DM 2

Vous enverrez votre fichier, sous la forme décrite à l'exercice 2, pour le 10/01/2016 à : vojpetrov@gmail.com (PT, PSI*), resp. gbarat@free.fr (PT)*

Ce devoir traite de deux techniques différentes de codage de textes.

La cryptographie consiste à envoyer un message codé. Ce message peut être transmis tel quel. Un lecteur qui ne disposerait pas du protocole de codage/décodage ne pourrait en théorie comprendre son sens.

La stéganographie consiste d'un autre côté à cacher un message dans un autre. Le message, qui pourrait de surcroît être crypté, mais ne l'est pas forcément, est destiné à ne pas être détecté par quelqu'un qui ignorerait son existence.

Si la législation en vigueur précise quels sont les protocoles cryptographiques qui peuvent être employés lors de transmissions par internet, mais qui ne sont pourtant pas forcément décryptables lors d'enquêtes, elle a en revanche peu de sympathie avec la stéganographie, qui paraît pourtant bien plus innocente !

Exercice 1. Chiffrage de Vigenère

Le chiffrage de Vigenère est une méthode de cryptographie utilisée, au moins dans une de ses variantes, dès le XVI$^{\text{ème}}$ siècle.

C'est un système de chiffrement polyalphabétique par substitution, mais une même lettre du message clair peut, suivant sa position dans celui-ci, être remplacée par des lettres différentes, contrairement à un système de chiffrement monoalphabétique comme le chiffre de César (qu'il utilise cependant comme composant).

Le chiffre de César, est une méthode de chiffrement très simple utilisée par Jules César dans ses correspondances secrètes (ce qui explique le nom « chiffre de César »).

Le texte chiffré s'obtient en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet. Pour les dernières lettres (dans le cas d'un décalage à droite), on reprend au début. Par exemple avec un décalage de 3 vers la droite, A est remplacé par D, B devient E, et ainsi jusqu'à W qui devient Z, puis X devient A, Y, B et Z, C. Il s'agit d'une permutation circulaire de l'alphabet. La longueur du décalage, 3 dans l'exemple évoqué, constitue la clé du chiffrement qu'il suffit de transmettre au destinataire — s'il sait déjà qu'il s'agit d'un chiffrement de César — pour que celui-ci puisse déchiffrer le message. Dans le cas de l'alphabet latin, le chiffre de César n'a que 26 clés possibles (y compris la clé nulle, qui ne modifie pas le texte).

Ce chiffrement introduit la notion de clé. Dans le chiffrage de Vigenère, une clé se présente généralement sous la forme d'un mot ou d'une phrase. Pour pouvoir chiffrer notre texte, à chaque caractère nous utilisons une lettre de la clé pour effectuer la substitution. Évidemment, plus la clé sera longue et variée et mieux le texte sera chiffré.

Dans la clé, le caractère en position i va déterminer le décalage à effectuer dans le texte à coder à la position i. Si le caractère dans la clé est un 'A' on effectuera un décalage de 0. Si le caractère est un 'B' on effectuera un décalage de +1. Si c'est un 'C' un décalage de +2 ... si c'est un 'Z' un décalage de +25.

Ainsi, 'KEBAB' avec la clé 'ADANA' se code 'KHBNB' (les 'A' dans "ADANA" n'ont pas d'effet de décalage ; le 'D' a pour effet de décaler de +3 donc transforme le 'E' en 'H' ; le 'N' a un effet +13, donc transforme un 'A' en 'N').

Lorsque la clé utilisée est plus courte que le texte, elle est répétée autant de fois que nécessaire. Ainsi 'KEBABBIENCUIT' avec la clé 'ADANA' est codé comme si la clé était 'ADANAADANAADA' et est codé 'KHBNBBLEACULT'.

Dans cet exercice, on utilisera des textes encodés en majuscules, sans ponctuation et sans accents. On n'écrira par ailleurs pas les espaces dans les textes codés et décodés. Ainsi « Je programme tous les jours pour m'améliorer ! » sera écrit sous la forme « JEPROGRAMMETOUSLESJOURSPOURMAMELIORER ».

1. Écrire une fonction CodageVigenere(texte,cle) de codage du texte passé comme argument `texte` en utilisant la clé passée comme argument `cle`.

2. Écrire une fonction DecodageVigenere(tcrypte,cle) qui décode un texte crypté `tcrypte` qui a été codé selon la clé `cle`.

Exercice 2. Stéganographie dans une image en niveaux de gris par réécriture du LSB

Dans cet exercice, on travaille avec des images en niveaux de gris, de préférence en format `.png`.

Si vous manquez d'imagination, vous pouvez travailler avec le fichier `lena_NB.png` que vous pourrez par exemple récupérer sur www.infojeanperrin.fr/PTE.html. Vous pouvez évidemment travailler avec n'importe quel autre image et laisser libre cours à votre originalité.

La stéganographie par réécriture du LSB (*least significant bit*) consiste à cacher un texte dans une image (en niveaux de gris, donc une matrice de nombres entiers compris entre 0 et 255) de la manière suivante :

- Pour chaque pixel de l'image, le bit le moins significatif (donc correspondant à 2^0 dans l'écriture binaire du nombre) ne sera pas en réalité associé à une différence de niveau de gris dans l'image mais à un bit dans l'encodage du message caché ;
- les bits extraits de l'image (les moins significatifs, donc) seront regroupés 8 par 8 ; on lit la matrice ligne par ligne ;
- chaque groupement de bits sera converti en un entier non signé (compris entre 0 et 255) ;
- cet entier sera converti en un caractère (encodage ASCII) à l'aide de la commande `chr` de Python. Par exemple chr(97) est évalué à `a` ;
- on admettra par convention que le texte s'arrête lorsque l'entier lu est 128. Celui-ci correspond au caractère "\x80" qui est un caractère conventionnel en ASCII.

Par exemple, supposons que la première ligne de l'image commence par 24 13 8 24 24 24 255 0
Les bits extraits sont 0100 0010 ; ceci correspond à 66 ; et, comme `chr(66)` est évalué à `B`, c'est ce caractère qui est caché dans ces pixels de l'image.
Notons que ces pixels auraient pu au départ être les mêmes ; ou 25 12 8 25 24 25 254 1 ou encore 24 13 9 25 25 24 255 0, etc...

1. Écrire une fonction `stegano(fichier, chaineaencoder)` qui prend en argument une chaîne de caractère `fichier` correspondant à un fichier image, et une chaîne de caractères `chaineaencoder` qui représente le texte que l'on souhaite cacher dans l'image. Cette fonction créera une image selon le protocole décrit ci-dessus, et l'enregistrera dans le fichier `"code.png"`.

2. Écrire une fonction `unstegano(fichier)` qui prend en argument une chaîne de caractère `fichier` correspondant à un fichier image dans lequel est caché un message par le procédé décrit ci-dessus et qui renvoie la chaîne de caractères cachée dans l'image.

3. Faites un copier-coller de votre DM dans une chaîne de caractères (entre guillemets simples si vous avez utilisé des guillemets doubles pour faire votre programme). Codez alors dans une image en niveaux de gris de votre choix votre code source. Et envoyez le résultat, qui correspond à votre copie, par courriel.