info614 : Mathématiques pour l'informatique
TD 2 : cryptographie I

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : http://www.lama.univ-savoie.fr/~hyvernat/
wiki : http://www.lama.univ-savoie.fr/wiki

Exercice 1 : Une cryptanalyse "facile"

Le système de César ($n \mapsto n+3 \pmod{26}$) ainsi que ces variantes (décalage de lettres) est assez facile à analyser : une fois qu'on a trouvé une seule lettre, les autres suivent...

Question 1. Essayez de faire une cryptanalyse de ce code, et envoyez moi le résultat par email. (Et un bonbon pour le premier qui me donne la référence du texte...)

```
Ib kbteczg, yb ahbygzs gt zig fgme-oeitg f'gbz-fg-veg pzg j'bvbew dzg gi bdbifniibit yg
vbewwgbz, tnzt agyb m'graetb b fnsmes. Yg mg anzahbe wzs y'hgsdg, pze gtbet tsgw keig, nz
jg kzw degitnt giwgvgye fbiw zi osnknif wnmmgey, pze fzsb igzk hgzsgw. Az dnzt fg ag
tgmow-yb, m'gtbit gvgeyyg, j'gwwbBbe fg mg ygvgs; mbew ag kzt gi vbei. Yg m'gtbew anzahg
wzs yg fnw; jg tsnzvbe mgw dsbw gt mgw jbmdgw bttbahgw b yb tgssg fg y'zi gt fg y'bztsg
antg, gt mgw ahgvgzr bttbahgw fg yb mgmg mbiegsg.  Yg tsnzvbe mgmg oyzwegzsw yecbtzsgw
tsgw meiagw pze gitnzsbegit mni ansow, fgozew mgw bewwgyygw jzwpz'b mgw azewwgw. Yg ig
onzvbew pzg sgcbsfgs gi hbzt; yg wnygey anmmgiabet b gtsg knst ahbzf, gt wb csbifg aybstg
dygwwbet mgw Bgzr. Y'gitgifew zi dszet anikzw bztnzs fg mne, mbew, fbiw yb onwtzsg nz
j'gtbew, jg ig onzvbew segi vnes pzg yg wnygey. Fegitnt jg wgitew sgmzgs pzgypzg ahnwg wzs
mb jbmdg cbzahg, gt agttg ahnwg, bvbiabit fnzagmgit wzs mb onetseig, mnitgs osgwpzg
jzwpz'b mni mgitni.  Wzgy kzt mni gtniigmgit ynswpzg j'bogsazw zig ogtetg keczsg fg
asgbtzsg hzmbeig hbztg tnzt bz oyzw fg tsnew onzagw, zi bsa gt zig kygahg b yb mbei, bvga
zi abspznew wzs yg fnw! Y'gi vew gi mgmg tgmow bz mneiw pzbsbitg bztsgw fg yb mgmg gwogag.
Yg mg mew wnzfbei b jgtgs fgw asew we hnssedygw, pzg tnzw agw ogtetw biembzr wg sgtesgsgit
tsbiwew fg ogzs; gt ey B gi gzt mgmg pzgypzgw-ziw, anmmg jg y'be boosew giwzetg, pze
kzsgit fbicgsgzwgmgit dygwwgw obs ygw ahztgw osgaeoetggw pz'eyw kesgit gi wbztbit fg
fgwwzw mni ansow b tgssg.
```

Question 2. Essayez de faire une cryptanalyse du code suivant (plus dur), et envoyez moi le résultat par email. (Et un paquet de bonbons pour le premier qui me donne la référence du texte...)

```
ymgxj u gbn.g,nfol ea gyngxj gofulegojycjgymgmfgc uxbyngxjbxgnfgu bmfnbtl ghbn
cfslegefstxgyx?gxjymgrs mxyfnpgojycjgbxgiyumxgmyajxghyajxgnfxgm  hgeyiiycslxp
ymgu bll.gfn gfigxj ghfmxgeyiiycslxgxjbxgcbngt gbm, evgoj ngo gjbw gu blyk e
xj gftmxbcl mgyngxj gob.gfigbgmxubyajxifuobuegbnegcfniye nxgbnmo upgo gmjbll
t go llglbsncj egfngxj gmxse.gfigqjylfmfqj.--ifugqjylfmfqj.gymgh u l.gxj
bxx hqxgxfgbnmo ugmscjgslxyhbx grs mxyfnmpgnfxgcbu l mml.gbnegefahbxycbll.pgbm
o gefgyngfueynbu.glyi gbneg w ngyngxj gmcy nc mpgtsxgcuyxycbll.pgbix u
 zqlfuynagbllgxjbxghb, mgmscjgrs mxyfnmgqskklynapgbnegbix ugu blykynagbllgxj
wbas n mmgbnegcfnismyfngxjbxgsne uly gfsugfueynbu.gye bmv

yngebyl.glyi pgo gbmmsh gbmgc uxbynghbn.gxjynamgojycjpgfngbgclfm ugmcusxyn.p
bu gifsnegxfgt gmfgisllgfigbqqbu nxgcfnxubeycxyfnmgxjbxgfnl.gbgau bxgbhfsnxgfi
xjfsajxg nbtl mgsmgxfg,nfogojbxgyxgymgxjbxgo gu bll.ghb.gt ly w vgyngxj
```

```
m bucjgifugc uxbynx.pgyxgymgnbxsublgxfgt ayngoyxjgfsugqu m nxg zq uy nc mpgbne
yngmfh gm nm pgnfgefstxpg,nfol ea gymgxfgt ge uyw egiufhgxj hvgtsxgbn.
mxbx h nxgbmgxfgojbxgyxgymgxjbxgfsugyhh eybx g zq uy nc mghb, gsmg,nfogymgw u.
ly, l.gxfgt goufnavgyxgm  hmgxfgh gxjbxgygbhgnfogmyxxynagyngbgcjbyupgbxgb
xbtl gfigbgc uxbyngmjbq pgfngojycjgygm  gmj  xmgfigqbq ugoyxjgouyxynagfu
quynxvgt.gxsunynagh.gj begygm  gfsxgfigxj goynefogtsyleynamgbnegclfsemgbnegxj
msnvgygt ly w gxjbxgxj gmsngymgbtfsxgnyn x.-xju  ghyllyfnghyl mgiufhgxj
 buxj;gxjbxgyxgymgbgjfxgalft ghbn.gxyh mgtyaa ugxjbngxj g buxj;gxjbxpgfoynagxf
xj g buxj'mgufxbxyfnpgyxguym mg w u.ghfunynapgbnegoyllgcfnxyns gxfgefgmfgifu
bngyne iynyx gxyh gyngxj gisxsu vgygt ly w gxjbxpgyigbn.gfxj ugnfuhblgq umfn
cfh mgynxfgh.guffhpgj goyllgm  gxj gmbh gcjbyumgbnegxbtl mgbnegtff,mgbne
qbq umgbmgygm  pgbnegxjbxgxj gxbtl gojycjgygm  gymgxj gmbh gbmgxj gxbtl gojycj
ygi  lgqu mmynagbabynmxgh.gbuhvgbllgxjymgm  hmgxfgt gmfg wye nxgbmgxfgt
jbuel.gofuxjgmxbxynapg zc qxgyngbnmo ugxfgbghbngojfgefstxmgoj xj ugyg,nfo
bn.xjynavgg. xgbllgxjymghb.gt gu bmfnbtl.gefstx epgbnegbllgfigyxgu rsyu mghscj
cbu islgeymcsmmyfngt ifu go gcbngt gmsu gxjbxgo gjbw gmxbx egyxgyngbgifuhgxjbx
ymgojfll.gxus
```

- les codes sont disponibles sur le wiki
- la fréquence des lettres en français est disponible sur wikipedia
- vous pouvez écrire votre propre petit programme pour compter ou changer les lettres (quelques infos supplémentaires sur le wiki)

Exercice 2 : Un exercice un peu chiant : le DES "simplifié"

Nous allons utiliser une version simplifiée de DES : "simplified DES". Une description succincte suit, et les détails seront donnés à l'oral.

Une clé secrète partagée de 10 bits permet de générer deux sous clés de 8 bits :
- on applique la permutation $(3, 5, 2, 7, 4, 10, 1, 9, 8, 6)$ sur la clé
- le résultat est divisé en deux parties de 5 bits, et chaque partie est "décalée" (permutation circulaire) vers la gauche. On obtient (A_1, A_2)
- on applique la fonction $P = (6, 3, 7, 4, 8, 5, 10, 9)$ sur (A_1, A_2) pour obtenir la première sous-clé : K_1
- on décale A_1 et A_2 de deux bits à gauches pour obtenir (B_1, B_2) et on applique la fonction P pour obtenir la deuxième sous-clé K_2

Pour l'encodage proprement dit, on utilise des blocs de 8 bits. On procède comme suit :
- on commence par permuter le bloc avec $(2, 6, 3, 1, 4, 8, 5, 7)$
- le bloc est divisé en deux : (L_0, R_0) et on calcule $\big(L_1 = R_0, R_1 = L_0 \oplus F(R_0, K_1)\big)$
- on calcule $\big(L_2 = L_1 \oplus F(R_1', K_2), R_2 = R_1\big)$
- on termine en utilisant la permutation inverse de la première étape : $(4, 1, 3, 5, 7, 2, 8, 6)$

Il reste maintenant à définir la fonction F : elle prend en entrée un bloc de 4 bits et une sous clé K_i. Ensuite,
- elle l'étend en un bloc de 8 bits avec la fonction $(4, 1, 2, 3, 2, 3, 4, 1)$
- elle ajoute ($\oplus$) la sous-clé K_i
- les 4 premiers bits sont passés dans S_1 et les 4 derniers dans S_2 (voir plus bas)
- le résultat (4 bits) est finalement permuté avec $(2, 4, 3, 1)$; c'est le résultat.

Les matrices S_1 et S_2 sont définies par

$$S_1 = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \qquad \text{et} \qquad S_2 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}$$

et sont utilisées comme suit : étant donnés 4 bits (b_1, b_2, b_3, b_4)

- on utilise les bits (b_1, b_4) pour former un entier en base 2, qui nous donne la ligne
- les bits (b_2, b_3) forment un entier en base 2 qui donne la colonne
- on renvoie les deux bits constituant l'entrée correspondante dans la matrice.

Le véritable DES est très similaire, mais utilise
- des clés de 58 bits
- des blocs de 64 bits
- 16 itérations au lieu de 2.

Question 1. Dessinez le circuit correspondant au DES simplifié.

Question 2. On veut encoder, avec la clé (1001111011), le texte "`OK`" qui correspond, en ASCII, à la chaîne de bits (1001111 1001011). Comment l'encodez-vous ?

Question 3. On veut décrypter, avec la clé (1111111111) la chaîne : (11110010 00101001). Quel est le texte clair ?

Question 4. Programmez (chez vous) le DES simplifié dans le langage de votre choix.

Question 5. Le DES double n'est pas considéré comme sûr. La raison est que le temps nécessaire pour casser DES double n'est pas très différent du le temps pour casser DES. Le seul problème est qu'il faut disposer d'une quantité de mémoire énorme...

Supposons qu'Alice envoie un message à Bob DES double : $c = \mathrm{DES}\big(\mathrm{DES}(m, k_1), k_2\big)$. Si un observateur malveillant (Eve) peut essayer de casser le code par une attaque "meet in the middle". Pour faire ça, Eve doit disposer d'au moins un message clair m et de son code c ; Eve peut ensuite calculer $\mathrm{DES}(m, k)$ pour toutes les clés k possible, et décoder c avec $\mathrm{DES}^{-1}(c, k)$ pour toutes les clés possibles k.

Comment peut-on se servir de ces résultats pour casser l'encryption de DES double ? Est-ce que c'est facilement faisable en pratique ?

Exercice 3 : échange de clé de Diffie-Hellman

Question 1. Faites tourner l'algorithme d'échange de clé de Diffie-Hellman avec les valeurs suivantes
- $p = 11$ comme nombre premier
- $g = 2$ comme générateur de $\mathbf{Z}/p\mathbf{Z}$
- $a = 4$ comme nombre secret choisi par Alice
- $b = 8$ comme nombre secret pour Bob

Détaillez les calculs en mettant en avant les messages échangés par Alice et Bob. Quelle est la clé ainsi obtenue ?

Question 2. Vérifiez que 2 est bien un élément générique de $\mathbf{Z}/p\mathbf{Z}$. Est-ce que 3 est générique ? Que se passe-t'il si g n'est pas générique ?

Question 3. Que se passe-t'il si le canal de communication est compromis et qu'un observateur malveillant (Eve) écoute les communications ?

Question 4. Pouvez-vous généraliser le protocole d'échange pour partager une clé entre trois personnes ? Entre quatre ?