ISIFC 3-cryptographie : chiffrement symétrique, TD

Jean-François COUCHOT
couchot[arobase]femto-st[point]fr

1 Ordres de grandeur

On commence un petit exercice calculatoire pour ensuite évaluer le temps moyen pour trouver la clef de chiffrement lorsqu'on dispose d'un couple (clair, chiffré) et où l'algorithme de chiffrement est AES. Ces deux exercices sont inspirés de [1].

Exercice 1.1 (Vider l'océan avec un dé à coudre). *On considère qu'un dé à coudre est un cylindre de 1,5cm de hauteur pour 1,5cm de diamètre. On rappelle que le volume d'un cylindre de rayon R et de hauteur h s'exprime sous la forme $V = \pi R^2 h$. Selon l'Institut Français des Mers, les océans couvrent 360 millions de km² avec une profondeur moyenne de 3800m.*

Encadrer entre deux puissances de 2 consécutives le nombre de dés à coudre d'eau que contiennent les océans.

Exercice 1.2 (La force brute pour décrypter AES). . *Le facteur de travail d'un algorithme est le nombre d'instructions élémentaires nécessaires à son exécution. La puissance d'une machine est le nombre d'instructions qu'elle exécute par unité de temps. La puissance d'un PC actuel (en 2021) est d'environ 2,4M Mips (millions d'instructions par secondes). Le facteur de travail d'un algorithme optimisé pour tester une clé de 128bits de l'algorithme AES est d'environ 1200 instructions élémentaires. On dispose d'un couple (clair,chiffré) connu et on désire trouver la clé utilisée par force brute, c'est-à-dire en testant toutes les clés les unes après les autres. Une clé est constituée d'un mot de 128 symboles binaires. On suppose que toutes les clés sont équiprobables.*

1. *En combien de temps t une machine actuelle teste-t-elle une clé ?*
2. *Combien y a-t-il de clés possibles ?*
3. *Quel est le nombre moyen E de clés à tester avant de trouver la bonne ?*
4. *Quel est le temps de travail moyen pour trouver la clé sur un processeur actuel ? On pourra comparer ceci à l'âge de l'univers (environs 13 milliards d'années). Et le nombre de dés à coudre d'eau que contiennent les océans ?*

2 AES

D'un point de vue pratique, AES représente chaque octet (8 bits) sous la forme d'un mot de deux chiffres hexadécimaux. Théoriquement, AES code chaque octet à l'aide d'un polynome de degré inférieur ou égal à 7, où chaque coefficient est 0 ou 1 et les calculs se font modulo le polynome de de Rijndael $R(X) = X^8 + X^4 + X^3 + X + 1$. L'addition correspond au "ou exclusif" sur les mots binaires .

Exercice 2.1 (Petits calculs dans $\mathbb{F}_{2^8}$).

1. *Représenter à l'aide de deux chiffres hexadécimaux le mot binaire "1001 1010"*
2. *Représenter à l'aide de polynômes de degré inférieur ou égal à 7 les mots "2A" et "37".*
3. *Calculer la somme de "2A" et "37".*
4. *Calculer les produits "01"×"37", "2A"×"37", "48"×"3F".*

Exercice 2.2 (L'opération Mixcolumn). *L'opération Micolumns opère colonne par colonne. Pour un vecteur (a_0, a_1, a_2, a_3) elle effectue le calcul*

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

1. *Calculer l'image de la colonne $(01, 01, 01, 01)$.*
2. *Calculer l'image de la colonne $(DB, 13, 53, 45)$*

1. Inspiré de https://www.di.ens.fr/~nitulesc/files/CRYPTO13/TD1-crypto.pdf