Travaux pratiques

informatique commune

Cryptographie à clef secrète

Le principe de la cryptographie à clef secrète est probablement le principe le plus naturel auquel on pense lorsqu'on envisage de cacher une information : l'émetteur et le récepteur partagent un secret commun qui permet de chiffrer et de déchiffrer un texte. Les opérations pour le codage et pour le décodage sont alors essentiellement les mêmes, d'où le qualificatif « symétrique » pour de tels cryptosystèmes.
Dans la première partie de ce TP nous allons nous intéresser à un procédé de chiffrement à clef secrète apparu au XVI^{e} siècle : le chiffrement de Vigenère. Nous verrons qu'avec les moyens de calcul actuels ce type de cryptosystème est facile à casser. Dans un second temps nous nous intéresserons à un problème induit par la cryptographie à clef secrète : comment partager une clef entre deux personnes à travers un canal non sécurisé ? Nous étudierons la méthode de Diffie-Hellman [1] et l'attaque de Shanks de cette méthode, à travers l'utilisation des courbes elliptiques.

1. Chiffrement de Vigenère

Dans un but de simplification, nous ne considérerons que des messages non accentués écrits en lettres majuscules, sans espace ni ponctuation. Par exemple, le message à coder « *Les sanglots longs des violons de l'automne* » sera transmis sous la forme : `LESSANGLOTSLONGSDESVIOLONSDELAUTOMNE`.
Commencez donc par définir l'alphabet :

```
alph = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Cet alphabet est identifié à $\mathbb{Z}/26\mathbb{Z}$; ainsi à la lettre `A` correspond la valeur 0, à la lettre `B` la valeur 1, etc. Cette identification induit en outre une opération d'addition (ou de décalage si on préfère) sur les lettres : par exemple, `D + K = N` puisque $(3+10) \bmod 26 = 13$, ou encore `J + T = C` puisque $(9+19) \bmod 26 = 3$.

La clef secrète de la méthode de Vigenère est un mot c de longueur ℓ sur cet alphabet. Le message à coder est découpé en blocs b de longueur ℓ et chaque lettre de chacun de ces blocs est décalée de la valeur associée à la lettre de même rang dans la clef secrète.
Par exemple, le chiffrement du message ci-dessus à partir de la clef secrète `VERLAINE` se réalise ainsi :

	L	E	S	S	A	N	G	L	O	T	S	L	O	N	G	S	D	E	S	V	I	O	L	O	N	S	D	E	L	A	U	T	O	M	N	E
+	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L	A	I	N	E	V	E	R	L
=	G	I	J	D	A	V	T	P	J	X	J	W	O	V	T	W	Y	I	J	G	I	W	Y	S	I	W	U	P	L	I	H	X	J	Q	E	P

Le message chiffré est donc : `GIJDAVTPJXJWOVTWYIJGIWYSIWUPLIHXJQEP`.

Question 1.

a) Rédiger deux fonctions `chiffre(clef, message)` et `dechiffre(clef, message)` réalisant respectivement le chiffrement et le déchiffrement d'un message à partir d'une clef secrète en utilisant la méthode de Vigenère.

b) Dans le fichier `vigenere.txt` que vous pouvez récupérer dans le dossier `http://info-llg.fr/commun-mp/textes/` se trouvent plusieurs textes qui ont été chiffrés à l'aide de la méthode de Vigenère. Déchiffrer le premier sachant qu'il a été codé à l'aide de la clef `VERLAINE`.

Attaque du chiffrement de Vigenere

Nous allons maintenant nous intéresser aux méthodes qui permettent de « casser » un texte chiffré par la méthode de Vigenère, c'est-à-dire qui permettent de reconstituer le texte initial sans en posséder la clef.
Dans un premier temps, nous allons supposer connue la longueur ℓ de la clef. On sait alors que pour tout $k \in [\![0, \ell-1]\!]$, toutes les lettres dont les indices sont congrus à k modulo ℓ sont décalées d'une même valeur $d_k \in [\![0, 25]\!]$. Notons M_k l'ensemble de ces lettres. Si le message est suffisamment long il y a de fortes chances pour que la valeur de d_k soit celle pour laquelle la fréquence d'apparition des lettres de l'ensemble $\left\{x - d_k \;\middle|\; x \in \mathrm{M}_k\right\}$ soit la plus proche possible de la fréquence d'apparition de ces mêmes lettres dans la langue française.

1. Récents lauréats (2016) du prix Turing pour leurs travaux en cryptographie.

Nous avons donc besoin des fréquences d'apparition des différentes lettres de la langue française.
Le fichier `http://info-llg.fr/commun-mp/textes/hugo.txt` contient la reproduction du roman « Quatrevingt-Treize » de Victor Hugo. Outre les 26 lettres de l'alphabet ce texte comporte des espaces (`' '`) et des passages à la lignes (`'\n'`).

Question 2. Utiliser ce document pour créer un tableau `francais` de 26 cases contenant la fréquence d'apparition de chacune des 26 lettres de l'alphabet dans ce roman (consulter la figure 1 pour savoir comment travailler sur le contenu d'un fichier texte en Python).

Pour accéder au contenu d'un fichier `exemple.txt` il faut commencer par l'ouvrir en mode lecture :

```
f = open('exemple.txt', 'r')
```

On crée ainsi un objet que l'on peut énumérer ligne par ligne :

```
for l in f:
    ...
```

donne à `l` la valeur de chacune des lignes de ce document (y compris le caractère `'\n'` en fin de ligne).
On met fin à l'ouverture de ce fichier à l'aide de la méthode `close` :

```
f.close()
```

Figure 1 – Lecture d'un fichier texte.

Nous considérerons désormais que ce tableau correspond à la fréquence d'apparition des lettres dans la langue française.

Question 3. Rédiger une fonction `frequence(texte)` qui prend en argument une chaîne de caractère et retourne un tableau de 26 cases contenant la fréquence d'apparition de chacune des 26 lettres de l'alphabet dans ce texte.

Les tableaux renvoyés par la fonction `frequence` sont destinés à être comparés au tableau `francais` calculé à la question 2. Pour cela, nous allons calculer le *coefficient de corrélation* qui lie ces deux tableaux. Ce dernier se définit de la manière suivante : si $X = [x_0, x_1, \dots, x_{n-1}]$ et $Y = [y_0, y_1, \dots, y_{n-1}]$ sont deux tableaux de valeurs de même longueur alors

$$\text{cor}(X, Y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2}\sqrt{\sum_i (y_i - \overline{y})^2}}$$

où $\overline{x}$ et $\overline{y}$ désignent respectivement les moyennes des valeurs contenues dans les tableaux X et Y.
Cette quantité est comprise entre -1 et 1, et plus $\text{cor}(X, Y)$ est proche de 1, plus on peut considérer que les deux tableaux X et Y sont corrélés.

Question 4. Rédiger une fonction `correlation(X, Y)` qui calcule le coefficient de corrélation de deux listes de valeurs X et Y de même taille. On s'efforcera de réaliser ce calcul en n'effectuant qu'un seul parcours en parallèle de ces deux tableaux (pour cela, on développera les sommes présentes dans cette formule).

On considère maintenant un message M chiffré par la méthode de Vigenère à l'aide d'une clé de longueur ℓ. Ce message est découpé en ℓ messages $M_0, \dots, M_{\ell-1}$, chacun de ces messages ayant été produit par le même décalage d_k à partir de message original. Pour chaque valeur de k on calcule la corrélation maximale existant entre les fréquences de références et le message M_k décalé de $d \in [\![0, 25]\!]$. Cette valeur maximale indique la valeur du décalage associé à la $(k+1)^{\text{e}}$ lettre de la clef secrète.

Question 5.

a) Rédiger une fonction `clefL(message, l)` qui prend en arguments un message crypté à l'aide d'une clé de longueur ℓ et qui renvoie la clef utilisée pour chiffrer ce message, en suivant la démarche décrite ci-dessus.

b) Déchiffrer le second message du fichier `vigenere.txt` sachant qu'il a été chiffré à l'aide d'une clé de longueur 6.

Enfin, lorsqu'on ne connait pas la longueur ℓ de la clé, on fait l'hypothèse que celle-ci est comprise entre par exemple 4 et 20. Pour chacune de ces valeurs on découpe le message chiffré M et ℓ messages M_k et on calcule la corrélation maximale c_k entre les fréquences de référence et le message M_k décalé de i pour $0 \leqslant i \leqslant 25$. On note d_k le décalage correspondant à cette corrélation maximale.
On calcule ensuite la moyenne m_ell des valeurs de c_k pour $0 \leqslant k < \ell$. L'entier $\ell \in [\![4, 20]\!]$ pour lequel m_ℓ est maximal est la longueur de la clé, et celle-ci est égale au mot $d_0 d_1 \cdots d_{k-1}$.

Question 6.

a) Rédiger une fonction `clef(message)` qui prend en argument un message crypté pour lequel la clé a une longueur comprise entre 4 et 20 et qui renvoie la clef utilisée pour chiffrer ce message.

b) Déchiffrer enfin le troisième message du fichier `vigenere.txt`.

2. Échange de clefs Diffie-Hellman

Comme nous venons de le constater, les méthodes de chiffrement à base de substitution alphabétique ne constituent pas des méthodes de chiffrement fiables. Mais il existe des systèmes modernes de chiffrement à clef secrète tel AES (pour *advanced Encryption Standard*) développé pour la NSA [2] qui sont reconnus pour leur résistance aux attaques.
Cependant, un problème continue de se poser : comment transmettre la clef secrète aux deux utilisateurs ? Ce problème est souvent modélisé en faisant intervenir deux personnes nommées conventionnellement Alice et Bob, qui doivent se mettre d'accord sur un nombre (qui servira à chiffrer et déchiffrer leurs messages) sans qu'un troisième personnage ayant écouté tous leurs échanges puisse découvrir ce nombre.

En 1976, Diffie et Hellman ont proposé une méthode qui répond au problème. Le protocole de Diffie et Hellman utilise un groupe cyclique G (noté additivement) et repose sur l'idée suivante :

- étant donné un nombre entier k et un élément P dans le groupe G, il est facile de calculer kP ;
- il est en revanche difficile de retrouver k connaissant P et Q = kP.

Le fonctionnement du protocole est le suivant : Le groupe G et l'élément P sont publics ; Alice choisit secrètement un nombre a, calcule aP et l'envoie à Bob. Ce dernier choisit à son tour un nombre secret b et envoie à Alice l'élément bP. Alice peut alors calculer K = a(bP) et Bob calculer b(aP) et obtenir la même clef K (ici un point de G) qu'Alice.

Courbes elliptiques

Ce protocole d'échange est souvent utilisé avec un groupe construit sur une courbe elliptique. À partir d'un corps $\mathbb{K}$ de caractéristique différente de 2 et 3 on considère l'ensemble des points $(x, y) \in \mathbb{K}^2$ qui vérifient une équation de la forme : $y^2 = x^3 + ax + b$ où a et b sont des éléments de $\mathbb{K}$ qui vérifient : $4a^3 + 27b^2 \neq 0$. À l'ensemble des ces points est adjoint un « point à l'infini » O qui jouera le rôle de l'élément neutre pour la loi d'addition sur cette courbe. Ainsi,

$$\mathrm{G} = \left\{(x, y) \in \mathbb{K}^2 \;\middle|\; y^2 = x^3 + ax + b\right\} \cup \{\mathrm{O}\}.$$

Pour définir l'addition de deux points $\mathrm{P} = (x_\mathrm{P}, y_\mathrm{P})$ et $\mathrm{Q} = (x_\mathrm{Q}, y_\mathrm{Q})$ on distingue quatre cas.

- Si P ≠ Q et $x_\mathrm{P} \neq x_\mathrm{Q}$, la droite (PQ) d'équation $y = rx + s$ recoupe G en un troisième point. La somme des points P et Q est alors donnée par le symétrique R de ce point par rapport à l'axe des abscisses.
  Pour calculer les coordonnées de ce point R on calcule tout d'abord :
  $$r = (y_\mathrm{P} - y_\mathrm{Q})(x_\mathrm{P} - x_\mathrm{Q})^{-1} \qquad \text{et} \qquad s = (y_\mathrm{Q} x_\mathrm{P} - y_\mathrm{P} x_\mathrm{Q})(x_\mathrm{P} - x_\mathrm{Q})^{-1}$$
  puis on utilise les formules : $\begin{cases} x_\mathrm{R} = r^2 - x_\mathrm{P} - x_\mathrm{Q} \\ y_\mathrm{R} = -r x_\mathrm{R} - s \end{cases}$
- Si P ≠ Q et $x_\mathrm{P} = x_\mathrm{Q}$ la droite (PQ) est verticale et on pose P + Q = O.
- Si P = Q et $y_\mathrm{P} \neq 0$ on remplace la droite (PQ) par la tangente en P d'équation $y = rx + s$ avec cette fois :
  $$r = (3x_\mathrm{P}^2 + a)(2y_\mathrm{P})^{-1} \qquad \text{et} \qquad s = y_\mathrm{P} - r x_\mathrm{P}$$
  et on calcule x_R et y_R comme dans le premier cas.
- Enfin, si P = Q et $y_\mathrm{P} = 0$ la tangente à la courbe en P est verticale et on pose P + Q = O.

Nous admettrons cette loi confère à G une structure de groupe additif.

2. *National Security Agency*, agence gouvernementale américaine en charge de la sécurité des systèmes d'information.

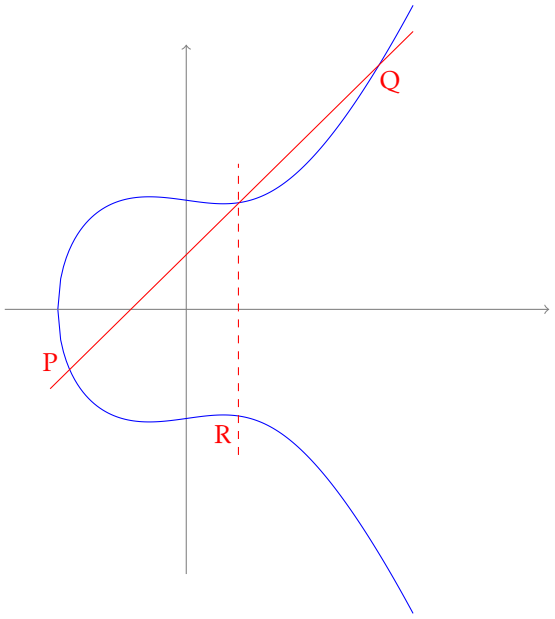

FIGURE 2 – Calcul de R = P + Q.

Question 7.

a) Rédiger une fonction récursive `pgcd(a, b)` qui calcule le pgcd de deux entiers positifs en appliquant l'algorithme d'EUCLIDE.

b) Modifier ensuite cette fonction pour obtenir une fonction récursive `bezout(a, b)` qui outre le pgcd de a et de b renvoie deux entiers u et v vérifiant $ua + vb = \text{pgcd}(a, b)$.

c) En déduire une fonction `inverse(x, p)` qui calcule l'inverse dans $\mathbb{Z}/p\mathbb{Z}$ de l'entier x si ce dernier est inversible.

Nous allons désormais considérer le corps $\mathbb{K} = \mathbb{Z}/p\mathbb{Z}$ avec $p = 17252297107$ (qui est un nombre premier) et la courbe elliptique G d'équation $y^2 = x^3 + 5$.

Question 8. Rédiger une fonction `addition(P, Q)` qui calcule la somme de deux points P et Q de G. On représentera le point O par le couple `(inf, inf)` avec `inf = float('inf')`.
Vous pourrez vérifier la validité de votre fonction à l'aide des exemples ci-dessous :

$$\text{P} = (6587596005, 10930896470), \qquad \text{Q} = (2846256190, 0), \qquad \text{R} = (1099048983, 3110000776)$$

$$\begin{array}{ll} \text{P} + \text{Q} = (10674847433, 12569638509) & \text{Q} + \text{Q} = \text{O} \\ \text{P} + \text{R} = (6587596005, 6321400637) & 3\text{P} = (8376961733, 938225291) \\ \text{P} + \text{P} = (1099048983, 14142296331) & 5\text{P} = (1708109053, 5741342158) \end{array}$$

Question 9. Rédiger une fonction récursive `mult(k, P)` qui calcule le point kP. On utilisera l'algorithme de multiplication rapide basé sur les relations $(2k)\text{P} = k(2\text{P})$ et $(2k+1)\text{P} = \text{P} + k(2\text{P})$.

Question 10.

a) Montrer que si $p = 4m + 3$ et si $x \in \mathbb{K}$ possède une racine carrée dans $\mathbb{K}$ alors $y = x^{m+1}$ est l'une de ces racines.

b) Rédiger une fonction récursive qui calcule x^n dans $\mathbb{K}$ en appliquant le principe de l'exponentiation rapide, autrement dit en utilisant les formules : $x^{2k} = (x^2)^k$ et $x^{2k+1} = x(x^2)^k$.

c) En déduire une fonction `genere_point()` qui calcule un point au hasard pris dans G.

Question 11. Choisir un nombre secret a (un entier à huit chiffres) et s'en servir pour générer avec l'un de vos voisins une clef privée K à partir du point P, comme expliqué dans l'introduction de cette section.

L'attaque de Shanks

La méthode de Shanks (appelée encore attaque *pas de bébé, pas de géant*) permet lorsque l'ordre de G n'est pas trop important de résoudre le problème du logarithme discret, ici présenté en notation additive : si $Q = k\mathrm{P}$ trouver k connaissant P et Q.

Il faut tout d'abord connaitre l'ordre du groupe G. Il existe des algorithmes efficaces permettant de le calculer dans le cas d'une courbe elliptique. Nous admettrons que l'ordre du groupe G avec lequel nous travaillons vaut $r = 4313008603$.
En notant $m = \lceil \sqrt{r} \rceil$, on peut écrire tout entier $x \leqslant r$ sous la forme $x = x_1 + x_2 m$ avec $x_1 < m$ et $x_2 \leqslant m$. Par conséquent l'égalité $Q = x\mathrm{P}$ peut s'écrire $Q - x2(m\mathrm{P}) = x_1\mathrm{P}$.
Dans un premier temps (le pas de bébé) on construit l'ensemble $L = \{O, P, 2P, \ldots, (m-1)P\}$.
Dans un deuxième temps (le pas de géant) on calcule $S = (-m)\mathrm{P}$ puis on recherche un entier j tel que $Q + jS \in L$: si $Q + jS = i\mathrm{P}$ alors $Q = (i + mj)\mathrm{P}$ et $k \equiv i + mj \mod r$.

Question 12. Utiliser l'attaque de Shanks pour retrouver le secret de votre voisin.