Université Paris Diderot
L3 informatique
Sécurité 2014-2015

TP Chiffrement par substitution

Les textes considérés sont composés des lettres romaines minuscules (`'abcdefghijklmnopqrstuvwxyz'`) et de l'espace (`' '`). Cependant pour simplifier le décryptage on supposera que l'espace n'est pas chiffré.

Dans le chiffre de César, le texte chiffré s'obtient en remplaçant chaque lettre du texte en clair par une lettre à distance fixe dans l'ordre de l'alphabet. Par exemple si cette distance est 3, `'abd'` sera chiffré en `'deg'`.

Dans le chiffrage par permutation, étant donnée une permutation des lettres σ, chaque lettre x est chiffrée par $\sigma(x)$.

Le chiffre de Vigenère réalise un décalage différent suivant la position de la lettre dans le texte. Le décalage est définie à l'aide d'un mot-clef. Dans ce mot-clef, chaque lettre correspond à un décalage. Ainsi si le mot-clef est `bal` (correspondant au décalage de 1, de 0 et de 11) pour le texte à chiffrer "`bonjour`", le `'b'` sera décalé de 1, le `'o'` ne sera pas décalé, `'n'` sera décalé de 11, `'j'` sera décalé de 1, et ainsi de suite en réutilisant cycliquement le mot-clef. On obtient ainsi "`coykofs`".

1 Pour commencer

Dans cette partie on devra écrire des classes java permettant de faire le chiffrage et le déchiffrage pour ces chiffres alphabétiques. Pour cela:

- On commencera par définir une interface avec des méthodes chiffrer et dechiffrer pour le chiffrage et déchiffrage (pour un caractère, pour une String et pour tout autre type de données souhaitable).
- Ensuite on créera des classes ad hoc qui implémentent cette interface pour les trois types de chiffres décrits ci-dessus.

Pour pouvoir tester on définira deux applications chiffre et dechiffre, une pour le chiffrage la deuxième pour le déchiffrage, ayant comme premier paramètre le type de chiffrage (c/p/v pour César, permutation et Vigenère), comme deuxième paramètre la "clé" du chiffrage (valeur du décalage pour César, la permutation pour un chiffre par permutation, le mot-clef pour Vigenère), comme troisième un fichier contenant le texte à chiffrer ou à déchiffrer et qui affichent sur la sortie standard le texte chiffré ou déchiffré suivant les cas.

2 Décryptage

Le but de cette partie est d'essayer de déchiffrer ces différents chiffres sans la clef. Pour cela on suppose que l'on dispose d'un texte chiffré qui était en français (suffisamment long) dont on connait la méthode de chiffrement mais dont on ne connait pas la clé (décalage pur Cesar, la permutation pour le chiffre par permutation ou le mot-clef si c'est le chiffre de Vigenere (mais dans ce cas on suppose la longueur du mot-clef connue)).

On peut disposer d'une liste des mots du français (accessible sous didel) ainsi que des fréquences des lettres en français (par exemple `wikipedia.org/wiki/Fréquence_d'apparition_des_lettres_en_français`). (Comme on ne considère ici que les lettres minuscules sans accent et l'espace (qui rappelons-le n'est pas chiffré et apparait comme tel dans le texte chiffré), il faudra en préliminaire éliminer les accents et autres signes de façon à ne garder que les lettres alphabétiques minuscules et l'espace).

1. Du fait du nombre limité de ses clefs le chiffre de César est facile à décrypter. On définira et implémentera trois méthodes:

(a) La première basée sur la connaissance d'un mot: on sait que le mot est dans le texte en clair (mais on ne sait pas à quelle position). En cas d'ambiguité on pourra utiliser la liste des mots du français pour lever cette ambiguité.

(b) La deuxième est basée sur les fréquences: à partir de l'analyse des fréquences des lettres on déterminera le décalage.

(c) Par force brute on essaiera tous les décalages jusqu'à obtenir un texte dont tous les mots sont dans le dictionnaire.

1. Pour le chiffrage de Vigenère, on suppose que l'on connaît la taille du mot-clef, définir des décryptages qui pourront utiliser les classes définies dans la question précédente.

   Pour ceux qui veulent plus: Que peut-on faire si on ne connaît pas la taille de la clef? Ecrire une classe permettant le décryptage dans ce cas.

2. Pour le codage par permutation, écrire une classe qui réalise un décryptage partiel basé sur les fréquences des lettres.

   Pour ceux qui veulent plus, réaliser un décryptage complet basé sur les fréquences des lettres et la liste de mots du français.

Pour pouvoir tester on définira une application `decrypt` ayant comme premier paramètre le type de codage, comme deuxième paramètre un fichier contenant le texte à décrypter et qui affiche sur la sortie standard le texte décrypté. Dans le cas d'un décryptage de Cesar, un troisième paramètre donnera la stratégie (`1/2/3`) et pour la stratégie 1 un quatrième paramètre donnera le mot qui appartient au texte en clair. Dans le cas d'un décryptage de Vigenère un troisième paramètre donnera la taille du mot-clef.

Vous afficherez sur la sortie erreur une évaluation du temps d'exécution.

```
long startTime = System.currentTimeMillis();
      //          votre méthode
long endTime = System.currentTimeMillis();
System.err.println("Temps de xxxxx:" + (endTime-startTime) + "ms");
```

3 Travail à rendre

Vous pouvez réaliser ce travail par groupe d'au plus 3 personnes. Vous devez rendre sous didel **un travail par groupe**: pour le 13 mars la section 1 et pour le 20 mars la section 2. Ainsi qu'un court rapport décrivant vos choix d'algorithme et de programmation et indiquant **précisément** comment exécuter votre application `decrypt`.

Tous vos codes java seront dans le même package `securiteL3`. Un test automatique à l'aide de fichiers script shell sera fait. Faites en sorte qu'il suffise d'exécuter ces fichiers shell dans le même catalogue que celui où se trouve le catalogue `securiteL3` contenant les fichiers `.class`. Typiquement le code d'un de ces fichiers, qui s'exécute avec un fichier en argument, sera:

```
java securiteL3/chiffre p "azertyuiopqsdfghjklmwxcvbn"  $1> $1.chiffre
java securiteL3/dechiffre  p "azertyuiopqsdfghjklmwxcvbn"  $1.chiffre > $1.clair
if diff $1 $1.clair
then echo chiffre réussi
else echo chiffre rate
fi
java securiteL3/chiffre c 4  $1> $1.chiffre
java securiteL3/decrypt  c $1.chiffre  1 licence  >$1.clair
if diff $1 $1.clair
then echo decrypt réussi
else echo decrypt rate
fi
```