École normale supérieure **2019-2020**
Département d'informatique

Introduction à la cryptologie
TD n° 1 : Correction

Exercice 1. Pour cryptanalyser un tel message, il suffit de trouver le chiffré de "e" qui est la lettre la plus fréquente en français, et d'en déduire le décalage. Ici la lettre du chiffré la plus fréquente est le "s", on en déduit que le décalage est de 14. Le texte clair est le suivant :

```
horsdicirentrezfaineansrentrezchezvousestceaujour
dhuifetequoinesavezvouspasquevousautresartisansvo
usnedevezcirculerdanslesrueslesjoursouvrablesquav
eclessignesdevotreprofessionparlequelesttonmetier
```

Il s'agit du début du *Julius Cæsar* de Shakespeare (en VF).

Exercice 2.

1. La matrice $(((i+j) \pmod n)+1)_{1\leq i,j\leq n}$ vérifie les propriétés voulues.
2. On appelle M, C, K les variables alétoires du message, du chiffré, et de la clef. Soit $(m,c) \in \mathcal{M}\times\mathcal{K}$. On pose k l'unique entier qui vérifie $T(k,m) = c$. On montre que $P(M = m|C = c) = P(M = m)$.

$$P(M = m|C = c) = \frac{P(M = m \ \wedge \ C = c)}{P(C = c)}$$

   - $P(M = m \wedge C = c) = P(M = m)P(C = c|M = m) = P(M = m)P(K = k) = P(M = m)/n$.
   - $P(C = c) = \sum_{(m',k'):T(k',m')=c} P(M = m' \wedge K = k') = \sum_{m'} P(M = m')/n = 1/n$.
3. Soit T le carré latin dont les lignes et les colonnes sont indexées par $\mathbb{Z}_2^k$, et où l'intersection de la ligne i avec la colonne j contient la valeur $i + j \in \mathbb{Z}_2^k$. On remarque que le one-time pad est le chiffrement associé au carré latin T. La question précédente permet de conclure.

Exercice 3 (Schéma de Feistel)**.**

1. On remarque qu'avec un schéma de Feistel à un tour, la moitié de droite de l'état est préservée avec probabilité 1. Dans le cas d'une permuation aléatoire, cette probabilité est de $2^{-n/2}$.
2. On demande de chiffrer $(0^{n/2}, 0^{n/2})$, puis $(1^{n/2}, 0^{n/2})$, et on regarde le XOR entre les blocs de droite des deux cryptogrammes obtenus. Si on a affaire à un schéma de Feistel à deux tours, on obtient $1^{n/2}$ avec probabilité 1. Pour une permutation aléatoire, la probabilité est proche de $2^{-n/2}$.
3. On suit l'indication. Pour simplifier les dessins, par rapport à l'énoncé, on choisit de réprésenter le schéma de Feistel « en échelle », c'est-à-dire qu'on « déplie »la transposition entre les branches gauche et droite à chaque tour, comme sur la Figure 1 (en particulier, comme il y a un nombre impair de tours, les branches gauche et droite sont inversées dans cette représentation par rapport à la représentation classique). Dans le cas où on a affaire à un schéma de Feistel à trois tours, on peut vérifier la relation $X_0^R \oplus Z_0^R = X_3^R \oplus Y_3^R$ avec probabilité 1 : voir Figure 1 pour une illustration de la propagation des erreurs. Dans le cas aléatoire on aura seulement une probabilité proche de $2^{-n/2}$ que cette relation soit vérifiée.

Exercice 4 (Cryptanalyse de Ladder-DES)**.**

1. On pose $X_0^R = 0^n$. Si $(i \neq j)$, $X_{3,i}^L \oplus X_{3,j}^L = f_2(X_{0,i}^R \oplus f_1(0)) \oplus f_2(X_{0,j}^R \oplus f_1(0))$.
   Comme les f_i sont des permutations, on en déduit que le résultat est nécessairement non nul.
2. On pose $N = 2^n$. La probabilité de non-collision est égale à $\prod_{i=0}^{t-1}\left(1 - \frac{i}{N}\right)$. On a l'encadrement suivant : $\left(1 - \frac{t}{N}\right) \leq \left(1 - \frac{i}{N}\right) \leq e^{-\frac{i}{N}}$ (pour l'inégalité de droite, un argument rapide est que $x \mapsto e^x$ est convexe, et $x \mapsto 1 + x$ est sa tangente en 0). On obtient :

$$\left(1 - \frac{t}{N}\right)^t \leq \prod_{i=0}^{t-1}\left(1 - \frac{i}{N}\right) \leq e^{-\sum_{i=0}^{t-1}\frac{i}{N}}.$$

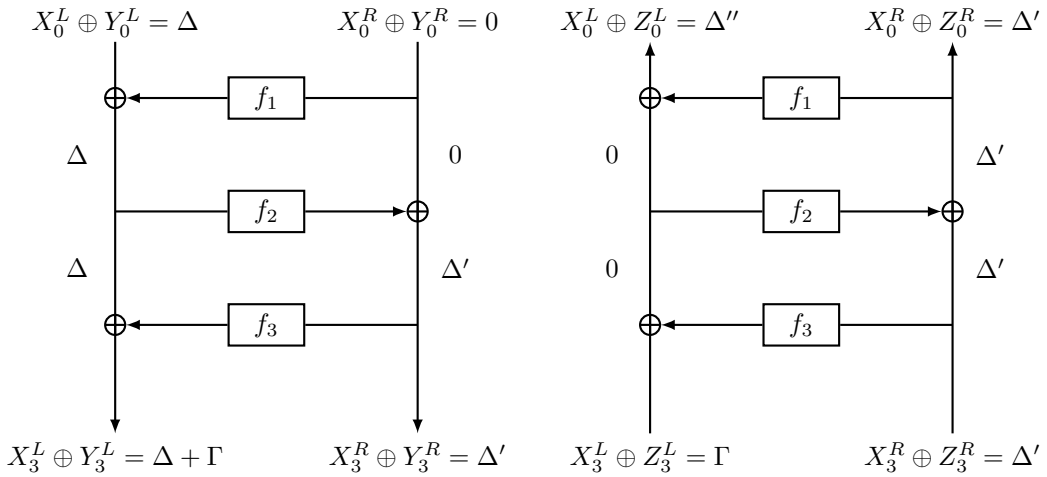

FIGURE 1 – Propagation de différence (XOR) sur un schéma de Feistel à trois tours. À gauche, différences entre les chiffrements de X_0 et Y_0. À droite, différences entre les déchiffrements de X_3 et de Z_3. Noter que dans le schéma de droite, les entrées de f_3 sont les mêmes que dans le schéma de gauche, c'est pourquoi la différence en sortie de f_3 est la même (Γ avec les notations du schéma).

Du côté gauche, pour $N \to \infty$ on a $\left(1 - \frac{t}{N}\right)^t = e^{\frac{t^2}{N} \ln\left(1-\frac{t}{N}\right)\frac{N}{t}} \sim e^{-\frac{t^2}{N}}$. Donc si t est négligeable en l'infini devant $\sqrt{N}$, on n'aura jamais de collision car le terme de gauche tend vers 1.
Du côté droit, $e^{-\sum_{i=0}^{t-1} \frac{i}{N}} = e^{\frac{t(t-1)}{2N}}$. Donc si $t = \sqrt{N}$ Le terme de droite tend vers $\frac{1}{\sqrt{e}}$. On en déduit que la probabilité de collision est minorée (asymptotiquement) par $1 - \frac{1}{\sqrt{e}} \geq 0.39$.

3. Ici on va utiliser plusieurs hypothèses heuristiques : on va tout d'abord supposer que $N = 2^{64}$ est suffisament grand pour que le raisonnement de la question précédente soit valide. De plus, soit $E(a) = \{Enc((X_{0,i}^L, X_0^R))\}$ un ensemble construit comme dans la question 1, où on a fixé $X_0^R = a$.
On suppose que pour $K \neq K_4$, la fonction $F_K\left(C^l, C^R\right) \mapsto \mathsf{DES}_K^{-1}\left(C^L\right) \oplus C^R$ se comporte comme une fonction aléatoire en ce qui concerne le nombre de collisions qu'elle génère sur les éléments de $E(a)$ (quel que soit a) ; et que tous ces évènements sont indépendants.
On choisit $E(a)$ de taille maximale, c'est-à-dire $|E(a)| = 2^{32}$ (en effet DES opère sur des blocs de 64 bits, donc une branche du schéma de Feistel comporte 32 bits). Suite aux hypothèse ci-dessus, on peut appliquer la question précédente : pour $K \neq K_4$, la probabilité qu'il y ait une collision entre les éléments $F_K(x)$ quand x parcourt $E(a)$ (pour a fixé) est minorée par $e^{-1/2} \approx 0,6$. Par contre, d'après la question 1, si $K = K_4$ cette probabilité est 0. On peut utiliser cette différence de comportement pour retrouver K_4, par exemple comme suit.
On crée une liste contenant toutes les clefs K possibles (2^{56} possibilités). On choisit un valeur de a, et pour chaque clef on calcule $F_K(x)$ pour tout $x \in E(a)$. S'il y a une collision, K est retirée de la liste. Tant qu'il reste plus d'une clef dans la liste on recommence avec une nouvelle valeur de a. D'après la question précédente et les hypothèses ci-dessus, on s'attend à ce qu'à chaque itération une proportion au moins $e^{-1/2} \approx 0,6$ des mauvaises clefs (les clefs $K \neq K_4$) soient éliminées. Si on choisit n tel que $e^{-n/2} = 2^{-56}$, i.e. $n = 128\log(2) \approx 90$, et qu'on répète ce processus n fois, on s'attend donc à ce qu'il ne reste qu'une clef environ. Lorsqu'il ne reste qu'une clef, cette clef est nécessairement K_4, puisque d'après la question 1, K_4 n'est jamais éliminée. .
Quelle est la quantité de calculs attendue ? Une clef $K \neq K_4$ non encore éliminée est éliminée à l'itération suivante (i.e. lorsqu'on la teste sur un nouvel $E(a)$) avec probabilité au moins $e^{-1/2}$. L'espérance du nombre d'itérations pour éliminer une clef K donnée est donc $\sqrt{e}$. Chaque itération où cette clef n'a pas encore été éliminée coûte $|E(a)| = 2^{32}$ calculs d'un tour de Feistel (i.e. un DES, qu'on va utiliser comme unité) pour cette clef, donc le total des 2^{56} (techniquement, $2^{56} - 1$) clefs à éliminer va coûter environ $2^{56} \cdot 2^{32}\sqrt{e} \approx 2^{90}$ opérations.
De plus, à chaque itération il faut calculer $E(a)$ pour un nouveau a, ce qu'on peut estimer à 2^{32} opérations supplémentaires par itération (un peu plus si on compte le nombre de DES, mais dans la mesure où nous calculons un ordre de grandeur il n'est pas nécessaire d'être trop précis). Le coût total pour les calculs des $E(a)$ peut donc être estimé à $2^{32} \cdot 128\log(2) \approx 2^{63}$ opérations. Il est donc, en principe, négligeable. (En pratique on remarque cependant que pour calculer $E(a)$ nous

devons faire ce nombre d'appels « online » à la primitive de chiffrement qu'on attaque, tandis que les calculs du paragraphes précédents étaient « offline », c'est-à-dire qu'on peut les faire à loisir chez soi, on n'est pas tributaire de la vitesse avec laquelle on peut obtenir le chiffré de messages clairs choisis.)

Globalement, on peut donc évaluer grossièrement le nombre d'opérations requises à 2^{90} opérations. Crucialement, ce nombre est plus petit qu'une attaque par force brute (2^{224}), et même qu'une attaque générique par rencontre au milieu, comme dans l'exercice 6 (2^{112}). Il s'agit donc d'une attaque valide.

Note : il y a des manières plus efficaces de trouver K_4 en utilisant la même idée générale ; en particulier on peut facilement concevoir une variante sans mémoire de l'algorithme précédent.

4. Avec la question précédente, on sait retrouver la clef du dernier tour du schéma de Feistel. On peut donc annuler le dernier tour et se ramener à un schéma de Feistel à 3 tours au lieu de 4 (en effet à chaque fois qu'on chiffre, on peut inverser le dernier tour du schéma de Feistel grâce à la clef connue). On peut alors répéter le principe de l'exercice. En effet on remarque qu'après chaque étape, on a un morceau de chiffré partiel qui est strictement différent pour tout les clairs demandés, c'est-à-dire qu'on a un distingueur du type de la question 1. On peut donc appliquer la méthode de la question précédente pour retrouver la clef du tour concerné, et itérer le processus.

   Une autre approche est de dégager l'idée suivante de cet exercice : si on a un comportement « anormal »pour un schéma de Feistel à k tours (par exemple celui de la question 1 ; on parle en cryptographie de *distingueur* entre ce schéma et une permutation aléatoire), on peut attaquer la clef du schéma à $k+1$ tours en testant toutes les clefs possibles sur le dernier tour. L'occurrence du comportement anormal fournit un filtre qui permet de tester si la clef qu'on a devinée sur le dernier tour est correcte. En suivant cette idée on peut utiliser les distingueurs de l'exercice 3 pour attaquer un schéma de Feistel sur 2,3,4 tours. À chaque fois on retrouve la clef du dernier tour, ce qui permet de se ramener à un schéma de Feistel avec un tour de moins.

Exercice 5 (Cryptanalyse de Magenta)**.** En utilisant l'oracle de chiffrement Enc, on peut créer la fonction suivante indexée par une clef de 64 bits K : $G_K : C \mapsto F_K^{-2}(Enc(F_K^{-2}))$. Si $K = K_1$ la fonction G_K se comporte comme un schéma de Feistel à deux tours. Sinon elle se comporte comme une permutation aléatoire (hypothèse heuristique). On peut donc utiliser la question 2 de l'exercice 3 pour distinguer dans quel cas, on se trouve, et donc si $K = K_1$.
Il y a 2^{64} choix pour K, et le filtre en question conserve une proportion 2^{-64} des mauvaises clefs (distinctes de K_1). On s'attend donc à avoir éventuellement quelques faux positifs. Ces quelques clefs peuvent être éliminées (avec très forte probabilité) en refaisant un test du même type une seconde fois (pour refaire un test du même type, on choisit par exemple une autre constante pour la partie droite de l'état dans la question de l'exercice 3). En fin de compte de compte on obtient K_1. On peut ensuite trouver K_2 par force brute.

Exercice 6 (Chiffrement Triple-DES avec deux clefs indépendantes)**.** L'idée est de faire une rencontre au milieu en devinant séparément K et K^*, comme suit. On remarque :

$$\mathsf{DES}_K^{-1} \circ \mathsf{Triple\text{-}DES}_{K,K^*} \circ \mathsf{DES}_K^{-1}(0) = \mathsf{DES}_{K^*}^{-1}(0)$$

où 0 représente le message nul.

- Pour chaque clef L on calcule $\mathsf{DES}_L^{-1} \circ \mathsf{Triple\text{-}DES}_{K,K^*} \circ \mathsf{DES}_L^{-1}(0)$ (l'appel à Triple-DES est réalisé en utilisant un message à clair choisi). On enregistre le résultat dans une table de hachage, en lui associant la valeur de L correspondante.
- Pour chaque clef L^* on calcule $\mathsf{DES}_{L^*}^{-1}(0)$. Si cette valeur existe dans la table de hachage précédente, on ajoute (L, L^*) à une liste de clefs candidates (initialement vide).
- Si au terme des étapes précédentes on obtient plus d'une clef candidate, on vérifie la validité de chaque clef candidate (ici une clef est une paire (L, L^*)) en utilisant un message clair connu. On élimine les clefs qui ne coïncident pas avec l'oracle $\mathsf{Triple\text{-}DES}_{K,K^*}$ sur ce clair connu. On recommence jusqu'à ce qu'il ne reste qu'une clef, qui est nécessairement la clef cherchée (K, K^*).

Il y a 2^{112} clefs (K, K^*) possibles. Le filtre fourni par le test d'égalité ci-dessus porte sur 64 bits ; on s'attend donc à obtenir environ 2^{48} clefs candidates. Le test sur sur un message clair choisi de la dernière étape fournit un autre filtre sur 64 bits. On s'attend donc qu'après seulement un seul tel test par clef, seule la bonne clef reste. La quantité de calcul peut donc être estimée à 2^{56} opérations environ. (Chaque opération correspond à quelques chiffrements ou déchiffrement DES.)