Segurança computacional
Trabalho 2

Guilherme Mattos - 17/0104508

Introdução

O trabalho a seguir implementa a cifração de um arquivo utilizando o AES - 128 bits. O código foi implementado em C, sem utilização de bibliotecas específicas de criptografia.

Modos de operação

Existem dois modos de operação determinados pelo primeiro argumento na linha de comando:

- Criptografar arquivo: “e”
- Descriptografar arquivo: “d”

Além disso, existe um parâmetro opcional (-v) para o modo verboso, onde todos os passos de criptografia são impressos na tela.

A aplicação funciona lendo um arquivo de input que será criptografado ou descriptografado e um arquivo (.txt)de chave, contendo a representação por caracteres ASCII do valor da chave em hexadecimal . É possível encontrar instruções de compilação e descrição completa dos modos e exemplos no arquivo READ.ME .

Implementação:

A seguir serão discutidos os diversos passos para a criptografia de um input. Para cada passo existe um conjunto de funções implementadas, cada classe de funções pode ser encontrada separadamente no código, dentro de seus marcadores.

1. Ler o arquivo de input e seccioná-lo em blocos de 128 bits, adicionando padding,se necessário

2. Ler o arquivo texto com a chave e transformar a representação de hexadecimal, em ascii, efetivamente nos valores hexadecimais da chave

```
//////////////////////////////////////////// file functions ////////////////////////////////////////////

int max_int(int a, int b);

int min_int(int a, int b);

int file_size(FILE* file_ptr);

void ascci_to_hex(char ascii_key[32],uint32_t key[4]);

void file_to_hex(char* key_file_name, uint32_t key[4]);

Block_Array file_to_block_array(char* input_file_name);

int block_to_file(char* input_file_name,Block_Array block_array_stc, int inverse);



////////////////////////////////////////////////////////////////////////////////////////////////////////
```

header.h - funções de manipulação de arquivo (Passos 1 e 2)

3. Expansão de chave: expandir a chave para o número de rodadas necessário, criando para cada rodada uma sub-chave do tamanho do bloco.

```
void key_expansion(uint32_t key[4],uint32_t expanded_key[44])
```

main.c -função de expansão de senha (Passo 3)

4. Criptografia AES: Para cada bloco são repetidos os seguintes procedimentos
    a. Permutação de colunas (Shift lines)
    b. Substituição pela s-box (substitute_s_box)
    c. XOR do bloco de input com a sub-chave da rodada (add_round_key)
    d. Embaralhamento de colunas (Utilizando campo de Galois) (mix_collum)

```
//////////////////////////////////////////// AES procces functions ////////////////////////////////////////////

void shift_lines(Block_128* block, int inverse)

void substitute_s_box(Block_128* block, int inverse)

void add_round_key(Block_128* block, uint32_t round_key[4])

void mix_collum(Block_128* block,int inverse)
```

main.c - funções de cada processo para o algoritmo AES (Passo 4)

5. Escrita dos dados criptografados no arquivo

```
int block_to_file(char* input_file_name,Block_Array block_array_stc, int inverse);
```

header.h -função de escrita em arquivo (Passo 5)

Considerações finais

Com esse trabalho foi possível se aprofundar nos conhecimentos de criptografia e realmente entender como cada processo opera para a criação de difusão e confusão nos dados.

É possível encontrar o código completo, instruções de uso e exemplos em [Github-Guilherme Camargo]