Criptografia de clave secreta

Daniel Otero Avalle

Facultadad de Informática de Barcelona
Universidad Politécnica de Cataluña

I. Nota previa

Para responder a las preguntas planteadas se han realizado varios scripts en python y modificaciones de una implementación del aes de libre distribución también implementado en python que se ha adjuntado a los ficheros de esta práctica con el nombre aes.py y que puede también encontrarse en la siguiente dirección url `http://anh.cs.luc.edu/331/code/aes.py`.

II. Pregunta 1

.1. Apartado a

El script pregunta1a.py de la carpeta pregunta_1_a muestra que si la función subBytes se implementa como la identidad, dado tres mensajes donde dos de ellos difieren de uno original en 1 bits y el tercero en los dos bits de los dos anteriores respecto el original, el resultado de hacer la xor entre los tres mensajes resulta en el mensaje original. El script también muestra que este comportamiento no pasa con la implementación original de subBytes.

.2. Apartado b

El modificar shiftRows por la identidad provoca la pérdida de la propiedad de difusión de la información del aes. Como se muestra en el script pregunta1b.py de la carpeta pregunta_1_b, si ciframos tres mensajes muy similares (la letra a, luego la b y luego la c en nuestro caso ya que a nivel de bits son muy similares), los criptogramas resultantes tienen una gran similitud entre si en gran parte de ellos mientras que si los mismos mensajes son cifrados con el aes original los criptogramas resultates son completamente distintos entre si pese a que la gran mayoría del texto a cifrar en los tres mensajes sea el mismo.

.3. Apartado c

Si cambiamos la función mixColumns de manera que actúe como la identidad se produce el mismo efecto descrito en el apartado b; se pierde la propiedad de difusión de la información en el mensaje. El script pregunta1c.py de la carpeta pregunta_1_c muestra tal comportamiento contrastándolo con el aes original. También se han usado los mismos mensajes que en el apartado b para mostrar el comportamiento ya que muestran bien el fenómeno que se intenta enseñar.

III. Pregunta 2

Para descifrar el mensaje se ha usado el script aes_modp2.py situado en la carpeta pregunta_2. El mensaje descifrado también se encuentra en la misma carpeta con el nombre output.html.

IV. Pregunta 3

El proceso de descifrado de este archivo se ha basado en encontrar una debilidad en la seguirdad de cifrado. Me he fijado que la clave se podía obtener haciendo una xor a todos los valores del vector de inicialización con un número que podía tomar un valor entre 0 y 255. Me he dado cuenta que con la mayoría de las claves el algoritmo de descifrado generaba un error así que he hecho un script en python llamado script.py que probaba con todas las 256 claves posibles descifrar el criptograma para captar los que no generaban error. De ese subgrupo ya mas reducido he comprovado personalmente que el resultado de descifrar fuese algo coherente, es decir, que no fuesen caracteres sin sentido, hasta dar con el mensaje original. Para descifrar también he hecho servir un script en pyhton llamado aes_modp3.py. Ambos scripts mencionados en este apartado se encuentran en la subcarpeta pregunta_3.