Antonio Guilera Domingo
Lluís Marquès i Peñaranda

Informe Práctica 2

## 1 El cuerpo finito GF(2^8)

Todas las están implementadas en el fichero gf_functs y la comparación de tiempo entre gf_product_p i gf_product_t se encuentra en el fichero comparativa.txt.

## 2 AES

2.1

El algoritmo AES que hemos usado para esta parte es una implementación Java que se encuentra en la carpeta AES

Hicimos los cambios correspondientes al código de AES para ver cómo afectaba esto al cifrado

2.1.1

Comprobamos como en el AES original cambiar un bit del mensaje modificaba todo el mensaje cifrado, en cambio sustituyendo SubBytes por la identidad solo se ven afectados 4 bytes (de forma diagonal, es decir aunque sea poco todas las filas y columnas se ven modificadas) (Ejemplo de ejecución para el original en CifradoAESLegit.txt y ejemplo de ejecución para la versión sin SubBytes en CifradoAESMod.txt)

2.1.2

Modificando ShiftRows por la identidad los efectos que tenemos son que al modificar un bit del mensaje el cambio no se propaga al resto de columnas del cifrado y se vuelve más predecible.  (Ejemplo de ejecución en ShiftRowsIdentidad.txt)

2.1.3

Hicimos el mismo proceso pero para MixColumns, los efectos fueron que al modificar un bit del mensaje únicamente se modifica un byte en el cifrado. (Ejemplo de ejecución y conclusiones generales en MixColumnsIdentidad.txt)

2.2

En este apartado hicimos los histogramas correspondientes al modificar clave y mensaje, y obtuvimos como conclusión que es equiprobable que al cambiar un bit del mensaje cambie uno en el cifrado, y que todas las posiciones del mensaje tienen una frecuencia de modificación similar. (Los histogramas se encuentran en los ficheros Histograma_canvis_bits, Histograma_canvis_posicions tanto para canvios en mensaje como para clave respectivamente)

2.3

Hicimos un código en Python para automatizar el proceso haciendo 3000000 repeticiones, generando claves y mensajes aleatorios (código coreespondiente en max0.py) . Máximo Número de ceros : 22 Con el mensaje : ec2ba7d9796d75a6e5aadd50f3b6d10f y la clave 0baa8df9339e3314ca63968eda4c89bd

**3 Criptografía de clave secreta**

Esta última parte era descifrar los archivos que nos enviaron. Adjuntamos el código de descifrado en Python (archivos ex31.py y ex32.py) y los archivos resultantes del descifrado (decryptedPractica2Lluis, decryptedPractica2Toni son el descifrado del primer fichero, del cual ya teniamos la clave y decrypetedPractica23LluisFINAL, decrypetedPractica23ToniFINAL son el descifrado del segundo fichero puertatrasera)