Practical Work 1

Chaining modes and MAC with AES

Abstract

In this practical work you will program in C language the AES ciphering and deciphering of files with different chaining modes, as well as the computation and verification of a MAC_AES_CBC message authentication code.

Good work!

In this practical work you can use, as basic blocks, the functions `aes_encrypt` and `aes_decrypt` provided in the file “aes.c”. These functions respectively compute an AES ciphertext block and an AES plaintext block given a 128-bit key. They have the following prototypes:

```
void aes_encrypt(BYTE* cipher, BYTE* message, BYTE* key)
void aes_decrypt(BYTE* message, BYTE* cipher, BYTE* key)
```

where `BYTE` is a user-defined type which aliases to `unsigned char` to represent a byte value.

The first parameter is an output parameter. This is a pointer on a 16-byte memory area intended to receive the ciphertext block (resp. the plaintext block). The input parameters `message` (resp. `cipher`) and `key` are pointers on 16-byte memory areas supposed to contain, at function call, the plaintext block (resp. the ciphertext block) and the key.

In the following exercises you must apply a padding to the data before ciphering, in order to obtain a padded data which is an even number of blocks long. You will use the padding scheme defined in Section 6.3 of the RFC 3852 document.

For testing purpose, it is recommended to use in your programs the fixed example key given in Appendix B of the AES standard document FIPS PUB 197. The value of this key is :

```
K = 2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
```

Exercise 1

Write in C language a program which ciphers a file in ECB chaining mode.

Write in C language a program which deciphers a file whose content has been ciphered in ECB chaining mode.

Both programs should be executed by means of the following shell commands:

```
aes-encrypt-ecb input_file output_file
aes-decrypt-ecb input_file output_file
```

Exercise 2

Write in C language a program which ciphers a file in CBC chaining mode[1].

Write in C language a program which deciphers a file whose content has been ciphered in CBC chaining mode.

Both programs should be executed by means of the following shell commands:

```
aes-encrypt-cbc input_file output_file
aes-decrypt-cbc input_file output_file
```

Exercise 3

Write in C language a program which computes the MAC_AES_CBC message authentication code of a file, and outputs it in hexadecimal form.

This program should be executed by means of the following shell command:

```
mac-aes-cbc input_file
```

Exercise 4

Write in C language a program which verifies the consistency between a file and a MAC value given as a string made of only hexadecimal digits. Your program must output a message informing whether the MAC is correctly verified for this file or not.

This program should be executed by means of the following shell command:

```
verify-mac-aes-cbc input_file mac_value
```

[1] For all CBC chaining programs, it is suggested to use the null initialisation vector which is 16 zero bytes.