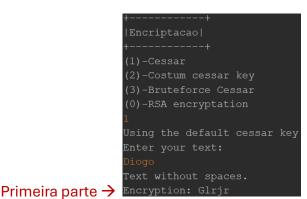
Introdução

A encriptação é um processo fundamental para garantir a segurança e a privacidade das comunicações. O projeto foca-se no ambiente informático. Este relatório tem o intuito de demonstrar o funcionamento de quatro opções de encriptação presentes no código, desde métodos simples como a Cifra de César até algoritmos complexos como o RSA. Cada opção será apresentada num output , destacando suas funcionalidades, vantagens e desvantagens.

Notas importantes \rightarrow As mensagens para o algoritmo RSA estão pre-definidas no código fonte isto é caso avance de imediato para a opção 0 . Somente se for submetido a opção 1 ou 2 em termos de encriptação e desencriptação, essa mesma mensagem será usada pela pessoa A.

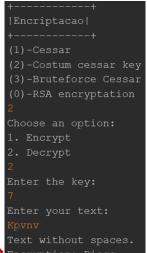
Exemplo: Caso use a primera opção, e depois a opção 0, será transferida a mensagem encriptada como exemplo para o algoritmo RSA. Forma correta de executar o programa será:

Usar a opção 1 ; Usar a opção 2 (dentro dessa opção utilizar primeiro a encriptação em modo costum e depois desincriptar); Opçao 3 é opcional. Por ultimo utilizar a opção 0 . Abaixo estarão imagens de forma a utilizar . A laranja são os inputs colocados pelo utilizador.



Segunda parte

Text without spaces. (Key/deslocamento aconselhável usar entre 1 a 10 por causa dos caracteres não identificados que aparecem na consola, mas regra geral);



Terceira parte Diogo (Para o bom funcionamento da desencriptação será necessária a chave usada na encriptação, neste exemplo a chave é 7) (Será também necessário colocar como input a mensagem encriptada recebida no passo 2)

Bruteforce \rightarrow So poderá ser utilizada o bruteforce caso a mensagem atual ainda esteja de forma encriptada . Cada coluna vertical é um nível da chave (primeira coloca = deslocamento 0 , etc) ;

Quarta parte >

Opção 1: Cifra de César com Deslocamento Fixo

A primeira opção utiliza a Cifra de César, um método de cifra que consiste em deslocar cada letra do alfabeto por um número fixo de posições . Para o caso geral , a deslocação será de 3.

Funcionamento: Apos inserção do texto a ser encriptado, cada letra desse texto será transformada na 3 letra seguinte (na ordem alfabética).

Exemplos:

Texto original: "ABC"

Texto encriptado: "DEF"

Opção 2: Cifra de César com Deslocamento Personalizado

A segunda opção permite ao utilizador escolher o deslocamento da Cifra de César, na qual irá ser apresentada duas opções : para encriptar e desencriptar mensagens.

| Encriptação: O usuário terá que inserir um número entre 1 e 26 (a chave) e o texto a ser cifrado. |
|--|
| Exemplo: |
| Chave: 5 |
| Texto original: "Diogo" |
| Texto encriptado: "Intlt" |
| Desencriptação: O usuário insere a chave correta e o texto encriptado para obter o texto original. |
| Exemplo: |
| Chave: 5 |
| Texto encriptado: "Intlt" |
| Texto original: "Diogo" |
| |
| Opção 3: Ataque de Força Bruta na Cifra de César |
| A terceira opção realiza um ataque de força bruta (bruteforce em inglês) na Cifra |

A terceira opção realiza um ataque de força bruta (bruteforce em inglês) na Cifra de César (1º opção) . Neste caso irá testar todas as possibilidades de deslocamentos do alfabeto.

Funcionamento: O bruteforce testa todos os deslocamentos de 1 a 26 (de A a Z) e apresenta todas as variantes da mensagem encriptada.

```
Glrjr | Fkqiq | Ejphp | Diogo | Chnfn | Egmem | Afld| | Sekok | 2djb| > ciai | =bh'h | Cag | 1*frf | Egiph | Diogo | Chnfn | Egmem | Afld| | Sekok | 2djb| > ciai | =bh'h | Cag | 1*frf | Egiph | Egiph | Diogo | Chnfn | Egmem | Afld| | Sekok | 2djb| > ciai | =bh'h | Cag | 1*frf | Egiph | Egiph | Diogo | Chnfn | Egmem | Afld| | Sekok | 2djb| > ciai | =bh'h | Cag | 1*frf | Egiph | Eg
```

Objetivo desta opção : Demonstrar a vulnerabilidade da Cifra de César, destacando que qualquer pessoa pode quebrar a encriptação ao tentar todas as chaves possíveis.

Conclusão da primeira parte \rightarrow Estas primeiras 3 opções serviu para mostrar como funciona uma encriptação simétrica .

Vantagens : Rápida e fácil de configurar ;

Desvantagens: A chave secreta que for comprometida, a comunicação também fica comprometida. E normalmente para haver comunicação entre duas pessoas, ambas terão que "combinar" a chave secreta. Em ambiente tecnológico, se essa mesma chave for compartilhada com o destinatário, existe um risco que a chave seja roubada se a rede não for segura.

Opção 0: Algoritmo RSA

A quarta opção utiliza o algoritmo RSA, um método de encriptação assimétrica que gera quatro chaves: duas públicas e duas privadas. Esta opção será apresentada em ultima e utilizará o texto que foi colocado no inicio da

apresentação do trabalho . (Será necessário começar utilizar a cifra de cessar para depois demonstrar esta opção)

Criação de Chaves: Cada pessoa recebe um par de chaves (uma pública e uma privada) que são matematicamente relacionadas. Neste caso utilizei 2048 bits para criação das chaves.

Comunicação:

Pessoa A encripta a mensagem usando a chave pública de Pessoa B.

Pessoa B desencripta a mensagem usando sua própria chave privada.

O processo é reversível, permitindo que Pessoa B também envie mensagens seguras para Pessoa A.

Segurança: A força do RSA reside na dificuldade de fatorar grandes números primos, tornando-o muito mais seguro que métodos de cifra de substituição simples, como a cifra de cesar . Esta encriptação é assimétrica

Exemplo:

Pessoa A deseja enviar uma mensagem segura para Pessoa B.

Pessoa A usa a chave pública de Pessoa B para encriptar a mensagem.

Pessoa B usa sua chave privada para desencriptar a mensagem.

Conclusão da ultima parte \rightarrow Demonstração de uma encriptação assimétrica .

Vantagens : Permite a autentificação e verificação da origem de uma certa mensagem ;

As chaves publicas normalmente são distribuídas em servidores com certificados digitais e seguros , ou seja não existe mal em a chave publica se tornar conhecida .

As chaves publicas são geradas a partir da chave privada por métodos matemáticos complexos tornando assim difícil achar qual é a chave privada (a não ser que seja divulgada) .

Desvantagens : A encriptação assimétrica é mais lenta que a simétrica . Se a chave privada se perder , ninguém poderá descriptografar a informação

Conclusão

Este trabalho demonstra a utilização das quatro opções apresentadas , desde a encriptação simétrica da Cifra de César e da complexidade do algoritmo RSA (encriptação assimétrica) .

Cada encriptação tem suas próprias funcionalidades e níveis de segurança, sendo essencial escolher a melhor encriptação apropriada conforme o nível de segurança.

Referências:

Cifra de César: Simplicidade e vulnerabilidades

RSA: Complexidade e segurança matemática

Fontes:

Algoritmo RSA → Blackbox.ai