Correction du contrôle continu

Ex1. Questions de cours

1. *Nous avons une image de 3870 pixels par 7600 pixels. Nous voulons cacher une donnée sur le bit de poids faible de chaque composante de couleur (en partant du principe qu'on est en RGB sans canal alpha). Pour rappel, nous avons besoin de trois octets qui coderont la taille de la donnée à cacher. Quelle taille maximale peut avoir cette donnée pour que la stéganographie soit possible selon la méthodologie vue en TP ?*

Réponse Dans un premier temps, calculons $\mathrm{NBOctets}$ le nombre d'octets disponibles en prenant 1 bit par composante de couleur par pixel. Alors, $\mathrm{NBOctets} = 3870 * 7600 * 3/8 = 11029500$. Comme dans la vidéo associée au TP 0, les trois premiers octets permettent de connaître le nombre d'octets cachés pour une donnée. Si $2^{24} - 1 > \mathrm{NBOctets} - 3$ alors on peut cacher n'importe quelle donnée de poids inférieur ou égale à $\mathrm{NBOctets} - 3$. Si ce n'était pas le cas, on serait alors limité par la taille encodable sur 3 octets i.e. $2^{24} - 1$. En résumé, $2^{24} - 1 = 16777215 > 11029500 - 3$, donc toute donnée dont la taille est inférieure à 11029497 octets peut être cachée dans cette image.

2. *D'ailleurs, en quoi est-ce utile d'enregistrer la taille de la donnée cachée ?*

Réponse Cela permet de savoir exactement combien d'octets il faut extraire de l'image afin de reconstruire la donnée originale.

3. *Soit o une variable stockant un octet (représenté sous la forme d'un entier entre 0 et 255). Soit d la donnée de 2 bits « 10 ». Donnez, en Python, les lignes de code qui permettront de remplacer les deux bits de poids faible de o par la donnée d.*

Réponse Rien ne vaut quelques lignes de Python.

```
d= 0b10
o = (o & 252)^d
```

4. *En quoi l'indice de coïncidence est-il utile dans la cryptanalyse de Vigenère ?*

Réponse L'idée en général est de maximiser l'indice de coïncidence sur des sous-textes extraits à partir du texte original (en utilisant l'algo de répartition des lettres pour une longueur de clé supposée). Le maximum obtenu signifie alors que nous avons une substitution mono-alphabétique dans chaque sous-texte extrait pour la longueur de clé étudiée. La taille est donc probablement la bonne. Reste à trouver chaque lettre de la clé.

Ex2. Double-CTR

1. *Donnez précisément les schémas du CTR en chiffrement et en déchiffrement.*

Réponse Pour éviter un dessin, on peut le décrire simplement comme suit. Posons $M = M_1, \ldots, M_k$ et $C = C_0, \ldots, C_k$ représentant respectivement le message clair en k blocs et le message chiffré résultant en $k+1$ blocs. Soit K la clé utilisée pour AES. Soit n le nombre aléatoire qui nous sert de valeur initiale pour notre compteur qui correspond au bloc C_0. Alors, pour tout $i \geq 1$, $C_i = \mathrm{AES}_K(n - 1 + i) \oplus M_i$. Respectivement, en mode déchiffrement, pour tout $i \geq 1$, nous avons $M_i = \mathrm{AES}_K(n - 1 + i) \oplus C_i$.

2. *Nous avons intercepté l'échange suivant. Visiblement il y a du CTR dans l'air, et on fonctionne avec un mini-AES et des blocs de 16 bits. On sait de source sûre que le premier message de Yoh est : "Mat Mat !". On sait aussi que le second message clair de Yoh et le premier message clair de Mathieu sont identiques à 100 %.*

   (a) *Détaillez précisément la méthodologie qui vous permettra de déchiffrer l'intégralité de cette discussion.*

Réponse Il semble évident que Mat utilise toujours la même valeur initiale du compteur. On constate la même chose du côté de Yoh mais avec un compteur différent. S'ils utilisent toujours la même clé en considérant une clé pour chaque discussion, alors nous n'avons pas besoin de les connaître. Nous allons renommer symboliquement tous les blocs de la conversation en supposant N le compteur de Yoh et N′ le compteur de Mat.

1. Yoh : N C_1 C_2 C_3 C_4
2. Yoh : N C_5 C_6 C_7
3. Mat : N′ C_8 C_9 C_{10}
4. Mat : N′ C_{11} C_{12}
5. Yoh : N C_{13} C_{14}

A chaque bloc C_i, il y a un bloc de texte clair associé M_i. Tous les blocs C_i sont connus de par la discussion donnée. On sait par hypothèse que M_1 = "Ma" = 4d61,M_2 = "t " = 7420, M_3 = "Ma" = 4d61 et M_4 = "t!" = 7421.

Il est à noter aussi que $C_1 = AES_K(N) \oplus M_1$, $C_2 = AES_K(N+1) \oplus M_2$, $C_3 = AES_K(N+2) \oplus M_3$ et $C_4 = AES_K(N+3) \oplus M_4$. Comme on connaît M_1, M_2, M_3, M_4, C_1, C_2, C_3 et C_4, on peut calculer les blocs secrets $AES_K(N)$, $AES_K(N+1)$, $AES_K(N+2)$ et $AES_K(N+3)$ sans connaître K. En effet, $AES_K(N) = M_1 \oplus C_1$, $AES_K(N+1) = M_2 \oplus C_2$, $AES_K(N+2) = M_3 \oplus C_3$ et $AES_K(N+3) = M_4 \oplus C_4$. De la même façon, on sait que $C_5 = AES_K(N) \oplus M_5$, $C_6 = AES_K(N+1) \oplus M_6$ et $C_7 = AES_K(N+2) \oplus M_7$. Alors on peut deviner M_5, M_6 et M_7 en faisant les calculs suivants :

— $M_5 = AES_K(N) \oplus C_5 = M_1 \oplus C_1 \oplus C_5$
— $M_6 = AES_K(N+1) \oplus C_6 = M_2 \oplus C_2 \oplus C_6$
— $M_7 = AES_K(N+2) \oplus C_7 = M_3 \oplus C_3 \oplus C_7$

On peut procéder de manière similaire pour le traitement du premier de message de Mat. En effet, on sait par hypothèse que $M_5 = M_8$, $M_6 = M_9$ et que $M_7 = M_{10}$ (le second message clair de Yoh et le premier message clair de Mat sont identiques à 100%).

On peut alors trouver les blocs secrets de Mat qui correspondent exactement à $AES_{K'}(N')$, $AES_{K'}(N'+1)$, $AES_{K'}(N'+2)$ et qui peuvent être calculés comme suit :

— $AES_{K'}(N') = C_8 \oplus M_8 = C_8 \oplus M_5 = C_8 \oplus AES_K(N) \oplus C_5 = C_8 \oplus M_1 \oplus C_1 \oplus C_5$
— $AES_{K'}(N'+1) = C_9 \oplus M_9 = C_9 \oplus M_6 = C_9 \oplus AES_K(N+1) \oplus C_6 = C_9 \oplus M_2 \oplus C_2 \oplus C_6$
— $AES_{K'}(N'+2) = C_{10} \oplus M_{10} = C_{10} \oplus M_7 = C_{10} \oplus AES_K(N+2) \oplus C_7 = C_{10} \oplus M_3 \oplus C_3 \oplus C_7$

Au final, nous pouvons alors décrypter le message de Mat envoyé en étape 4 en faisant ceci :

— $M_{11} = C_{11} \oplus AES_{K'}(N') = C_{11} \oplus C_8 \oplus M_1 \oplus C_1 \oplus C_5$
— $M_{12} = C_{12} \oplus AES_{K'}(N'+1) = C_{12} \oplus C_9 \oplus M_2 \oplus C_2 \oplus C_6$

Pour le dernier message de Yoh, on connaît ses blocs secrets, donc sans surprise :

— $M_{13} = C_{13} \oplus AES_K(N) = C_{13} \oplus M_1 \oplus C_1$
— $M_{14} = C_{14} \oplus AES_K(N+1) = C_{14} \oplus M_2 \oplus C_2$

   (b) *Déchiffrez le dernier message envoyé par Mat. Tous les calculs doivent apparaître sur votre copie. Aucune réponse magique sortant du chapeau ne sera considérée. Ne soyez pas surpris par une incohérence apparente des messages déchiffrés.*

1. *Yoh :24ae 4d(ça se fait sans problème avec l'indicde de8a 64cd b5f0 bcb1*

2. *Yoh :24ae 31d9 68dc cbae*
3. *Mat :bf24 3e86 f781 dc8a*
4. *Mat :bf24 3294 bb82*
5. *Yoh :24ae 4d8e 628e 91b0*

Réponse Pour déchiffrer le message de Mat à l'étape 4, nous pouvons nous baser sur notre résultat qui est le suivant :

— $M_{11} = C_{11} \oplus C_8 \oplus C_5 \oplus C_1 \oplus M_1 = 3294 \oplus 3e86 \oplus 31d9 \oplus 4d8a \oplus 4d61 = 3d20 =$"= "

— $M_{12} = C_{12} \oplus C_9 \oplus C_6 \oplus C_5 \oplus C_2 \oplus M_2 = bb82 \oplus f781 \oplus 68dc \oplus 64cd \oplus 7420 = 3432 =$"42"

Le message répondu par Mat est donc "= 42".

3. *Normalement, vous avez désormais toutes les billes en main pour envoyer un message à Yoh, en vous faisant passer pour Mat. Avec la configuration CTR de Mat, chiffrez les 4 octets du message suivant : "Top !"*

Réponse Mettons "Top!" en hexadécimal est nous obtenons 546f 7021. Il nous reste plus qu'à appliquer les blocs secrets de Mat i.e. $AES_{K'}(N')$ et $AES_{K'}(N'+1)$. D'après la question précédente, nous avons :

— $AES_{K'}(N') = C_8 \oplus C_5 \oplus C_1 \oplus M_1 = 0fb4$

— $AES_{K'}(N'+1) = C_9 \oplus C_6 \oplus C_2 \oplus M_2 = 8fb0$

Donc je peux envoyer : 546f $\oplus$ 0fb4 7021 $\oplus$ 8fb0 = bf24 5bdb ff91

Ex3. $\oplus$-Vigenère

1. Détaillez la marche à suivre pour déchiffrer le message à partir du tableau des occurrences.

Réponse On va procéder de la même façon que pour Vigenère classique. On cherche la taille de la clé par l'indice de coïncidence. Une fois la taille de clé trouvée, on cherche dans chaque sous-texte décomposé dans le tableau d'occurrences, l'octet qui apparaît le plus souvent. Celui ci sera le résultat du chiffrement de la lettre **E**. Une fois la clé recomposée, il reste à répéter la clé sous le texte chiffré puis de xor une lettre du chiffre avec la lettre de clé correspondant.

2. *Donnez la clé de chiffrement que vous aurez trouvée à partir des éléments donnés.*

Réponse Il y a une façon très simple de résoudre ce problème. En particulier, pour trouver la taille de clé, il n'est pas indispensable ici de calculer l'indice de coïncidence pour chaque sous texte. Il suffit de compter le nombre de symboles dans chaque alphabet chiffré et de comprendre que la taille de l'alphabet est préservée par l'opération $\oplus$.

Soient n la taille de clé et M le message à chiffrer.

Pour tout i et k tels que $i+n*k \leq |M|$, si $M[i] = M[i+n*k]$ alors $C[i] = C[i+n*k]$. En effet, par application de $\oplus$−Vigenère, $C[i] = ord(M[i]) \oplus ord(K[i\%n])$ et $C[i+n*k] = ord(M[i+n*k]) \oplus ord(K[(i+n*k)\%n])$. Comme $M[i] = M[i+n*k]$ et $(i+n*k)\%n = i$, nous avons bien $C[i+n*k] = ord(M[i+n*k]) \oplus ord(K[(i+n*k)\%n]) = ord(M[i]) \oplus ord(K[i]) = C[i]$. Si $M[i]! = M[i+n*k]$ alors $C[i]! = C[i+n*k]$. Donc si on a 26 lettres différentes dans un alphabet dans le texte clair, on aura alors 26 octets différents dans le texte chiffré.

Il est en particulier pas possible d'obtenir des alphabets de plus de 26 octets à partir d'un alphabet de 26 lettres. Ce qui élimine de fait les tailles de clés 1, 2 et 4 qui engendrent des alphabets de tailles strictement supérieures à 26.

Il est tout à fait possible de calculer les indices de coïncidence et vous arriveriez à la même conclusion.

La taille de la clé est de 3.

Donc maintenant il suffit de procéder comme pour Vigenère i.e. trouver l'octet qui a chiffré le **E** dans chaque sous-texte.

3	1	248	20→2 | 21→24 | 22→2 | 23→1 | 24→14 | 25→7 | 26→18 | 27→11 29→23 | 2b→8 | 2c→10 | 2d→37 | 2e→3 | 2f→2 | 30→1 | 38→6 39→8 | 3a→9 | 3b→21 | 3c→20 | 3d→17 | 3e→4
	2	248	20→16 | 21→14 | 22→7 | 23→16 | 25→2 | 26→21 | 27→3 | 28→3 29→5 | 2a→40 | 2b→11 | 2c→7 | 2d→2 | 2e→26 | 37→2 | 39→3 3a→17 | 3b→19 | 3c→8 | 3d→12 | 3e→5 | 3f→9
	3	247	20→7 | 21→4 | 22→17 | 23→17 | 24→14 | 25→22 | 26→8 | 29→1 31→20 | 32→1 | 33→5 | 34→4 | 35→35 | 36→5 | 38→3 | 39→26 3c→15 | 3d→6 | 3e→21 | 3f→16

— Dans le sous-texte 1, l'octet le plus fréquent est **2d**.
— Dans le sous-texte 2, l'octet le plus fréquent est **2a**
— Dans le sous-texte 3, l'octet le plus fréquent est **35**.

Comme en CM, TD et TP, il suffit de *soustraire* **E** à l'octet. On parle de soustraction dans Vigenère car la transformation lors du chiffrement est l'addition. La soustraction compense l'addition. Ici, l'opération qui compense le $\oplus$ est lui même. En effet, $x \oplus x = 0$.

La clé est donc $2d \oplus 45 = 68$, $2a \oplus 45 = 6f$ et $35 \oplus 45 = 70$ i.e. hop.

3. *Déchiffrez les 4 derniers octets du message chiffré.*

Réponse Assez facilement on repère qu'il y a 18 octets par lignes. 18 étant un multiple de 3, à chaque nouvelle ligne, on redémarre sur la première lettre de la clé.

Donc $3c \oplus 68$, $28 \oplus 6f$, $3f \oplus 70$, $23 \oplus 68$ et $3a \oplus 6f$.

On obtient alors $28 \oplus 6f = 47 = G$, $3f \oplus 70 = 4f = O$, $23 \oplus 68 = 4b = K$ et $3a \oplus 6f = 55 = U$ i.e. **GOKU**.