Chiffrement

Le point sur le chiffrement

Le principe consiste à utiliser une bijection de $\mathbb{Z}/p\mathbb{Z}$ dans lui-même pour crypter le message. Le codage de Jules César peut être vu comme l'addition d'une constante dans $\mathbb{Z}/26\mathbb{Z}$, Vigenère est l'addition des lettres du message à crypter avec les lettres d'un texte fixé. Le codage affine utilise une application affine $x \to ax + b$
avec a inversible modulo p (le calcul de l'inverse fait intervenir l'identité de Bézout). Aucune de ces méthodes n'est résistante à une analyse statistique du message crypté (dans le cas de Vigenére, l'attaque est plus complexe à mettre en oeuvre).
Le chiffrement de Hill groupe les lettres du message à crypter par paquets de n (n fixé) pour éviter l'attaque par analyse statistique, on a donc un vecteur $v \in (\mathbb{Z}/p\mathbb{Z})^n$ dont on calcule l'image par un chiffrement affine $v \to Av + b$
où A est une matrice inversible sur $\mathbb{Z}/p\mathbb{Z}$ (donc est à la limite du programme sauf dans des cas comme n=2 où on peut exprimer l'inverse explicitement de manière simple, malheureusement dans ce cas n n'est pas suffisamment grand pour que ce code soit résistant à une analyse statistique).
De plus toutes ces méthodes supposent que les clés de chiffrement et de déchiffrement sont secretes (en effet si on connait l'une des clefs on en déduit l'autre), alors que RSA par exemple permet de publier une des deux clefs.
Exercices : écrire des fonctions de chiffrement/déchiffrement par ces méthodes. Instruction en Xcas : asc, char,

Chiffrement

1 Travaux pratiques

Les fichiers affine.py et decalage.py sont des programmes en python qui vous permettront toute sorte d'analyse en vu de décoder les textes fournis.
Attention, toutes les chaines de caractères doivent être tapée entre guillemet ! ! !

1.1 Prise en main rapide

Ouvrez le fichier `Decalage.py` avec le programme `IDLE`. Jetez un oeil aux procèdures.
En fin de programme taper par exemple la commande

```
l=Decale([12 , 24],4)
print(l)
```

Faire Ctrl-S pour enregistrer et F5 pour faire tourner le programme. Comprenez-vous le résultat ? Si oui, il est grand temps de passer à la suite...

1.2 Code cesar

c'est un decalage du type $x \to x + b$. A chaque lettre on associe un entier, on y rajoute la valeur b, on regarde le reste modulo 26 et on repasse en lettre. On donne le texte suivant.

Texte 1

AP VGTCDJXAAT P VGPCST QDJRWT VDQT STH BDJRWTH PKTR HP VGPCST QDJRWT TAAT KXI SPCH JCT BPGT HJG JC CTCJEWPG FJX AJX HTGI ST EADCVTDXG BPXH KDXAP FJ JC HDXG TAAT TC P BPGGT STH BDJRWTH PJ ETIXI STYTJCTG STH BDJRWTH PJ SXCTG STH BDJRWTH IDJIT AP YDJGCTT TAAT TC P PHHTO

1. Que veut-il dire ? (servez-vous du fichier `decalage.py`). Précisez la méthode utilisée.
2. Que pensez-vous du chiffrement Cesar ?

1.3 Chiffrement affine

On utilise un encodage du type $x \to ax + b$. C'est-à-dire une lettre associée à l'entier x est chiffrée par l'entier y reste de la division euclidienne de $ax + b$ par 26.

1. Peut-on toujours decoder ? c'est-à-dire retrouver x quand on connait y ? Donner des conditions et le nombres de codages possibles.
2. Ouvrez le fichier `Affine.py` avec le programme `IDLE`. Jetez un oeil aux procédures. En fin de programme, rajoutez les commandes

```
l=Affine([5,8],3,7)
print(l)
```

Faire tourner. Comprenez-vous ?

3. On a le texte suivant. pouvez-vous le décoder ? Vous pouvez pour cela utiliser une analyse fréquentielle... Voir `http://www.dcode.fr/analyse-frequences`

Texte 2

WPCGCHUA CVRRPQQ H AJUUP RJU RPUR AP QH WPHVGP FHGIPFHGXNVP PU LPR GPCFPR QPR FHGIPFHGXNVPR LJURXAPCPPR H QPVC MVRGP FPRVCP YJRRPAPUG UJU RPVQPFPUG QH KPCXGP FHXR QH WPHVGP RVYCPFP VUP WPHVGP ECJXAP PG HVRGPCP LJFFP LPQQP A'VUP RLVQYGVCP RHUR CPEPCPULP H VUP YHCGXP AP UJGCP ECHTXQP UHGVCP RHUR QPR PEEPGR A XQQVRXJU FHTUXEXNVPR AP QH YPXUGVCP JV AP QH FVRXNVP YJVCGHUG YVC PG RVWQXFP LHYHWQP A VUP YPCEPLGXJU RPKPCP GPQQP NVP RPVQPFPUG QPR YQVR TCHUAR HCGR YPVKPUG QH FJUGCPC Q PRYCXG KCHX AV YQHXRXC Q POHQGHGXJU Q XFYCPRRXJU A PGCP YQVR NV VU IJFFP NVX PRG QH YXPCCP AP GJVLIP AP Q POLPQQPULP QH YQVR PQPKPP AJXG PGCP GCJVKP AHUR QPR FHGIPFHGXNVPR HVRRX RVCPFPUG NVP QH YJPRXP

4. Et maintenant, essayer avec celui là qui est plus court ...

Texte 3

SZ O PL DPDMR JZ OP WJVWR

5. Que pensez-vous de ce cryptage ?