Universidade de Brasília – UnB
Instituto de Ciências Exatas – IE
Departamento de Ciência da Computação – CIC
CIC0201 - Segurança da Computação - 2023/2
Professora: João Gondim
Aluno: João Vitor Abadio Siqueira

Descritivo - Trabalho de Implementação 2 - Cifra de Bloco e modo de operação CTR

Para compilar o programa:

gcc aes.c -o aes

Para executar:

./aes

Como o programa funciona

O programa implementa a cifra AES nos modos de bloco ECB e CTR. Ao executar o programa um menu será exibido com as opções de cifração e decifração dos modos mencionados.

O modo de bloco ECB é o modo padrão do AES. O programa pede, tanto na cifração quanto na decifração, os caminhos dos arquivos que se deseja realizar as transformações. Ao executar a cifração ou a decifração no modo ECB, será exibido a chave aleatória utilizada para fins de testes que o monitor ache necessários.

O modo de bloco CTR conta com a chave e o vetor inicial, ao executar a cifração ou a decifração, será exibido a chave e o vetor inicial para os mesmos fins de teste. No vetor inicial, os 12 primeiros bytes (3n/4) são o vetor inicial, o último quarto é usado como contador, como sugerido no livro Katz e Lindell.

unsigned char* cipher_ecb(unsigned char* input, int rounds, unsigned int* key):

A função que realiza a cifração no modo ECB. Segui a risca a documentação do NIST (FIPS 197) para a implementação.

O vetor inicial *input* é transformado em uma matriz 16x16 chamada *state*, e os seguintes passos são seguidos:

1. A transformação de **chave de rodada** 0 é feita;
2. Loop de acordo com a quantidade de *rounds*:
    a. Transformação de **substituição de bytes**: simples substituição na tabela S-box
    b. Transformação de **deslocamento de linhas**: linha 0 se mantém inalterada mas linhas 1-3 são rotacionadas seguindo a operação (linha + coluna) mod 4.
    c. Transformação de **embaralhamento de colunas**: usa uma matriz fixa para realizar uma multiplicação de matrizes sobre o GF(2^8).
    d. Transformação de **chave de rodada** é feita
3. Transformação de **substituição de bytes**

4. Transformação de **deslocamento de linhas**
5. Transformação de **chave de rodada** é feita com o último grupo de chaves

unsigned char* inv_cipher_ecb(unsigned char* input, int rounds, unsigned int* key):

A função que realiza a decifração no modo ECB. Segui a risca a documentação do NIST (FIPS 197) para a implementação.

O vetor inicial *input* é transformado em uma matriz 16x16 chamada *state*, e os seguintes passos são seguidos:

1. A transformação de **chave de rodada** com o último grupo de chaves é feita;
2. Loop de acordo com a quantidade de *rounds*:
    a. Transformação de **substituição de bytes inversa**: simples substituição na tabela S-box inversa
    b. Transformação de **deslocamento de linhas inversa**: linha 0 se mantém inalterada mas linhas 1-3 são rotacionadas seguindo a operação (linha - coluna) mod 4.
    c. Transformação de **embaralhamento de colunas inversa**: usa uma matriz fixa (diferente da anterior) para realizar uma multiplicação de matrizes sobre o GF(2^8).
    d. Transformação de **chave de rodada** é feita
3. Transformação de **substituição de bytes inversa**
4. Transformação de **deslocamento de linhas inversa**
5. Transformação de **chave de rodada** é feita

unsigned char* cipher_ctr(unsigned char* input, int rounds, unsigned int* key, unsigned char* iv)

Realiza a cifração e decifração (são a mesma operação) em modo CTR. Criptografa o vetor inicial usando a função **cipher_ecb()** descrita anteriormente e realiza o XOR com o *input*.

int main():

Na main o programa lida com dois arquivos *fp1* e *fp2* que são de leitura e escrita, respectivamente.

A chave aleatória é gerada e expandida e em seguida o IV é gerado. O programa será sempre executado com a mesma chave e IV, para que se use outra chave e vetor, é necessário sair do programa e executar novamente.

O programa lê e escreve blocos de 16 bytes, sempre antes de uma leitura, o buffer é zerado para que o padding já seja realizado.

OBS: os arquivos cifrados estão na página arquivos.