Entorno

OS: Ubuntu 18.04
Python: 3.6.4
IDE: Visual Studio Code

Tarea T1a

Siguiendo los ejemplos de la unidad 2 y la documentación oficial de la librería cryptography he codificado un script que encripta el texto 'a secret message' usando cifrado AES + CBC, lo desencripta y compara el resultado final con el el texto original.

Aunque la clave es siempre la misma, al generar un vector de inicialización pseudo aleatorio, el criptograma obtenido en cada ejecución es diferente.

- Captura de pantalla: T1a.png
- Salida consola: T1a.output.txt

Tarea T1b

El script usado es similar al de la tarea T1a con pequeñas modificaciones para cambiar el modo de cifrado del bloque. En el caso ECB no es necesario usar vector de inicialización.

- ECB
    - Captura de pantalla: T1b_ECB.png
    - Salida consola: T1b_ECB.output.txt

- OFB
    - Captura de pantalla: T1b_OFB.png
    - Salida consola: T1b_OFB.output.txt

- CFB
    - Captura de pantalla: T1b_CFB.png
    - Salida consola: T1b_CFB.output.txt

En el caso ECB al usar una clave estática, el criptograma obtenido es siempre el mismo.

En el caso OFB y CFB el criptograma varía en cada ejecución debido al uso de un vector de inicialización pseudo aleatorio.

Tarea T2

En este caso he creado un script que lee como argumento desde consola el fichero del que queremos obtener el hash MD5.

```
# python3 T2.py WinMD5.exe
INPUT: WinMD5.exe  <--> MD5 hash: 944a1e869969dd8a4b64ca5e6ebc209a
```

- Captura de pantalla: T2.png
- Salida consola: T2.output.txt
- Hashes obtenidos:
    - WinMD5.exe: 944a1e869969dd8a4b64ca5e6ebc209a
    - WinMD5_2.exe: d1284c8060db2d9e8960696b6b8ccfc4

El fichero correcto es “WinMD5.exe”.