

Algoritmo 1: Cifra de César

Descrição:

O Algoritmo 1 é baseado na Cifra de César, uma técnica de criptografia de substituição simples que desloca cada letra do texto original por um número fixo de posições no alfabeto. Este é um exemplo básico e inseguro de criptografia, usado apenas para fins educacionais.

Implementação:

```

import tkinter as tk

# Função para criptografar usando a Cifra de César
def cifra_de_cesar(texto, chave):
    texto_cifrado = ""
    for char in texto:
        if char.isalpha():
            shift = ord('a') if char.islower() else ord('A')
            texto_cifrado += chr((ord(char) - shift + chave) % 26 + shift)
        else:
            texto_cifrado += char
    return texto_cifrado

# Função para cifrar o texto quando o botão "Cifrar" é pressionado
def cifrar_texto():
    texto_original = entrada_texto.get()
    chave = int(entrada_chave.get())
    texto_cifrado = cifra_de_cesar(texto_original, chave)
    label_resultado["text"] = "Texto Cifrado: " + texto_cifrado

# Configuração da interface gráfica
janela = tk.Tk()
janela.title("Cifra de César")

entrada_label = tk.Label(janela, text="Texto Original:")
entrada_label.pack()

entrada_texto = tk.Entry(janela)
entrada_texto.pack()

chave_label = tk.Label(janela, text="Chave (deslocamento):")
chave_label.pack()

entrada_chave = tk.Entry(janela)
entrada_chave.pack()

botao_cifrar = tk.Button(janela, text="Cifrar", command=cifrar_texto)
botao_cifrar.pack()

label_resultado = tk.Label(janela, text="")
label_resultado.pack()

janela.mainloop()

```

Explicação:

- `cifra_de_cesar`: Esta função recebe um texto e uma chave como entrada e retorna o texto cifrado usando a Cifra de César. A cifra funciona deslocando cada letra do texto pelo valor da chave. Se o caractere não for uma letra, ele é deixado inalterado.
- A interface gráfica é criada usando a biblioteca `tkinter`. Ela inclui campos de entrada para o texto original e a chave, um botão para cifrar o texto e um rótulo para exibir o resultado.

Execução:

1. Abra o prompt de comando e navegue até a pasta onde você salvou o arquivo `cifra_de_cesar.py`.
2. Execute o script com o comando:

```
python cifra_de_cesar.py
```

3. A interface gráfica será aberta. Você verá os seguintes campos:
 - "Texto Original": Digite o texto que deseja cifrar. Por exemplo, digite "HELLO" (sem as aspas).
 - "Chave (deslocamento)": Insira um número inteiro que servirá como chave de deslocamento. Por exemplo, insira "3".
4. Clique no botão "Cifrar" para executar o algoritmo.
5. O campo "Texto Cifrado" mostrará o resultado da cifração. No exemplo acima, o resultado seria "KHOOR" como saída, pois cada letra foi deslocada 3 posições no alfabeto.

Algoritmo 2: AES (Advanced Encryption Standard)

Descrição:

O Algoritmo 2 utiliza o Advanced Encryption Standard (AES) para criptografar dados de forma segura. O AES é um algoritmo de criptografia simétrica amplamente adotado e considerado seguro para proteção de dados confidenciais.

Implementação:

```

import tkinter as tk
from cryptography.fernet import Fernet

# Chave Fernet válida
chave = b'MWVqsaTYKcg3sWVQZ22NE8KUxW_6SpOybkN-CRFdaSo='

# Função para cifrar usando AES
def cifrar_aes(texto, chave):
    fernet = Fernet(chave)
    texto_bytes = texto.encode()
    texto_cifrado = fernet.encrypt(texto_bytes)
    return texto_cifrado

# Função para cifrar o texto quando o botão "Cifrar" é pressionado
def cifrar_texto_aes():
    texto_original = entrada_texto_aes.get()
    texto_cifrado = cifrar_aes(texto_original, chave)
    texto_cifrado_str = texto_cifrado.decode()
    texto_cifrado_entry.delete(0, tk.END)
    texto_cifrado_entry.insert(0, texto_cifrado_str)

# Configuração da interface gráfica para AES
janela_aes = tk.Tk()
janela_aes.title("AES (Advanced Encryption Standard)")

entrada_label_aes = tk.Label(janela_aes, text="Texto Original:")
entrada_label_aes.pack()

entrada_texto_aes = tk.Entry(janela_aes)
entrada_texto_aes.pack()

botao_cifrar_aes = tk.Button(janela_aes, text="Cifrar", command=cifrar_texto_aes)
botao_cifrar_aes.pack()

texto_cifrado_entry = tk.Entry(janela_aes)
texto_cifrado_entry.pack()

label_resultado_aes = tk.Label(janela_aes, text="")
label_resultado_aes.pack()

janela_aes.mainloop()

```

Explicação:

- `gerar_chave_aes`: Esta função gera uma chave AES usando o módulo Fernet da biblioteca `cryptography`. A chave é uma sequência aleatória que será usada para criptografar e descriptografar o texto.

- `cifrar_aes`: Esta função recebe um texto e uma chave AES como entrada e retorna o texto cifrado usando o AES. A biblioteca Fernet facilita a cifragem e decifragem dos dados.
- A interface gráfica para o AES é semelhante à do Algoritmo 1, mas inclui a geração de uma chave AES e usa essa chave para cifrar o texto.

Execução:

Certifique-se de ter a biblioteca `cryptography` instalada. Se você ainda não instalou, você pode instalar via `pip`:

Abra o terminal e execute o comando:

```
pip install cryptography
```

Isso instalará a biblioteca `cryptography` necessária para o Algoritmo 2.

Execução do Algoritmo 2: AES (Advanced Encryption Standard)

1. Abra o terminal ou prompt de comando.
2. Navegue até a pasta onde você salvou o arquivo `aes.py` utilizando o comando `cd`.
3. Execute o script com o seguinte comando:

```
python aes.py
```

Isso iniciará a interface gráfica do algoritmo que utiliza o AES para criptografar texto.

Utilizando a interface gráfica:

Ao executar o script `aes.py`, uma interface gráfica será aberta.

- Você verá campos para inserir o texto que deseja cifrar e a chave de criptografia.
- Insira o texto e a chave nos campos correspondentes.
- Em seguida, clique no botão apropriado para realizar a criptografia.
- A interface exibirá o texto cifrado resultante na própria interface.

O AES é um algoritmo de criptografia simétrica, amplamente utilizado para criptografar e descriptografar dados. A interface gráfica permitirá que você experimente a funcionalidade de criptografia oferecida por esse algoritmo.

Decrypt_AES.py

Descrição:

O arquivo `decrypt_aes.py` é parte do Algoritmo 2 (AES - Advanced Encryption Standard) e é responsável por descriptografar um texto cifrado usando a mesma chave que foi usada para criptografá-lo. Este arquivo permite reverter o processo de criptografia, tornando o texto original novamente legível.

Explicação:

- `descriptografar_aes`: Esta função recebe um texto cifrado e a chave AES usada para criptografá-lo. Ela utiliza o algoritmo AES para descriptografar o texto e retorna o texto original em formato legível.
- A interface gráfica é semelhante à do Algoritmo 2, mas inclui a opção de descriptografar o texto. Ela permite que você insira o texto cifrado e a chave AES e, em seguida, clique no botão apropriado para realizar a descriptografia.

Implementação:

```
import tkinter as tk
from cryptography.fernet import Fernet

# Chave Fernet válida (cole a chave que você gerou anteriormente aqui)
chave = b'MWVqsaTYKcG3sWVQZ22NE8KUxW_6SpOybkN-CRFdaSo='

# Função para decifrar usando AES
def decifrar_aes(texto_cifrado):
    fernet = Fernet(chave)
    texto_decifrado_bytes = fernet.decrypt(texto_cifrado)
    texto_decifrado = texto_decifrado_bytes.decode()
    return texto_decifrado

# Função para decifrar o texto quando o botão "Decifrar" é pressionado
def decifrar_texto_aes():
    texto_cifrado = entrada_texto_cifrado_aes.get()
    texto_decifrado = decifrar_aes(texto_cifrado)
    label_resultado_decifrado_aes["text"] = "Texto Decifrado: " + texto_decifrado

# Configuração da interface gráfica para a descriptografia do AES
janela_decifrar_aes = tk.Tk()
janela_decifrar_aes.title("AES Decifrar")

entrada_label_cifrado_aes = tk.Label(janela_decifrar_aes, text="Texto Cifrado:")
entrada_label_cifrado_aes.pack()

entrada_texto_cifrado_aes = tk.Entry(janela_decifrar_aes)
entrada_texto_cifrado_aes.pack()

botao_decifrar_aes = tk.Button(janela_decifrar_aes, text="Decifrar", command=decifrar_texto_aes)
botao_decifrar_aes.pack()

label_resultado_decifrado_aes = tk.Label(janela_decifrar_aes, text="")
label_resultado_decifrado_aes.pack()

janela_decifrar_aes.mainloop()
```

Execução:

Certifique-se de ter a biblioteca `cryptography` instalada. Se você ainda não instalou, você pode instalar via `pip`:

Abra o terminal e execute o comando:

```
pip install cryptography
```

Isso instalará a biblioteca `cryptography` necessária para o `decrypt_aes.py`.

Execução do Decrypt_AES.py

1. Abra o terminal ou prompt de comando.
2. Navegue até a pasta onde você salvou o arquivo `decrypt_aes.py` utilizando o comando `cd`.
3. Execute o script com o seguinte comando:

```
python decrypt_aes.py
```

Isso iniciará a interface gráfica do arquivo `decrypt_aes.py`, que permite a descriptografia do texto.

Utilizando a interface gráfica:

Ao executar o script `decrypt_aes.py`, uma interface gráfica será aberta.

- Você verá campos para inserir o texto cifrado que deseja descriptografar e a chave de descriptografia (a mesma chave que foi usada para criptografar o texto).
- Insira o texto cifrado e a chave nos campos correspondentes.
- Em seguida, clique no botão apropriado para realizar a descriptografia.
- A interface exibirá o texto original resultante na própria interface.

O `decrypt_aes.py` é uma parte importante do Algoritmo 2 que permite descriptografar dados criptografados anteriormente usando a mesma chave AES.