Criptografía y Cifrado de Información 2021

Lab 05

19.agosto.2021

En este laboratorio implementaremos un cifrado AES. Usaremos este cifrado para encriptar y decriptar un archivo de texto.

1. Implementar un cifrado AES. Para ello, puede utilizar la librería de Python **pyCryptoDome**, la cual ya contiene varias implementaciones de cifrados importantes.
   La documentación de PyCryptoDome:
   `https://pycryptodome.readthedocs.io/en/latest/src/introduction.html`
   `https://pycryptodome.readthedocs.io/en/latest/src/cipher/classic.html`

   Esta implementación de AES, una una clave *key* de 16 bytes (128 bits), y usa bloques de tamaño 16 bytes (128 bits).

2. Experimente las rutinas `AES.encrypt()` y `AES.decrypt()`, para 3 ejemplos sencillos de cadenas de texto. Para cada ejemplo, explore diferentes modos de encripción, y observe las diferencias en los cifrados generados.

   En el camino tendrán que implementar funciones que conviertan cadenas de texto a formato *hex* y formato *bytes*, y de estos formatos de vuelta a cadenas texto.

3. Implemente un método cifrado AES, para encriptar archivos de texto.
   Para ello, deberá construir dos funciones, una para encriptado, y otra para decriptado.
   La función de encriptado debe recibir los siguientes parámetros:

   - la clave *key*, de 16 bytes (en formato bytes).
   - el nombre o ubicación de archivo de texto a encriptar.
   - el nombre o ubicación del archivo cifrado (sugerencia: usualmente conviene dejar el mismo nombre de archivo origen, y basta cambiar la extensión a `'.enc'` o similar.

   La función de decriptado debe recibir los siguientes parámetros:

   - la clave *key*, de 16 bytes (en formato bytes). Debe ser la misma que se usó en la encripción.
   - el nombre o ubicación de archivo de texto a decriptar.
   - el nombre o ubicación del archivo descifrado (sugerencia: usualmente conviene dejar el mismo nombre de archivo origen, y basta cambiar la extensión a `'.dec'` o similar.

   Para que sus implementaciones funcionen, deberá implementar un esquema de partición del archivo en bloques de 16 bytes. Cada uno de estos bloques entrará a los métodos `AES.encrypt()` o `AES.decrypt()`, y hara el cifrado por bloques.
   Recuerde utilizar un método de *padding* adecuado (por ejemplo, el esquema PCKS5) para completar el tamaño del último bloque.

   Puede utilizar cualquiera de los modos: OFB, CBC, CTR. Cuando corresponda deberá construir el IV o indicar la rutina de conteo.

   Implemente sus funciones en un archivo de texto de su elección. Compare el resultado de su archivo descifrado `.dec` contra el archivo origen `.txt` para ver si el contenido coincide.
   Junto con el código, debe incluir este archivo de texto `.txt`, así como sus archivos cifrado y descifrado (`.enc` y `.dec`). Deberá indicar en el reporte los parámetros usados para la encripción.