Universidade de Brasília - UnB
Departamento de Ciência da Computação
Disciplina: Segurança Computacional - CIC0201
Professor: Dr. João José Costa Gondim
Período: 2023/2 - Turma: 2

Alunos: Eder de Amaral Amorim - Matrícula: 170140636
Tais Alves Oliveira - Matrícula: 190117176

Trabalho de Implementação 2: Cifra de bloco e modo de operação CTR

Descritivo

Este projeto é uma implementação da cifragem AES utilizando o modo de operação contador (CTR). A estrutura do código é dividida em 4 arquivos python, descritos abaixo, além de um pasta com arquivos cifrados e decifrados:

1. **Main -** main.py

Este é o ponto de entrada do programa. Ele coordena a execução, chamando funções dos outros módulos conforme necessário. O usuário interage principalmente com este componente, que então interage com os módulos de entrada, AES e CTR para realizar as operações de cifragem e decifragem.

2. **Arquivo de Entrada** - entradas.py

Este arquivo contém as funções e lógicas relacionadas à leitura dos dados que serão cifrados. Ele é responsável por coletar entradas dos usuários, tais como arquivo e/ou texto, chave, vetor de inicialização (nonce) e quantidade de rodadas, e assim, possibilitar a cifragem.

3. **Implementação do AES (bloco= 128 bits, chave 128 bits) -** aes.py

Este arquivo contém a implementação principal do algoritmo de cifragem AES. Ele define as funções para cifrar e decifrar os dados. Os detalhes da cifragem,

como a substituição byte-a-byte, as permutações e as operações de mixagem de colunas, sendo composta pelas seguintes funções:

- *sub_bytes( )*

Função de substituição de bytes. Trabalha com uma matriz de estado 4x4 contendo os dados da entrada, os quais passam por uma substituição usando a S-Box (ou caixa de substituição), na implementação são uma lista de string que representação o número hexadecimal, que substitui os caracteres de entrada pelos caracteres da S-Box.

- *shift_rows( )*

Função para permutação de linhas. A matriz de estado é alterada de modo que as linhas são deslocadas para a esquerda seguindo os seguintes passos: A primeira linha não sofre alterações; a segunda linha é deslocada uma posição; a terceira linha é deslocada duas posições; a quarta linha é deslocada três posições.

- *mix_columns( )*

Função para mistura de colunas. A matriz de estado é misturada com outra matriz predefinida. Isso ocorre nas colunas da matriz. A função utiliza uma função auxiliar *gmul()* que calcula os bits resultantes de multiplicação (XOR) da matriz.

- *add_round_key( )*

Nesta função, a chave da rodada é obtida por meio da expansão da chave original e adicionada à matriz de estado.

- *key_expansion( )*

Aqui a chave original é expandida em subchaves, onde cada rodada recebe uma subchave diferente. Essas subchaves são resultados do deslocamento e substituição de caracteres da chave original.

- *rot_word( )*

Remove o primeiro caractere da chave e adiciona no final. Resultando assim numa rotação de palavras.

- *transpose( )*

Recebe uma matriz e calcula sua transposta (troca linhas por colunas e colunas por linhas), utilizada para facilitar os cálculos em alguns momentos.

- *xor( )*

Função auxiliar utilizada para operação de ‘ou exclusivo’ entre dois valores hexadecimais, fazendo as transformações necessárias para o cálculo.

- *to_matrix( )*

Converte um bloco de dados em uma matriz 4x4.

- *to_text( )*

Recebe uma matriz 4x4 e converte em uma string de texto.

- *estado_rod( )*

Retorna o estado do bloco em cada rodada processando as funções *sub_bytes( ), shift_rows( ), mix_columns( ) e add_round_key( )*

- *estado_rod_inv( )*

Faz uma rodada inversa processando as funções *sub_bytes( ), shift_rows( ) ,mix_columns( ) e add_round_key( )*

- *aes( )*

Essa é a função principal utilizada para cifrar e decifrar. Na cifragem trabalha com substituição, permutação, mesclagem e adição de chaves durante as rodadas. Na decifragem faz o processo inverso.

4. **Implementação do modo de operação CTR** - aes_ctr.py

O modo de operação CTR está implementado no arquivo aes_ctr.py, o qual conta com as seguintes funções para seu funcionamento:

- *cifrar_aes_ctr( )*

  A função *cifrar_aes_ctr( )* recebe como parâmetros: ‘*texto’* que será cifrado, a ‘*chave’* para criptografia AES, e o ‘*nonce’,* que se trata de um valor único que garante que os blocos cifrados sejam diferentes.

  A variável *‘contador’* inicializa com o valor do nonce, que foi informado pelo usuário. O laço de repetição principal itera sobre cada bloco de dados da entrada de 16 bytes (32 caracteres), e o ‘*contador’* é convertido em uma string do mesmo tamanho do bloco.

O contador é passado para a função *aes( )* para cifragem, e um *bloco_cifrado* é retornado. Após isso, o programa faz uma operação de ou exclusivo (xor) entre o '*bloco_cifrado'* e o bloco que não foi cifrado, resultando em um novo bloco, denominado '*bloco_xor',* que é então concatenado à instring 'ciphertext'. Feito isso, o *contador* é incrementado em 1 e é feita uma nova iteração, de modo que os próximos blocos sejam cifrados com valores diferentes do contador. O resultado consiste de uma string que é retornada após ter passado por todos os blocos.

Portanto, de maneira geral, o programa faz o processo de modo que cada bloco é cifrado de forma independente e combinado com a cifragem do contador.

5. Testes

Para os testes de cifragem, fornecemos os seguintes dados;

Entrada: "0123456789abcdeffedcba9876543210"

Chave: "0f1571c947d9e8590cb7add6af7f6798"

Nonce (para o modo CTR): "0f0e0d0c0b0a09080706050403020100"

Utilizamos o padrão de 10 rodadas e obtivemos as seguintes saídas:

Cifragem AES puro: "ff0b844a0853bf7c6934ab4364148fb9"

Cifragem Modo CTR: "f6f1eef8cc3770517a496d11a3f14bf9".

Quanto ao processo de decifragem, este também foi realizado com sucesso e as mensagens originais foram devidamente recuperadas.

Utilizamos também dois arquivos, sendo um deles txt e uma foto jpg, os resultados de cifragem e decifragem subsequentes estão disponíveis na pasta testes. A nomenclatura adotada para nomear os arquivos segue um padrão específico: ela começa com o nome do arquivo, seguido pelo termo "cifragem" (caso tenha sido cifrado), a quantidade de rodadas aplicadas durante o processo de cifragem e o termo "decifragem" (caso tenha sido decifrado) .