Cifragem x Codificação

Mário Leite

...

Desde há muito tempo o homem se preocupa com a proteção de suas mensagens enviadas; e a proteção varia desde a ocultação da mensagem até sua modificação textual de modo que somente o destinatário possa descobrir o seu conteúdo. Ao digitar sua *senha* num sistema aparece asteriscos no lugar dos caracteres digitados: é uma maneira de proteger essa *senha*. Um tipo de proteção de texto conhecido como **esteganografia** (derivada do grego; *steganos*=encoberto + *graphein*=escrever) foi muito usado durante as guerras entre a Grécia e o Império Persa. O método era o de *raspar a cabeça de um mensageiro e escrever a mensagem em sua cabeça*; depois deixar seu cabelo crescer e enviá-lo ao destinatário. A mensagem ficava oculta na cabeça dele e só poderia ser acessada pelo destinatário que tinha a "chave" para descobrir o seu conteúdo: *raspar a cabeça do mensageiro*. Embora este tipo de proteção pudesse funcionar na maioria das vezes, em algum momento o inimigo poderia desconfiar e "dar uma geral" no emissário, o que poderia incluir a raspagem de seu cabelo; e neste caso a mensagem seria revelada e com graves consequências para o mensageiro! Mais recentemente, durante a Segunda Guerra Mundial, os alemães utilizaram este método de proteção das suas mensagens com o famoso "microponto", que consistia em fotografar a mensagem e reduzi-la a um ponto no papel; e normalmente esse "microponto" era colocado ao final de um texto qualquer, sem importância, para não chamar a atenção caso o papel fosse interceptado pelo inimigo. Somente o destinatário sabia a "chave" para encontrar esse "microponto" e como aumentá-lo para, posteriormente, ler a mensagem. Um segundo tipo de proteção é a **criptografia**, que altera o texto da mensagem original através de substituição e/ou transposição de caracteres. O termo "criptografia" é originário do grego (*kryptós*=escondido + *gráphen*=escrita) e é definido pela enciclopédia Wikipédia como o "*estudo dos princípios e técnicas pelas quais a* informação *pode ser transformada da sua forma original para outra ilegível, de forma que possa ser conhecida apenas por seu destinatário*". Na prática o que se deseja é transformar um texto legível em algo ilegível. O objetivo final da criptografia é dar segurança a um texto confidencial, tornando a informação nele contida inacessível para aquelas pessoas não autorizadas.

De um modo geral, "cifrar" significa substituir os caracteres do texto original por outros, isto é, misturar uma mensagem usando uma cifra seguindo alguma lógica. Já o termo "codificar" é ocultar o significada da mensagem original usando algum código: e é aí que entra a **criptografia**. Deste modo, o verbo decifrar significa "revelar uma mensagem que foi cifrada", e decodificar é "revelar uma mensagem que foi codificada". Portanto, os termos "encriptar" e "decriptar" (criptografar e descriptografar) são mais gerais e podem ser empregados tanto para cifrar/codificar como para decifrar/decodificar.

O programa "**CifragemAtbash**" faz uma cifragem/decifragem de um texto usando o método conhec ido como "Cifragem Atbash", monoalfabético, onde cada letra do alfabeto é substituída pela letra correspondente na outra extremidade do alfabeto. Já, programa "**CriptoECC**" é mais sofisticado, pois faz a encriptação/decriptação (criptografa/descriptografa) um texto com o "Método de Curva Elíptica", usando chaves temporárias em protocolo comum para obter maior segurança e evitar problemas associados à reutilização de chaves, e a saída (texto codificado) é muito mais difícil de ser decodificado, pois são caracteres não legíveis

---

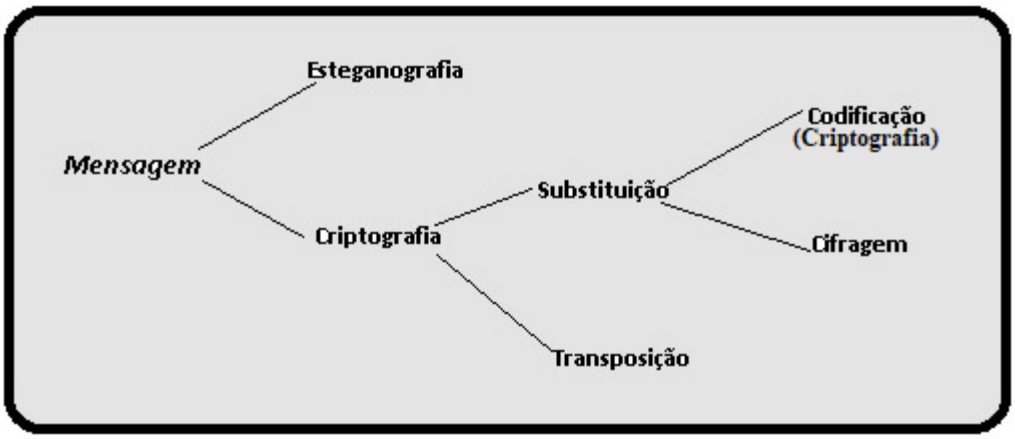

Figura 1 - Esquema de proteção de uma mensagem

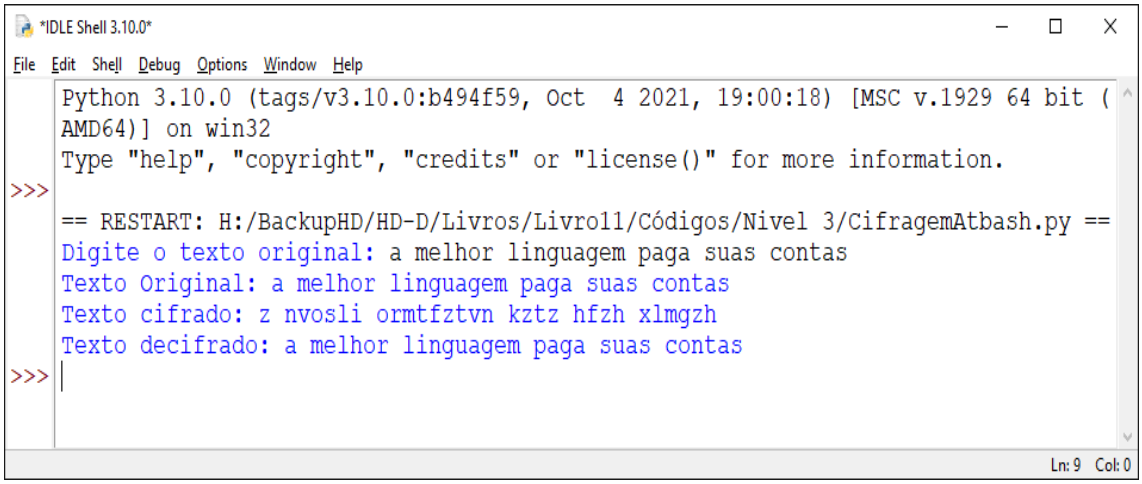

Figura 2 - Exemplo de saída do programa "CifragemAtbash"

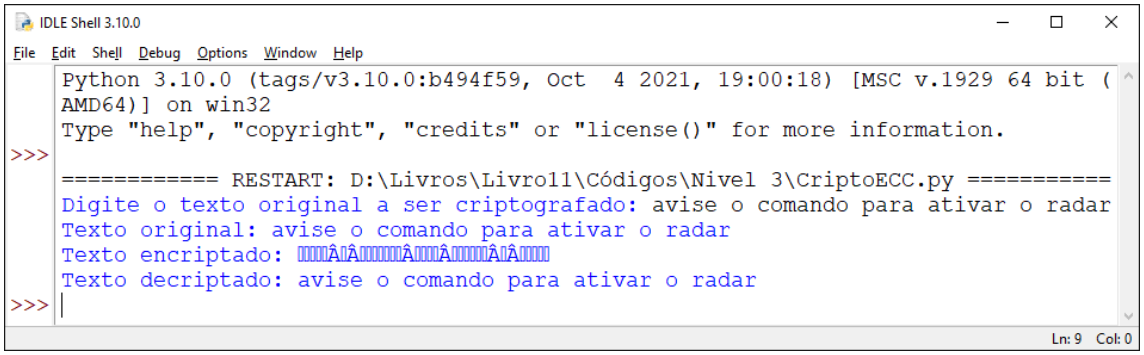

Figura 3 - Exemplo de saída do programa "CriptoECC"

```
'''
CifragemAtbash.py
-------------------------------------------------------------------------
Faz a cifragem/decifragem de um texto usando o "Cifragem Atbash" usando
inversão de letras da mensagem
-------------------------------------------------------------------------
'''
def Cifrar(text):
    result = ""
    for char in text:
        if(char.isalpha()):
            if(char.isupper()):
                result += chr(ord('Z') - (ord(char) - ord('A')))
            else:
                result += chr(ord('z') - (ord(char) - ord('a')))
        else:
            result += char
    return result
#=========================================================================
#Programa principal
textOriginal = ""
while(len(textOriginal) < 2):
    textOriginal = input("Digite o texto original: ")
textCifrado = Cifrar(textOriginal) #chama a função Cifrar para a cifragem
textDecifrado = Cifrar(textCifrado) #chama a função Cifrar para a decifragem
#Exibe os resultados
print(f"Texto Original: {textOriginal}")
print(f"Texto cifrado: {textCifrado}")
print(f"Texto decifrado: {textDecifrado}")
#Fim do programa "CifragemAtbash" ------------------------------------
```

```
'''
CriptoECC.py
-------------------------------------------------------------------------------
Faz a criptografia de chave dupla de um texto com o "Método de Curva
Elíptica", usando chaves temporárias comuns em protocolos.
-------------------------------------------------------------------------------
'''
import hashlib
import os

def GerarParChaves():
    ChavePrivada = int.from_bytes(os.urandom(32), byteorder='big')
    ChavePublica = (ChavePrivada*0x5D576E7357A4501DDE4EE2311B9BCF6C,
                    ChavePrivada*0x6AEBCA40BA255960A3178D6D861A54DB)
    return ChavePrivada, ChavePublica
#-------------------------------------------------------------------------------
def Encriptar(ChavePublica, plaintext):
    ChavePrivadaTemp = int.from_bytes(os.urandom(32),byteorder='big')
    ChavePublicaTemp=(ChavePrivadaTemp*0x5D576E7357A4501DDE4EE2311B9BCF6C
                      ChavePrivadaTemp*0x6AEBCA40BA255960A3178D6D861A54DB)

    ChaveSecretaX = ChavePrivadaTemp * ChavePublica[0]
    ChaveSecretaY = ChavePrivadaTemp * ChavePublica[1]
    ChaveSecreta = int(hashlib.sha256(str(ChaveSecretaX +
                   ChaveSecretaY).encode()).hexdigest(),16)%(1 << 256)
   textoEncriptado=''.join([chr((ord(char)+ChaveSecreta)%256) for char in plaintext])
```

```
    return ChavePublicaTemp, textoEncriptado
#-------------------------------------------------------------------------------
def Decriptar(ChavePrivada, ChavePublicaTemp, textoEncriptado):
    ChaveSecretaX = ChavePrivada * ChavePublicaTemp[0]
    ChaveSecretaY = ChavePrivada * ChavePublicaTemp[1]
    ChaveSecreta = int(hashlib.sha256(str(ChaveSecretaX +
                   ChaveSecretaY).encode()).hexdigest(),16)%(1 << 256)
    textoDecriptado = ''.join([chr((ord(char) - ChaveSecreta) % 256)
                      for char in textoEncriptado])
    return textoDecriptado
#===============================================================================
#Programa principal
ChavePrivada, ChavePublica = GerarParChaves()
textOriginal = ""
textOriginal = ""
while(len(textOriginal) < 2):
    textOriginal=input("Digite o texto original a ser criptografado: ")
ChavePublicaTemp, textoEncriptado = Encriptar(ChavePublica, textOriginal)
textoDecriptado = Decriptar(ChavePrivada, ChavePublicaTemp, textoEncriptado)
print("Texto original:", textOriginal)
print("Texto encriptado:", textoEncriptado)
print("Texto decriptado:", textoDecriptado)
#Fim do programa "CriptoECC" ----------------------------------------------------
```