



Broken AES Cipher E-CORP

D0pp3lgang3r

17 mai 2022

1 Introduction

Bonjour, chef je vous envoie le nouveau type de chiffrement par block que j'ai inventé, pour E-CORP.

Est-ce que vous pensez que l'algorithme est cassable ? Essayez par vous même et redites moi !

2 Algorithme de chiffrement par bloc

2.1 Divisé le message en matrice

On commence par prendre 64 octets du message en clair, ensuite nous prenons la valeur décimale de chacun des octets pour regrouper ces valeurs dans une grande matrice que l'on appellera "M"

$$M_{8,8} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,8} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ a_{8,1} & a_{8,2} & \cdots & a_{8,8} \end{pmatrix}$$

Puis nous créons 4 matrices 4x4 {M1, M2, M3, M4} et placons les coefficients de notre matrice M dans chacune d'entre elles :

$$M1_{4,4} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix}, \quad M2_{4,4} = \begin{pmatrix} a_{1,5} & a_{1,6} & a_{1,7} & a_{1,8} \\ a_{2,5} & a_{2,6} & a_{2,7} & a_{2,8} \\ a_{3,5} & a_{3,6} & a_{3,7} & a_{3,8} \\ a_{4,5} & a_{4,6} & a_{4,7} & a_{4,8} \end{pmatrix}$$

$$M3_{4,4} = \begin{pmatrix} a_{5,1} & a_{5,2} & a_{5,3} & a_{5,4} \\ a_{6,1} & a_{6,2} & a_{6,3} & a_{6,4} \\ a_{7,1} & a_{7,2} & a_{7,3} & a_{7,4} \\ a_{8,1} & a_{8,2} & a_{8,3} & a_{8,4} \end{pmatrix}, \quad M4_{4,4} = \begin{pmatrix} a_{5,5} & a_{5,6} & a_{5,7} & a_{5,8} \\ a_{6,5} & a_{6,6} & a_{6,7} & a_{6,8} \\ a_{7,5} & a_{7,6} & a_{7,7} & a_{7,8} \\ a_{8,5} & a_{8,6} & a_{8,7} & a_{8,8} \end{pmatrix}$$

Par la suite nous créons la clé de chiffrement "K" qui sera une matrice 4x4 avec ces coefficients :

$$K = \begin{pmatrix} 36 & 33 & 36 & 35 \\ 40 & 60 & 62 & 41 \\ 38 & 35 & 43 & 42 \\ 45 & 47 & 40 & 41 \end{pmatrix}$$

2.2 Algorithme

Maintenant que nous avons initialisé toutes les variables de l'algorithme, que diriez-vous de comprendre ce qu'il fait ?

2.2.1 Rotations

La première étape consiste à effectuer une rotation gauche de nos sous matrices {M1, M2, M3, M4}, ainsi M1 devient :

$$M1_{4,4} = \begin{pmatrix} a_{1,4} & a_{2,4} & a_{3,4} & a_{4,4} \\ a_{1,3} & a_{2,3} & a_{3,3} & a_{4,3} \\ a_{1,2} & a_{2,2} & a_{3,2} & a_{4,2} \\ a_{1,1} & a_{2,1} & a_{3,1} & a_{4,1} \end{pmatrix}$$

⚠ Nous répétons cette opération pour chaque sous matrice.

2.2.2 XOR

Après quoi nous utilisons l'opération XOR entre chaque coefficients de nos sous matrices et les coefficients de la clé, on a donc M1 :

$$M1_{4,4} = \begin{pmatrix} a_{1,4} \oplus 36 & a_{2,4} \oplus 33 & a_{3,4} \oplus 36 & a_{4,4} \oplus 35 \\ a_{1,3} \oplus 40 & a_{2,3} \oplus 60 & a_{3,3} \oplus 62 & a_{4,3} \oplus 41 \\ a_{1,2} \oplus 38 & a_{2,2} \oplus 35 & a_{3,2} \oplus 43 & a_{4,2} \oplus 42 \\ a_{1,1} \oplus 45 & a_{2,1} \oplus 47 & a_{3,1} \oplus 40 & a_{4,1} \oplus 41 \end{pmatrix}$$

⚠ Nous faisons de même pour chaque sous matrice.

2.2.3 L'addition de la Trace

Ensuite nous améliorons la "sécurité" de notre chiffrement en ajoutant la trace de notre matrice K, divisé par 2 [soit $tr = \frac{trace(K)}{2}$], à chaque coefficient xorés, comme vu précédemment, on a donc M1 :

$$M1_{4,4} = \begin{pmatrix} (a_{1,4} \oplus 36) + tr & (a_{2,4} \oplus 33) + tr & (a_{3,4} \oplus 36) + tr & (a_{4,4} \oplus 35) + tr \\ (a_{1,3} \oplus 40) + tr & (a_{2,3} \oplus 60) + tr & (a_{3,3} \oplus 62) + tr & (a_{4,3} \oplus 41) + tr \\ (a_{1,2} \oplus 38) + tr & (a_{2,2} \oplus 35) + tr & (a_{3,2} \oplus 43) + tr & (a_{4,2} \oplus 42) + tr \\ (a_{1,1} \oplus 45) + tr & (a_{2,1} \oplus 47) + tr & (a_{3,1} \oplus 40) + tr & (a_{4,1} \oplus 41) + tr \end{pmatrix}$$

⚠ Nous faisons de même pour chaque sous matrice.

2.2.4 Inversion

Par la suite nous inversons les coefficients $k_{i,j}$, que nous avons obtenus des opérations précédentes, donc $(a_{1,4} \oplus 36) + tr$ devient $k_{1,1}$, ainsi on a M1 :

$$M1_{4,4} = \begin{pmatrix} k_{4,4} & k_{4,3} & k_{4,2} & k_{4,1} \\ k_{3,4} & k_{3,3} & k_{3,2} & k_{3,1} \\ k_{2,4} & k_{2,3} & k_{2,2} & k_{2,1} \\ k_{1,4} & k_{1,3} & k_{1,2} & k_{1,1} \end{pmatrix}$$

⚠ Ne pas oublier de faire de même avec chaque matrice !

2.2.5 Ecrire les données

Voici la fonction qui écrit les données de manière séquentielle !

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define SUB_MATRIX_N 4
5 #define SUB_MATRIX_COL 4
6 #define SUB_MATRIX_ROW 4
7
8 void write_ciphered(Matrix **ma_tab, FILE *fp)
9 {
10     // ma_tab is the table of matrices {M1, M2, M3, M4}
11     for (int k=0;k<SUB_MATRIX_N;k++)
12     {
13         for (int i=0;i<SUB_MATRIX_ROW;i++)
14         {
15             for (int j=0;j<SUB_MATRIX_COL;j++)
16             {
17                 putc((int)ma_tab[k]->matrix[i][j], fp);
18             }
19         }
20     }
21     fclose(fp);
22 }
```

Vous avez besoin d'aide ? N'hésitez pas à me contacter [Doppelganger#5878](#) sur Brainshell !