ENTREGABLE LAB2:
Criptografia de clau secreta
Criptografia

Paula Gené paula.gene@estudintat.upc.edu
Anna Llanza anna.llanza@estudiantat.upc.edu

QT 2020-2021 23/10/2020

1. El cos finit GF

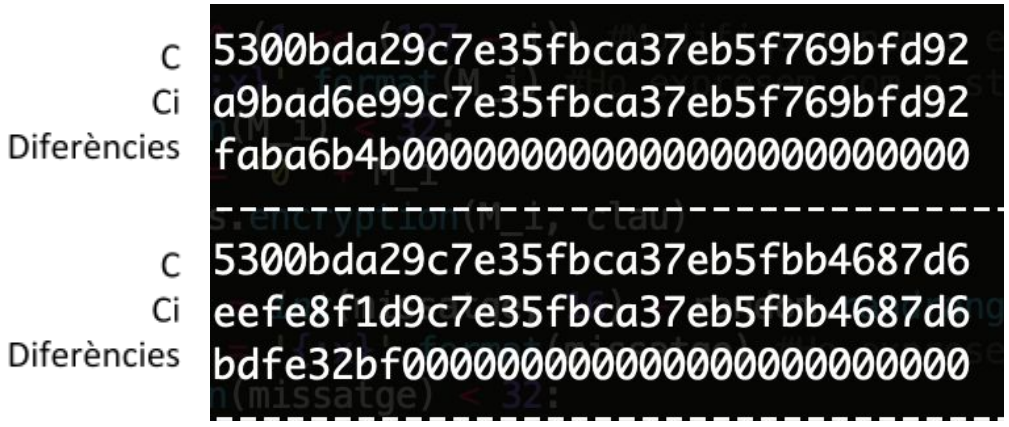
Feu taules comparatives dels temps d'execució fent servir les diferents funcions:

```
GF_product_p de a = 42, b = 2: temps = 1.0967254638671875e-05 a * b = 84
GF_product_t de a = 42, b = 2: temps = 9.5367431640625e-07 a * b = 84

GF_product_p de a = 42, b = 3: temps = 1.8835067749023438e-05 a * b = 126
GF_product_t de a = 42, b = 3: temps = 9.5367431640625e-07 a * b = 126

GF_product_p de a = 42, b = 9: temps = 2.09808349609375e-05 a * b = 103
GF_product_t de a = 42, b = 9: temps = 9.5367431640625e-07 a * b = 103

GF_product_p de a = 42, b = 11: temps = 2.5033950805664062e-05 a * b = 51
GF_product_t de a = 42, b = 11: temps = 1.1920928955078125e-06 a * b = 51

GF_product_p de a = 42, b = 13: temps = 7.033348083496094e-05 a * b = 207
GF_product_t de a = 42, b = 13: temps = 1.6689300537109375e-06 a * b = 207

GF_product_p de a = 42, b = 14: temps = 5.507469177246094e-05 a * b = 177
GF_product_t de a = 42, b = 14: temps = 1.9073486328125e-06 a * b = 177
```

2. Advanced Encryption Standard (AES)

2.1. Efectes de les funcions elementals

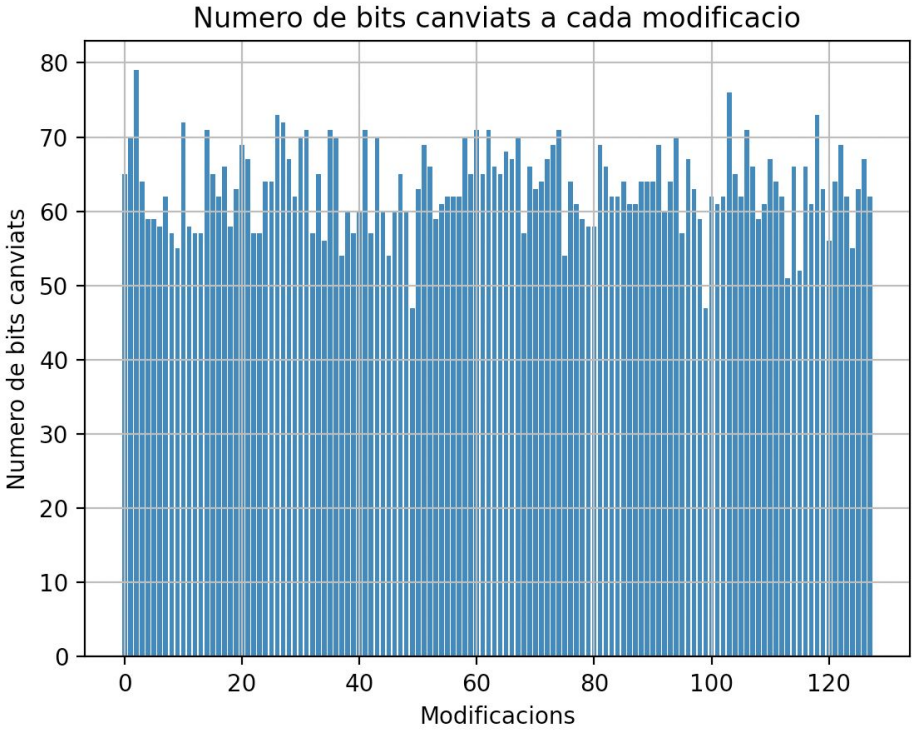
Canviant la funció de ShiftRows per la identitat, podem comprovar que el bloc C i Ci són molt similars, ja que al realitzar la XOR entre aquests dos blocs, el resultat obtingut té molts valors iguals a 0.

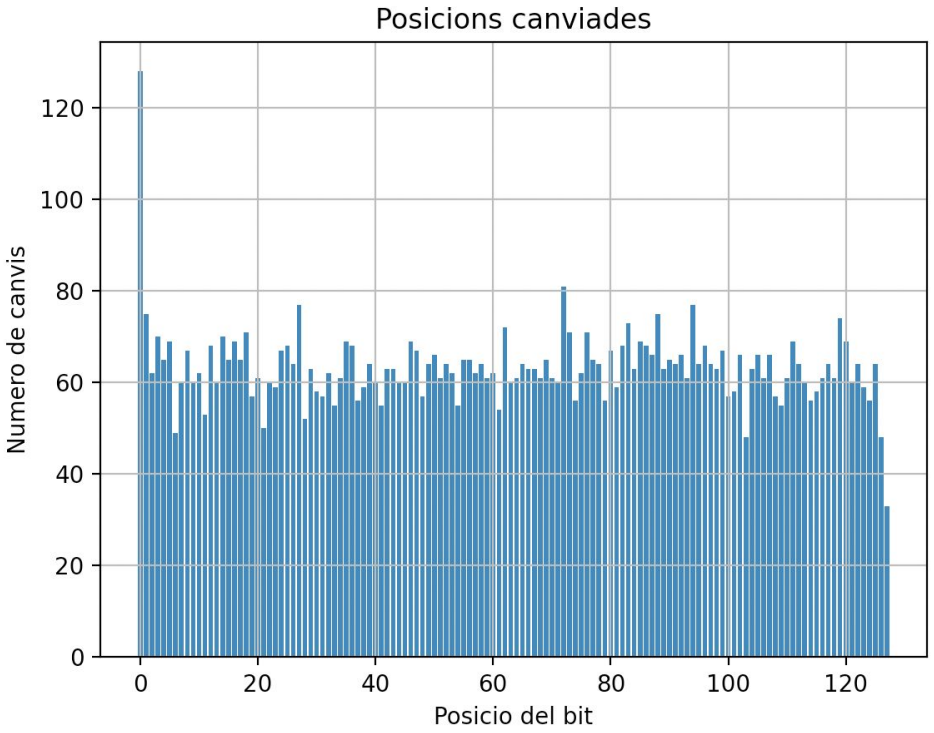


Canviant la funció de MixColumns per la identitat, al realitzar la XOR per veure les diferències entre els dos blocs, hem obtingut un resultat encara més similar que en l'anterior cas.

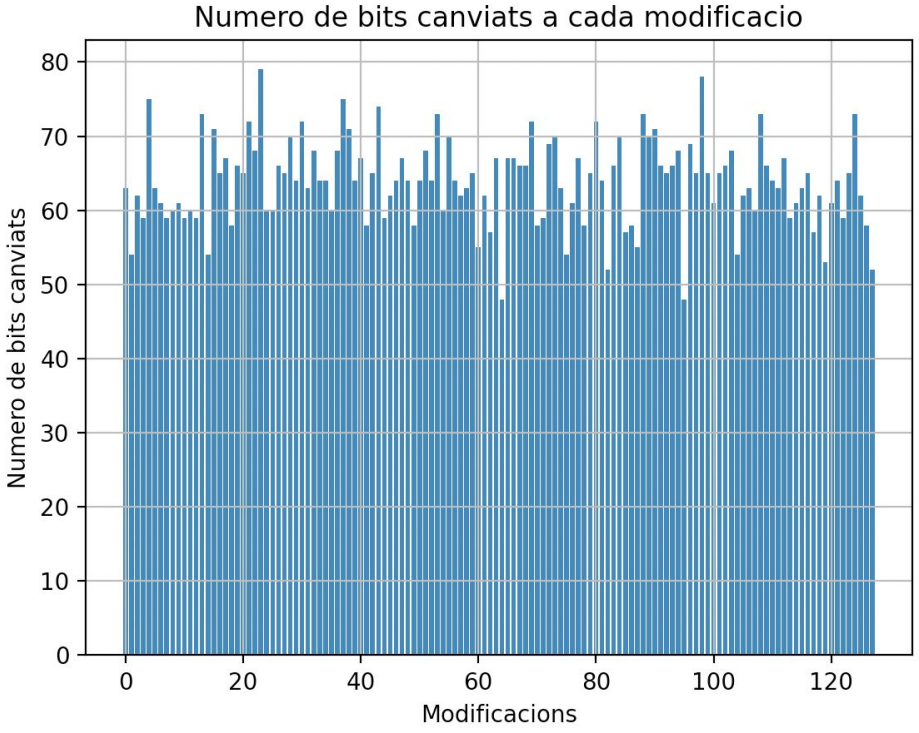
2.2. Propagació de petits canvis

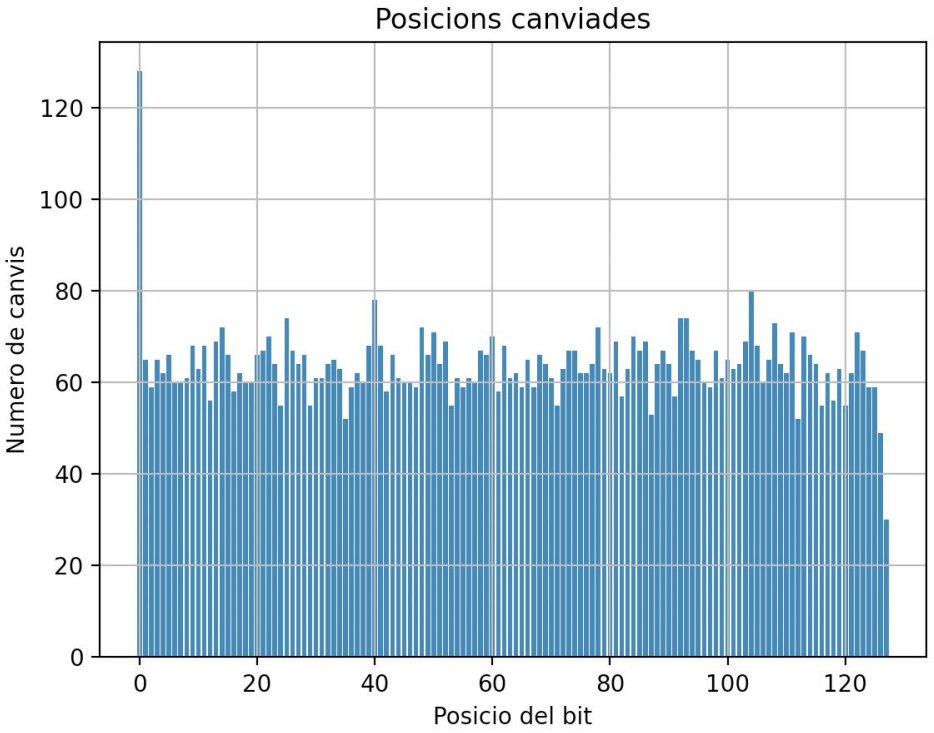
Modificació bit bloc M





Modificació bit bloc K





2.3. Ús com a funció unidireccional

- Per a obtenir el màxim nombre de zeros inicials en el missatge encriptat C hem implementat una funció que fixat el missatge M = 0x00112233445566778899AABBCCDDEEFF i inicialitzada la clau K = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF , aquest es va disminuint una unitat en cada iteració, i de totes aquestes es guarda la clau i el criptograma que més zeros inicials té. Hem fet la prova amb 20000000 iteracions i hem obtingut la clau K = 0xFFFFFFFFFFFFFFFFFFFFFFFFF287CB2, que genera el criptograma C = 0x0000000D4143BD2643D4B6C7A8A54E70, que permet obtenir 28 bits inicials amb el valor zero. Si augmentéssim el nombre d'iteracions fins a poder fer suficients variacions de la clau, arribariem a trobar l'òptima que ens permetria obtenir el nombre màxim de zeros.
- En aquest cas, a diferència de l'anterior, ens hem adonat que podem fer servir el mètode de desencriptar per tal d'aconseguir en nostre objectiu. Fixada la clau K = 0x0123456789ABCDEFFEDCBA9876543210, per tal d'obtenir el missatge xifrat C = 0x00000000000000000000000000000000 hem fet servir el mètode de desencriptar i hem obtingut el missatge M = a9c1017e612ff97efd7587cdd073cb50. D'aquesta manera no hem hagut de fer proves, i hem obtingut el màxim nombre de zeros, que és 128.
- Per aquest últim apartat hem utilitzat el mateix mètode que en l'anterior, i per fer la comprovació hem utilitzat una clau diferent K = 0x0425456799ABCBEFBEECBA9276583280, i hem obtingut el missatge M =

0x9e3fa2f672111cb397d149cb652d10d7, que permet que el missatge xifrat sigui igual a C = 0x00000000000000000000000000000000. Es podria provar amb les claus que vulguéssim que el programa sempre trobaria una M per tal que es complís la restricció del missatge xifrat.