais conhecida dentro das Cifras Polialfabeticas de Substituição – a **Vigenère Cipher** – de acordo com a página 102 do manual adotado - Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. 7th Edition (Global Edition), Prentice Hall.

[illegible]

O Menu inicial tem 3 opções: “[1] Encriptar Mensagem”, “[2] Decifrar Mensagem” e “[3] Sair do Programa”. Pedindo para selecionar uma das três opções.

```
-----
Vigenere Cypher - Menu Principal
-----
[1] Encriptar Mensagem
[2] Decifrar Mensagem
[3] Sair do Programa
-----
Seleccione uma opcao: [ ]
```

O domínio do texto compreende todos os caracteres pela constante “DOM_INF” e “DOM_SUP” que são todos os caracteres visíveis da tabela ASCII Standard do carácter 32 ao 126. Os restantes caracteres da tabela foram ignorados devido a questões de acentuação nas diferentes linguagens/teclado a nível internacional (p.e. ñ).

Na Encriptação cada caracter do texto cifrado é obtido deslocando o respetivo caracter do texto original (*plaintext*) um determinado número de posições ao longo da tabela ASCII. O número de posições a deslocar é determinado a partir do código ASCII de um dos caracteres da chave secreta.

De acordo com a equação para encriptação da página 103 (3.3): $C_i = (p_i + k_i \text{ mod } m) \text{ mod } 26$, o mod 26 são as 26 letras em maiúsculo. Neste caso específico como utilizei mais

do que os 26 caracteres será *mod 94* sendo que vai do código ASCII 32 ao 126 conforme supra explicado.

No caso da chave ser mais curta do que a mensagem (texto original – *plaintext*) o programa repete a chave, no caso da chave ser mais longa que a mensagem ele ignora os caracteres adicionais. ($ki \bmod m$) em que m é o comprimento de caracteres da chave.

Se a chave tiver apenas um caracter, todos os caracteres do texto cifrado irão ser obtidos aplicando o mesmo deslocamento ao texto original, isto é, o algoritmo comporta-se como a cifra de César.

Para decifrar a mensagem o programa comporta-se da mesma forma á exceção do deslocamento ser feito no sentido contrário, em vez da uma soma temos uma subtração conforme as próprias equações demonstram:

$$C_i = (p_i + k_{i \bmod m}) \bmod 26 \quad \text{e} \quad p_i = (C_i - k_{i \bmod m}) \bmod 26.$$

NOTA – Apesar da bibliografia adotada mencionar equações, por exemplo a (3.3) fugi um pouco ao mencionado para mostrar domínio sobre o assunto em questão, usar apenas letras maiúsculas (mod 26) seria uma cópia do que está no manual.

Para a **Cifra Assimétrica** escolhi o Algoritmo RSA criado por Ron Rivest, Adi Shamir e Len Adleman, as iniciais de cada apelido dão nome á cifra em questão – RSA – de acordo com a secção 9.2 (página 294) do manual adotado - Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. 7th Edition (Global Edition), Prentice Hall.

The screenshot shows a C++ IDE with a file explorer on the left containing files like main.cpp, chaves.cpp, chaves.h, geral.cpp, geral.h, mensagem.cpp, and mensagem.h. The main.cpp file is open, showing the implementation of the RSA algorithm. The code includes headers for iostream, math, chaves, geral, and mensagem, and uses the std namespace. The main function calls processMenu(). The processMenu function displays a menu with options: [1] Gerar Chaves, [2] Criptografar Mensagem, [3] Decifrar Mensagem, and [4] Sair do Programa. It then prompts the user to select an option. The console output shows the results of the RSA algorithm, including the generation of public and private keys, and the encryption and decryption of a message.

```

1 #include <iostream>
2 #include <math.h>
3 #include "chaves.h"
4 #include "geral.h"
5 #include "mensagem.h"
6 using namespace std;
7
8 int main() {
9     processMenu();
10
11     Mensagem msg("mensagem");
12     msg.displayNumeros();
13
14     Mensagem msgEnc(msg.getNumeros());
15     cout<<"Mensagem codificada:"<<msgEnc.getTexto()<<endl;
16
17     Chave ch(0,0);
18
19     double e=chave.getE();
20     double n=chave.getN();
21     double phi=chave.getPhi();
22     double d=chave.getD();
23
24     vector<double> txtCifrado=msg.criptografa(chave.getE(),chave.getN());
25
26     cout<<"Texto cifrado:"<<endl;
27     for(auto it:txtCifrado){
28         cout<<it<<" ";
29     }
30     cout<<endl;
31
32     vector<double> txtDecifrado;
33     for(auto it:txtCifrado){
34         double m= pow(it,d);
35         m=fmod(m,n);
36         txtDecifrado.push_back(m);
37         cout<<"m:"<<m<<endl;
38     }
39     cout<<endl;
40
41     Mensagem msgDecifrado(txtDecifrado);
42     cout<<"Mensagem Decifrada:"<<msgDecifrado.getTexto()<<endl;
43
44     cout<<"Texto decifrado:"<<endl;
45     for(auto it:txtDecifrado){
46         cout<<it<<" ";
47     }
48     cout<<endl;
49 }

```

Console Output:

```

> clang++ 7 -pthread -std=c++17 -o main chaves.cpp geral.cpp main.cpp mensagem.cpp 0 x
> ./main

Algoritmo RSA - Menu Principal
[1] Gerar Chaves
[2] Criptografar Mensagem
[3] Decifrar Mensagem
[4] Sair do Programa

Selecione uma opção: 1

Algoritmo RSA - Geração de Chaves:

Chave Publica:      e: 7
                   n: 4.78842e+07

Chave Privada:      d: 8.142857

[Enter] para continuar...

Algoritmo RSA - Menu Principal
[1] Gerar Chaves
[2] Criptografar Mensagem
[3] Decifrar Mensagem
[4] Sair do Programa

Selecione uma opção: 2

Algoritmo RSA - Criptografia da Mensagem Original:

Texto Original:
mensagem

Mensagem Cifrada:
3.30642e+41 7.34690e+40 2.09715e+13

[Enter] para continuar...

```

O Menu inicial, neste caso de cifra assimétrica tem 4 opção tendo em conta que existe a necessidade de gerar as chaves Pública e Privada, sendo: “[1] Gerar Chaves”, “[2] Encriptar Mensagem”, “[3] Decifrar Mensagem” e “[4] Sair do Programa”. Pedindo para seleccionar uma das quatro opções.

```
-----
Algoritmo RSA - Menu Principal
-----
[1] Gerar Chaves
[2] Encriptar Mensagem
[3] Decifrar Mensagem
[4] Sair do Programa
-----
Selecione uma opcao: █
```

A organização que vai receber a(s) mensagem(s) é que vai fornecer as chaves para encriptar a mensagem. Por exemplo: <https://www.cnccs.gov.pt/pt/chave-gpg/>

Escolhem-se os números primos p e q , quanto maior for o número mais difícil será de fatorizar fazendo com que haja uma menor probabilidade de sofrer um ataque por força bruta. Ambos os valores de p e q multiplicados obtem-se o valor de n que é um valor que faz parte das duas chaves.

De seguida calculamos o totiente de pq (que lhe dou o nome de phi no programa) através da formula $\phi(pq) = (p-1)(q-1)$ de acordo com o manual adotado.

De formas prática, para preparar a mensagem para ser cifrada existe um trabalho de preparação que passa por transformar a mensagem em números. Esta transformação passa por obter o código ASCII de cara character da mensagem a encriptar e subtrair-lhe 32, após termos a nossa mensagem transformada em números dividimos a cadeia de caracteres em blocos de 3 considerando da direita para a esquerda o character mais á direita é multiplicado por 10^0 , o do meio multiplicado por 10^2 e o da esquerda multiplicado por 10^4 , somando os três múltiplos dá origem a um número de 6 números prontos a ser cifrados, aplicando de seguida o algoritmo.

Deste modo, na decifragem conseguimos aplicar o mesmo método no sentido inverso sem perder o caminho até á mensagem original.

REFERÊNCIAS BIBLIOGRÁFICAS:

Stallings, William (2007) *Cryptography and Network Security: Principles and Practice*. 7th Edition (Global Edition), Prentice Hall.

Imagens - <https://pplware.sapo.pt/tutoriais/networking/criptografia-simetrica-e-assimetrica-sabe-a-diferenca/>

Conteúdo fórum: Modulo 2 – Encriptação de Dados e Infraestrutura de Chave Pública, Curso Especialização em Cibersegurança – Edição 2019/2020, Universidade Aberta.

FIM