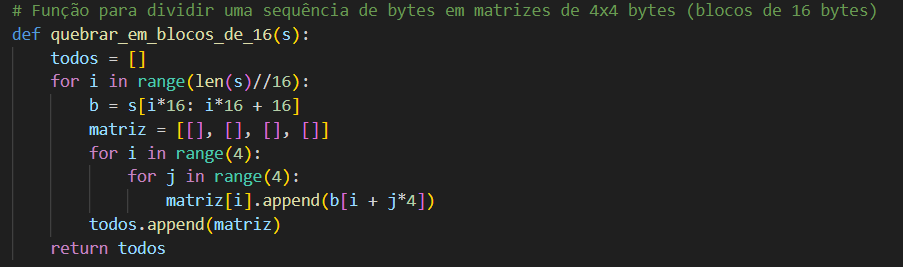
Trabalho de Segurança Computacional AES – ECB/CTR

Douglas Vieira Alcantara – 18/0031333

# AES ECB

Foi pedido a criação de um programa python, que realize a cifração e decifração utilizando o método AES nos modos ECB

Para a criação do programa, foram criados os seguintes elementos:

* quebrar\_em\_blocos\_de\_16: Esta função recebe uma sequência de bytes e divide-a em matrizes de 4x4 bytes, representando blocos de 16 bytes cada.
  + 
* Tabelas S-Box AES: Essas tabelas, aes\_sbox e reverse\_aes\_sbox, são usadas nas etapas de substituição da cifragem AES e da decifragem AES, respectivamente. Elas são tabelas de substituição que fornecem substituições para cada byte de dados durante a cifragem e decifragem.
  + Interface gráfica do usuário

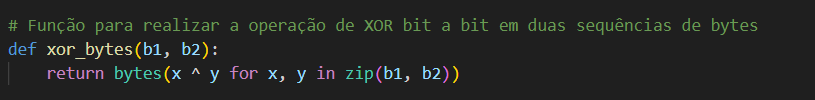
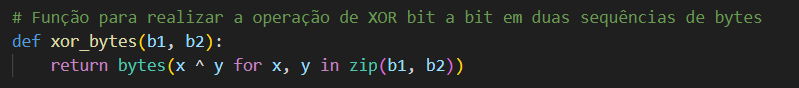
    Descrição gerada automaticamente com confiança média
* consultar e consultar\_reversa: Essas funções são usadas para realizar a substituição de bytes de acordo com as tabelas S-Box durante a cifragem e decifragem.
  + Texto

    Descrição gerada automaticamente
* expandir\_chave: Esta função expande a chave original em subchaves para cada rodada do algoritmo AES. O AES usa subchaves diferentes para cada rodada, e essa função gera essas subchaves a partir da chave original.
  + Texto

    Descrição gerada automaticamente
* rotacionar\_linha\_esquerda, multiplicar\_por\_2, e multiplicar\_por\_3: Essas funções são usadas para realizar operações nas matrizes do estado do AES, como a rotação de linhas e multiplicação de bytes.
  + Tela de computador com texto preto sobre fundo branco

    Descrição gerada automaticamente
* Funções misturar\_colunas e misturar\_coluna: Essas funções são usadas para realizar a etapa de mistura de colunas na cifragem do AES, que é usada para difundir os dados na matriz de estado.
  + Tela de computador com texto preto sobre fundo branco

    Descrição gerada automaticamente
* adicionar\_subchave e extrair\_subchave\_para\_rodada: A função adicionar\_subchave adiciona a subchave atual à matriz de estado, e a função extrair\_subchave\_para\_rodada extrai a subchave relevante para uma rodada específica.
  + Tela de computador com texto preto sobre fundo branco

    Descrição gerada automaticamente
* inverter\_bytes: Esta função inverte a ordem dos bytes em um objeto de bytes. Ela é usada na decifragem para desfazer o resultado da cifragem.
  + 
* xor\_bytes: Essa função realiza a operação XOR bit a bit entre duas sequências de bytes. É usada para combinar a matriz de estado com a subchave.
  + 
* Funções enc e dec: Essas funções implementam as etapas de cifragem e decifragem do AES.
  + A função enc aceita uma chave, dados e o número de rodadas e criptografa os dados usando o AES no modo ECB (Electronic Codebook).
    - Texto

      Descrição gerada automaticamente
  + A função dec aceita uma chave e dados criptografados e realiza a decifragem correspondente.
    - Tela de computador com texto preto sobre fundo branco

      Descrição gerada automaticamente

# AES ECB

Foi pedido a criação de um programa python, que realize a cifração e decifração utilizando o método AES nos modos ECB

* Importação de módulos:
  + O código começa importando funções do módulo C00\_01\_AES\_EBC. Este módulo contém funções relacionadas ao AES e operações de cifragem, como enc (cifragem no modo ECB).
* enc\_ctr:
  + A função enc\_ctr recebe quatro argumentos: chave (a chave de criptografia), dados (os dados que você deseja cifrar), nonce (número usado uma única vez, um valor exclusivo) e num\_rounds (número de rodadas para o AES).
  + O tamanho do bloco AES é definido como 16 bytes.
  + A variável dados\_cifrados é inicializada como uma sequência vazia.
  + Um contador contador é iniciado em 0.
  + O loop é usado para cifrar os dados em blocos de 16 bytes. A condição contador \* tamanho\_bloco < len(dados) verifica se ainda há mais dados para cifrar.
  + Texto

    Descrição gerada automaticamente
* dec\_ctr:
  + A função dec\_ctr é uma função de decifragem que utiliza a mesma lógica da função enc\_ctr. Ela cifra os dados cifrados para obter os dados originais de volta. Portanto, em vez de decifrar explicitamente, a função enc\_ctr é chamada com os dados cifrados e executa a operação de decifragem CTR.
  + Interface gráfica do usuário, Texto

    Descrição gerada automaticamente

# Teste AES ECB

Para o teste do AES ECB, devem ser executados os códigos (em azul):

* Texto

  Descrição gerada automaticamente
  + C01\_01\_cifrar\_texto\_AES\_ECB.py
    - Pega o conteúdo em texto do arquivo **04\_01\_texto\_para\_cifrar.txt** e o cifra, criando o arquivo **04\_02\_texto\_cifrado.txt**
  + C01\_02\_decifrar\_texto\_AES\_ECB.py
    - Pega o conteúdo do arquivo **04\_02\_texto\_cifrado.txt** e o decifra, gerando o arquivo **04\_03\_texto\_decifrado.txt**

# Teste AES CTR - Texto

Para o teste em texto do AES ECB, devem ser executados os códigos (em azul):

* Texto, Carta

  Descrição gerada automaticamente
  + C02\_01\_cifrar\_texto\_AES\_CTR
    - Pega o conteúdo em texto do arquivo **05\_01\_texto\_para\_cifrar\_CTR.txt** e o cifra, criando o arquivo **05\_02\_texto\_cifrado\_CTR.txt**
  + C02\_02\_decifrar\_texto\_AES\_CTR.py
    - Pega o conteúdo do arquivo **05\_02\_texto\_cifrado\_CTR.txt** e o decifra, gerando o arquivo **05\_03\_texto\_decifrado\_CTR.txt**

# Teste AES CTR - Imagem

Para o teste em imagem do AES ECB, devem ser executados os códigos (em azul):

* Tabela

  Descrição gerada automaticamente
  + C03\_01\_cifrar\_imagem\_AES\_CTR.py
    - Pega o conteúdo imagem do arquivo **imagem.png** e faz várias chamadas a função que faz a cifração em modo CTR, passando parâmetros de quantidade de rodadas diferentes.
      * Texto

        Descrição gerada automaticamente
    - Para cada chamada, é criado um novo arquivo **imagem\_cifrada\_X.png** onde X é o parâmetro passada para essa chamada.
  + C03\_02\_decifrar\_imagem\_AES\_CTR.py
    - A função de decifração funciona no mesmo molde, fazendo várias chamadas. Gerando **imagem\_decifrada\_X.png**
    - Texto

      Descrição gerada automaticamente