Criptografia com Shift em C

Thiago Rafael Kraus & Arthur Koji Maeda

Setembro 2025

1 Introdução

Neste material vamos estudar um algoritmo de criptografia baseado em **shift**. O objetivo é compreender esse algoritmo em C, utilizando ponteiros, funções, structs.....

2 Como funciona a criptografia com Shift

A ideia é simples: transformar cada caractere em seu código numérico (ASCII), somar ou subtrair um valor de *shift* e depois aplicar módulo 256.

$$\begin{aligned} \text{Criptografar:} \quad & C_{\text{cripto}} = (C_{\text{original}} + shift) \bmod 256 \\ \text{Descriptografar:} \quad & C_{\text{original}} = (C_{\text{cripto}} - shift + 256) \bmod 256 \end{aligned}$$

O `shift` é calculado somando os valores ASCII da chave e aplicando mod256.

3 Exemplo prático

Suponha:

- Mensagem: `"OLA"`
- Chave: `"abc"`

Cálculo do shift:

$$\texttt{'a'} = 97,\ \texttt{'b'} = 98,\ \texttt{'c'} = 99 \quad \Rightarrow \quad 97 + 98 + 99 = 294 \quad \Rightarrow \quad 294 \bmod 256 = 38$$

Criptografando "OLA"

Caractere	ASCII	Operação	Novo Código	Resultado
O	79	(79 + 38) % 256 = 117	117	u
L	76	(76 + 38) % 256 = 114	114	r
A	65	(65 + 38) % 256 = 103	103	g

Mensagem criptografada: `"urg"`.

Descriptografando "urg"

Caractere	ASCII	Operação	Novo Código	Resultado
u	117	(117 - 38 + 256) % 256 = 79	79	O
r	114	(114 - 38 + 256) % 256 = 76	76	L
g	103	(103 - 38 + 256) % 256 = 65	65	A

Mensagem descriptografada: `"OLA"`.

4 Explicação do Código

Bibliotecas

- `stdio.h` — entrada e saída (printf, fgets).
- `string.h` — manipulação de strings (strlen, strncmp, strcspn).
- `stdlib.h` — alocação dinâmica (malloc, free, exit).

Estruturas de dados

- `struct Senha` — armazena senhas possíveis em um vetor.
- `char[]` — arrays de caracteres para mensagens e senhas.
- `char*` — ponteiros para memória alocada dinamicamente.

Funções criadas

- `calcularShift` — calcula o deslocamento (shift) da chave.
- `decrypt` — descriptografa a mensagem.
- `main` — executa o programa e testa todas as senhas.

5 Conclusão

Esse algoritmo de criptografia com shift é simples, mas eficiente para fins didáticos. Ele permite mostrar na prática como trabalhar com ponteiros, funções, vetores, structs e manipulação de strings em C, além de introduzir conceitos básicos de criptografia.

6 O Código em C

A seguir, o código completo em C que tenta descriptografar mensagens usando várias senhas possíveis.

```c
// Bibliotecas utilizadas
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Estrutura para armazenar uma senha
typedef struct {
    char senha[20];
} Senha;

// Fun  o para calcular o shift a partir da chave
int calcularShift(const char *key) {
    int shift = 0;
    while (*key) {
        shift += (unsigned char)(*key);
        key++;
    }
    return shift % 256;
}

// Fun  o para descriptografar uma mensagem
char* decrypt(const char *encrypted, const char *key) {
    int shift = calcularShift(key);
    int length = strlen(encrypted);

    char *decrypted = (char*) malloc((length + 1) * sizeof(char))
       ;
    if (!decrypted) {
        printf("Erro de alocacao de memoria!\n");
        exit(1);
    }

    for (int i = 0; i < length; i++) {
        decrypted[i] = (char)(((unsigned char)encrypted[i] -
           shift + 256) % 256);
    }

    decrypted[length] = '\0';
    return decrypted;
}

int main() {
    // Vetor de senhas poss veis
    Senha senhas[] = {
        {"alpha1236r"},
        {"beta2421fd"},
        {"gamma3bdfb"},
```

```
        {"delta4fdbs"},
        {"epsilon5mm"},
        {"zeta6khfvd"},
        {"eta7cadw1v"},
        {"theta8vtqv"},
        {"iota943f3f"},
        {"kappa10ebb"},
        {"lambda11aa"},
        {"mu12v3333f"}
    };

    int total_senhas = sizeof(senhas) / sizeof(senhas[0]);

    char mensagem_criptografada[512];
    printf("Digite a mensagem criptografada: ");
    fgets(mensagem_criptografada, sizeof(mensagem_criptografada),
        stdin);
    mensagem_criptografada[strcspn(mensagem_criptografada, "\n")]
        = 0;

    const char *prefixo = "DEPARTAMENTO DE SEGURANCA DOS ESTADOS
       UNIDOS";

    for (int i = 0; i < total_senhas; i++) {
        char *tentativa = decrypt(mensagem_criptografada, senhas[
           i].senha);

        if (strncmp(tentativa, prefixo, strlen(prefixo)) == 0) {
            printf("\nSenha correta encontrada: %s\n", senhas[i].
               senha);
            printf("Mensagem descriptografada:\n%s\n", tentativa)
               ;
            free(tentativa);
            return 0;
        }
        free(tentativa);
    }

    printf("Nenhuma senha funcionou!\n");
    return 0;
}
```