Après avoir chargé les 3 fichiers sources, la première étape consiste à initialiser une subb_key de 256 valeurs entre 0 et 255. Ensuite, on initialise une matrice de 1000 lignes et 256 colonnes qui nous servira tout au long du traitement.

Le traitement se déroule en plusieurs étapes :

1. AddRoundKey
2. SubBytes
3. Calculs des poids de Hamming
4. Calcul de la corrélation
5. Identification de la clef
6. Affichage des données dans des graphes

AddRoundKey

A cette étape, pour Chaque ligne de la matrice, chaque valeur est remplacée par un ou exclusif entre la valeur de Input1 correspondant à l'indice de la ligne en question et la valeur de la subb_key correspondant à la colonne de la valeur en question.

Exemple :

$$matrice(i,j)$$

$$= \begin{pmatrix} Input1(1) \oplus subb\_key(1) & \cdots & Input1(1) \oplus subb\_key(256) \\ \vdots & \ddots & \vdots \\ Input1(1000) \oplus subb\_key(1) & \cdots & Input1(1000) \oplus subb\_key(256) \end{pmatrix}$$

Pour réaliser cela, j'ai fait le choix de parcourir à l'aide de 2 boucles for les indices des lignes et colonnes de la matrice.

SubBytes

Cette étape consiste à remplacer chaque valeur de la matrice obtenue à l'étape précédente par sa correspondance dans le vecteur SubBytes. Il a fallu prendre en compte que nos valeurs vont de 0 à 255 tandis que les indices d'une liste sur Matlab® vont de 1 à 256. Pour cela, lors de la correspondance, il a suffi de décaler de 1 la valeur pour obtenir la bonne correspondance.

Exemple :
Si

$$matrice(i,j) = 34$$

Alors, au lieu de faire

$$SubBytes(\ matrice(i,j)\ )$$

On fait,

$$SubBytes(\ matrice(i,j) + 1\ )$$

De manière à obtenir la valeur à l'indice 35, qui est la valeur recherchée.

Calculs des poids de Hamming

Le calcul des poids de Hamming consiste à retourner le nombre de 1 dans les valeurs binaires de la matrice. Les valeurs de la matrice étant codées sur 8 bits, j'ai fait le choix de parcourir à l'aide d'une boucle l'ensemble des bits de chaque valeur, et de compter le nombre de 1 présents dans chaque position de la matrice.
Ensuite, il a fallu remplacer dans la matrice chaque valeur par son nombre de 1 correspondant.

Calcul de la corrélation

Le calcul de la corrélation se fait à l'aide de la fonction corrcoef de Matlab®. Celle-ci prend 2 paramètres :

- La matrice utilisée depuis le début, contenant les poids de hamming de dimension 1000 lignes et 256 colonnes
- La matrice traces fournie dans le projet, de dimension 1000 lignes et 512 colonnes.

Le calcul prend en paramètre une colonne de la matrice et une colonne de traces.
Avec des itérations, on calcule le coefficient pour chaque colonne de la matrice, avec chaque colonne de traces

Exemple :

$$corrcoef\big(matrice(:,1), traces(:,1)\big)$$
$$\vdots$$
$$corrcoef\,(matrice(:,1), traces(:,512)$$

$$\vdots$$
$$\vdots$$

$$corrcoef\big(matrice(:,256), traces(:,1)\big)$$
$$\vdots$$
$$corrcoef\,(matrice(:,256), traces(:,512)$$

La fonction $corrcoef\,()$ retourne une matrice de dimension (2, 2) de la forme :

$$\begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix}$$

Les valeurs aux positions (1, 2) et (2, 1) sont identiques. Il suffit d'en garder une et de calculer sa valeur absolue.
Ensuite, chaque valeur obtenue est stockée dans une matrice de dimension 512 colonnes x 256 lignes.
On crée par la suite un vecteur contenant la valeur maximale de chaque ligne de la matrice créé à l'étape précédente. Ce vecteur contient donc 256 valeurs.

Identification de la clef

Pour trouver la clef, il faut partir du vecteur qui a été créé juste avant. Il suffit d'identifier la valeur maximale de ce vecteur et de trouver son index dans le vecteur.

Dans notre cas, la valeur maximale de ce vecteur est 0,1490.

Cette valeur se trouve à l'index 44 du vecteur. Comme expliqué précédemment, il y a un décalage de 1 entre les indices de Matlab® et les indices de nos valeurs.
Il fait donc soustraire 1 à l'index pour obtenir une valeur entre 0 et 255 et non plus entre 1 et 256.

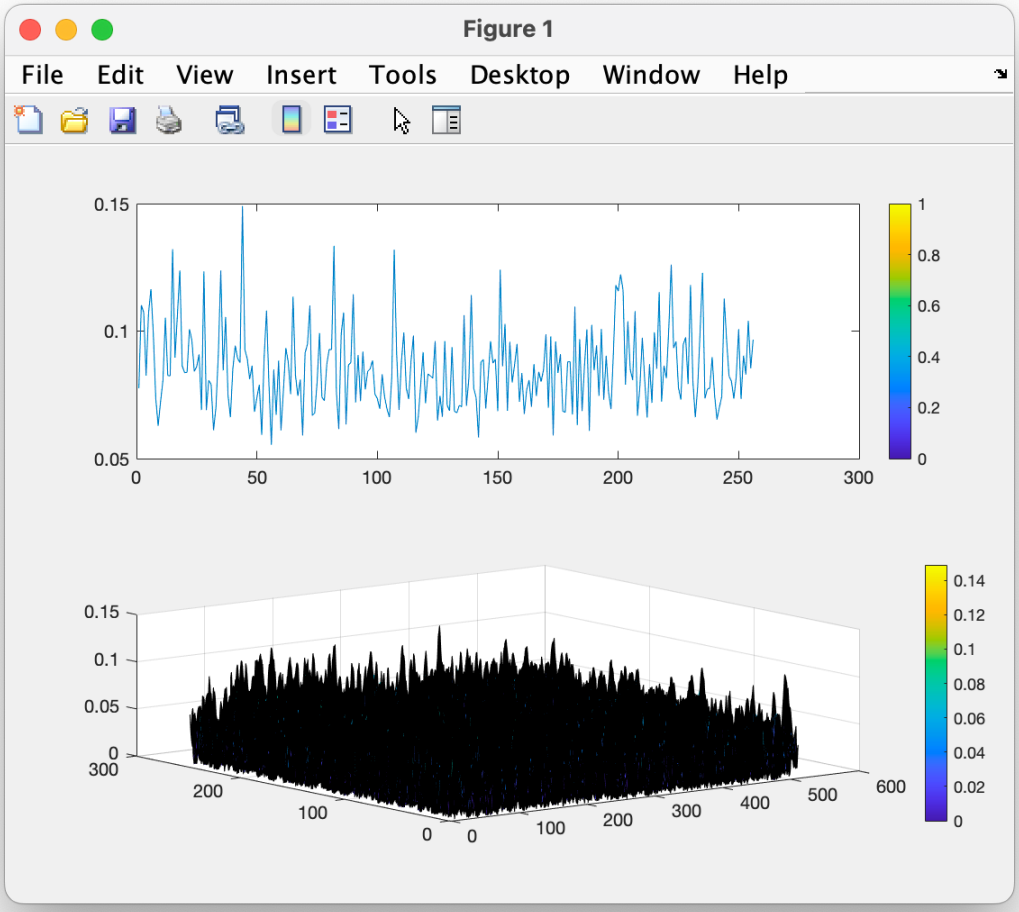
D'où 44 – 1 = 43

Donc : la clef est 43.

Affichage des données dans des graphes

Pour afficher les 2 graphes demandés, j'ai utilisé les fonctions suivantes :
tiledlayout(2,1) : qui permet d'afficher les 2 graphes sur la même fenêtre
plot() : pour afficher le graphe 2D
surf() : pour afficher le graphe 3D

Les graphes obtenus sont les suivants :



On constate bien qu'il y a un pic à la valeur 0,1490, qui sert à déterminer la clef.