Projet de Cryptologie

Master Mathématique MIC

Second semestre

Projet DES

Etudiants :

Berthet Pierre-Augustin
Checri Marina
Coolen Julien
Renard Marc
Scotti Martin
Trama Daphné

Professeur :

Mesnager Sihem

Janvier-Mai 2021

Table des matières

I Chiffrement DES

I.1 Principe et fonctionnement du DES

Définition I.1 (DES)**.**
Le Data Encryption Standard est un algorithme de chiffrement symétrique par bloc. Ce système de chiffrement symétrique fait partie de la famille des chiffrements itératifs par blocs, plus particulièrement il s'agit d'un schéma de Feistel.
DES chiffre des blocs de 64 bits. On observe un bit de partié tous les 8 bits dans la clef. Ils servent à garantir un nombre impair de « 1 » dans chaque octet.

La clef possède donc une longueur « utile » (une dimension) de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

Nota Bene. La technique du chiffrement par bloc (ou « block cipher ») consiste à découper le message clair en blocs de l bits et à appliquer à chacun de ces blocs une clef de tours de l bits.

Exemples I.2 (Chiffrement par blocs)**.**
À part le DES, on peut notamment citer les chiffrements de Vigenère, le l'AES ou encore le TEA.

Définition I.3 (Schéma de Feistel)**.**
Le schéma de Feistel est une fonction de chiffrement (et déchiffrement) qui opère sur des blocs de longueur $2t$ en r tours. Plus formellement :
Soit $\Sigma = \{0, 1\}$ un alphabet binaire. On fixe :

- une méthode pour associer à une clef k, une fonction $f_k : 2^t \to 2^t$ (mais attention, f_k n'est pas forcément injective)
- un nombre de tours r
- une méthode pour générer d'une clef k, r clefs de tour $k_1, k_2, ..., k_r$.

⇨ Le **chiffrement** se passe comme suit :

- On divise le texte clair p en deux moitiés L_0R_0.
- On définit pour $i = 1, ..., r$:

$$L_iR_i = R_{i-1}(L_{i-1} \oplus f_{k_i}(R_{i-1})).$$

- On pose pour la fonction de chiffrement avec la clef k :

$$E_k(p) = L_r R_r.$$

⇨ Enfin, pour le **déchiffrement**, il faut procéder comme suit :

- Pour $L_i R_i = R_{i-1}(L_{i-1} \oplus f_{k_i}(R_{i-1}))$ on dérive :

$$\begin{aligned} R_{i-1}L_{i-1} &= L_i(R_i \oplus f_{k_i}(R_{i-1}) \\ &= L_i(R_i \oplus f_{k_i}(L_i)). \end{aligned}$$

- Étant donné un bloc de texte chiffré c, on le divise en $L_r R_r$ et on calcule $L_i R_i$ pour $i = r-1, ..., 0$ en utilisant les clefs $k_r, ..., k_1$.
- Alors le déchiffrement du chiffré c est donné par :

$$D_k(c) = L_0 R_0.$$

(Voir figure 1)

Bien sûr, moins le réseau possède de tours, plus fragile il est.

Remarque I.4.
Lorsque l'on cherche à chiffrer un fichier, sa taille n'est pas toujours un multiple de la longueur des blocs chiffrés par l'algorithme. Il existe différentes techniques, dites de « padding ». En voici deux :

- Si le dernier bloc à chiffrer du fichier n'est pas entier, on écrit dans son dernier octet le nombre de bits manquants. Les octets précédents peuvent être remplis aléatoirement.
- Si le dernier bloc est entier, on rajoute tout un bloc pour qu'il n'y ait pas d'ambiguïté.

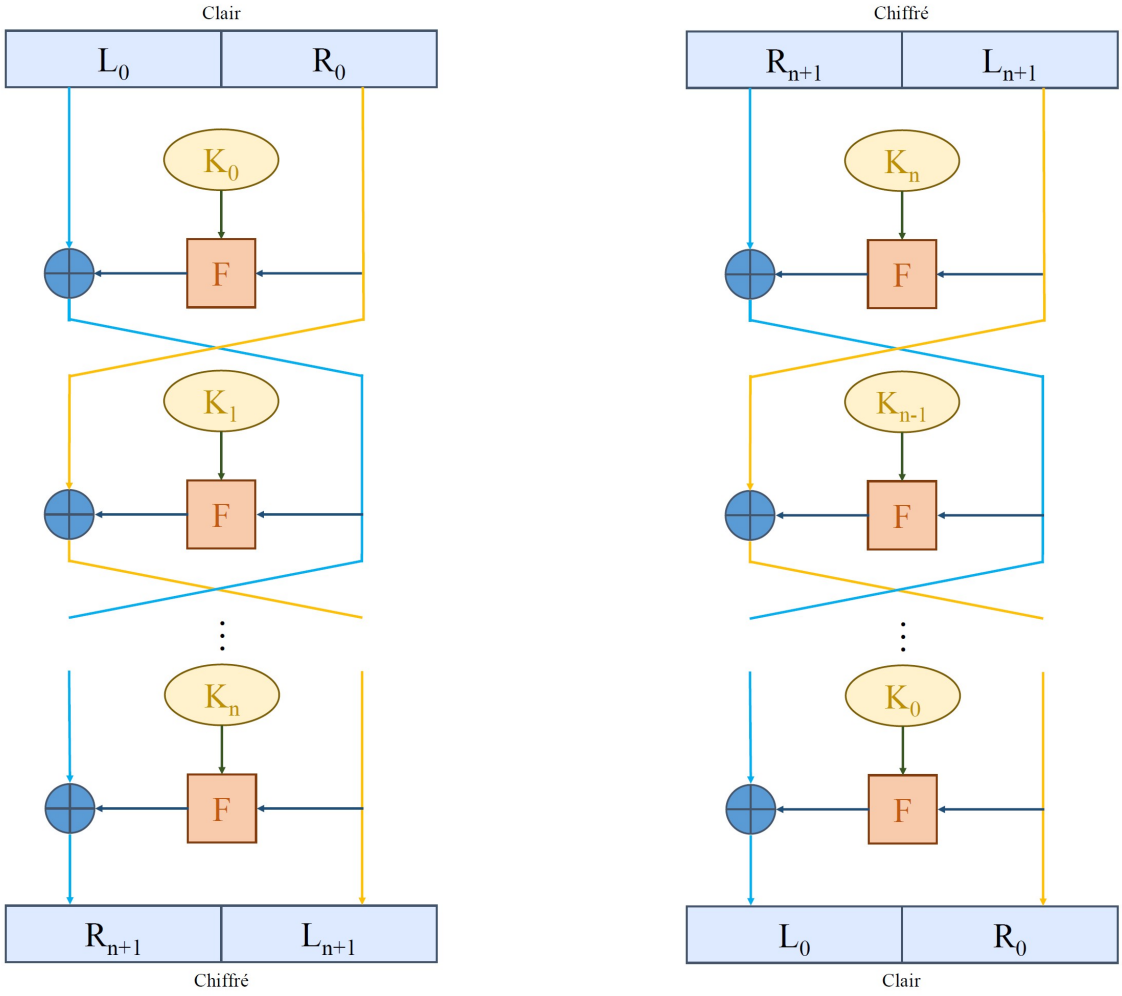

FIGURE 1 – Chiffrement (gauche) et Déchiffrement (droite) de Feistel

I.2 Diversification de la clef

Proposition I.5 (Diversification de la clé)**.**
En reprenant les explications du projet, la clef K est une chaîne de 64 bits dont 56 définissent la clef ($K \in {F_2}^{56}$), et 8 sont des bits de parité. Cette clef est diversifiée en 16 clefs de tour Kr, $r \in [\![0; 15]\!]$ de 48 bits. Les bits de parité sont ignorés dans le procédé de diversification : étant donnés les 64 bits de la clé K, on enlève les bits de parité et l'on ordonne les autres suivant une permutation PC1.

La figure 2 explique le principe pour obtenir des clefs diversifiées de 48 bits à partir d'une clef de 64 bits.

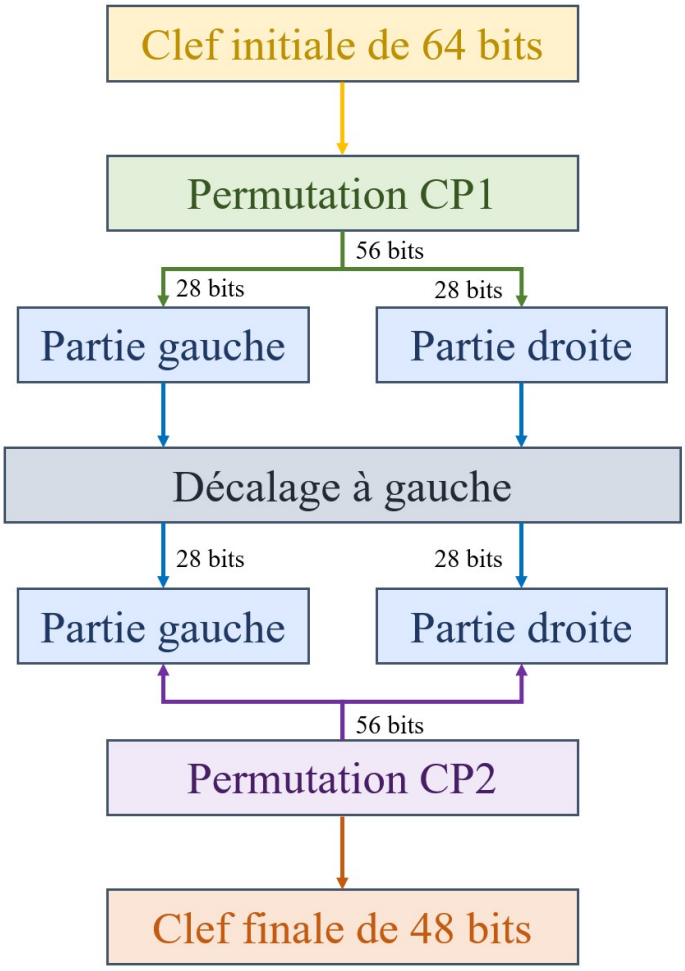

FIGURE 2 – Diversification des clefs

I.3 Chiffrement DES

Principe I.6 (Chiffrement DES)**.**
Le chiffrement DES suit, dans les grandes lignes, les étapes ci-dessous :

- *Couper le texte en blocs de* 64 *bits.*
- *Effectuer une permutation initiale fixe des blocs.*
- *Répeter des permutations et substitutions pendant* 16 *tours, chaque tour dépendant d'une clef* K_r, $r \in [\![0; 15]\!]$ *:*

  Pour r *allant de* 0 *à* 15*,*
  - *Découper le bloc de* 64 *bits en deux blocs de* 32 *bits.*
  - *Echanger les blocs selon un schéma de Feistel.*
  - *Recoller les deux parties.*

  Fin Pour
- *Effectuer une permutation finale, qui est la permutation initiale inverse.*

Propriété I.7.
L'algorithme de déchiffrement est le même que l'algorithme de chiffrement, moyennant l'inversion de l'ordre des sous-clefs.

Démonstration.
En pratique, le chiffrement consiste à effectuer une permutation initiale IP, puis à lancer des chiffrements de Feistel, et à appliquer la permutation inverse de l'initiale IP^{-1}. C'est-à-dire que le chiffré c obtenu à partir du message m se construit en calculant : $c = (\mathrm{IP} \circ F(K_0, \cdots, K_n) \circ \mathrm{IP}^{-1})(m)$, où $F(K_0, \cdots, K_n)$ représente le chiffrement de Fesitel effectué avec, dans l'ordre, les clefs K_0 à K_n.

Le déchiffrement consiste donc à appliquer la permutation initiale(qui s'annule avec l'inverse), puis à déchiffrer Feistel, c'est-à-dire, à utiliser le même algorithme que celui du chiffrement de Feistel, en prenant les clefs dans l'ordre inverse. Enfin, il suffit d'appliquer la permutation inverse de l'initiale, pour réobtenir le message original !

$$\begin{aligned} \text{On a} \quad & \mathrm{IP}^{-1} \circ i \circ g_k \circ \cdots \circ g_2 \circ g_1 \circ \mathrm{IP} \circ \mathrm{IP}^{-1} \circ i \circ g_1 \circ g_2 \circ \cdots \circ g_k \circ \mathrm{IP} \\ = \quad & \mathrm{IP}^{-1} \circ i(\circ g_k \circ (\cdots (\circ g_2 (\circ g_1 \circ i \circ g_1) \circ g_2) \circ \cdots) \circ g_k) \circ \mathrm{IP} \end{aligned}$$

Or, comme le montre la propriété I.8, $g_l \circ i \circ g_l = i$, pour tout $l \in [\![1; k]\!]$.

$$
\begin{aligned}
\text{D'où} \quad & \mathrm{IP}^{-1} \circ i \circ g_k \circ \cdots \circ g_2 \circ g_1 \circ \mathrm{IP} \circ \mathrm{IP}^{-1} \circ i \circ g_1 \circ g_2 \circ \cdots \circ g_k \circ \mathrm{IP} \\
= \quad & (\mathrm{IP}^{-1} \circ \mathrm{IP})(m) \\
= \quad & Id(m) \\
= \quad & m.
\end{aligned}
$$

De manière plus précise, on note f_{K_i} la suite des actions suivantes :

l'expansion appliquée à la partie droite,

le résultat « xoré » avec la clef,

le passage dans la Sbox

et l'application de la permutation P.

On suppose que l'on effectue k tours pour les opérations de chiffrement et déchiffrement. De la formule de chiffrement $L_i = R_{i-1}, R_i = L_{i-1} \oplus f_{K_i}(R_{i-1})$, on déduit la formule de déchiffrement $R_{i-1} = L_i, L_{i-1} = R_i \oplus f_{K_i}(L_i)$. Le chiffrement produit en sortie $\mathrm{IP}^{-1} R_k L_k$. On aboutit avec le déchiffrement à $\mathrm{IP}^{-1} R_0 L_0$ en appliquant la formule de déchiffrement k fois, puis on applique la permutation IP et l'on échange R et L pour obtenir $L_0 R_0$.

□

Propriété I.8.
Soit $F(K_i)$ un tour du schéma de Feistel, et i la permutation qui intervertit les parties gauches et droites.
Alors $i \circ F(K_i) \circ i \circ F(K_i) = i$

Démonstration.
La figure 3 montre l'idée derrière la démonstration.

Plus rigoureusement, prenons $r = 3$, pour simplifier. L'exemple sera généralisable à tout $r \in \mathbb{N}$ par récurrence immédiate.

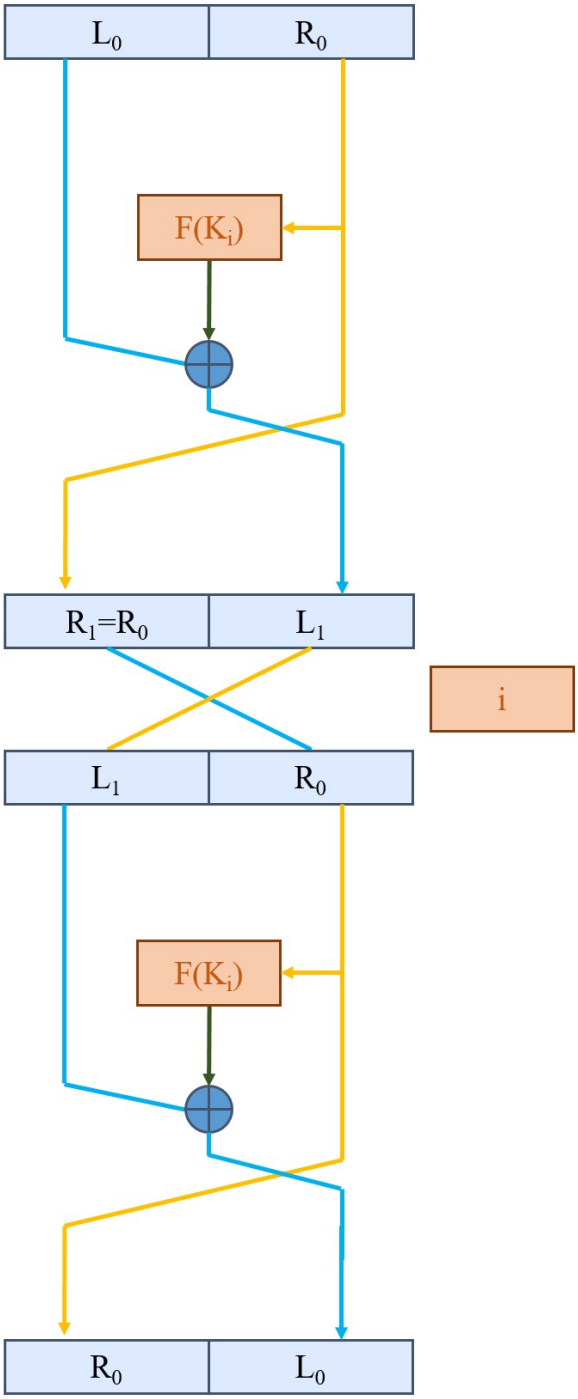

FIGURE 3 – $F(K_i) \circ i \circ F(K_i) = i$

On note donc g_1, g_2, g_3 les tours du schéma de Feistel.

$$\underbrace{\mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 \circ g_1 \circ \mathrm{IP}}_{\text{DES avec la clef dans l'ordre inverse pour déchiffrer}} \circ \underbrace{\mathrm{IP}^{-1} \circ i \circ g_1 \circ g_2 \circ g_3 \circ \mathrm{IP}}_{\text{DES initial}}$$

$$\begin{aligned}
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 \circ g_1 \circ (id) \circ i \circ g_1 \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 (\circ i \circ i) \circ g_1 \circ i \circ g_1 \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 \circ i \circ (i \circ g_1 \circ i \circ g_1) \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 \circ i \circ (id) \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ g_2 \circ i \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 (\circ i \circ i) \circ g_2 \circ i \circ g_2 \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ i \circ g_3 \circ i \circ g_3 \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} (id) \circ \mathrm{IP} \\
&= \mathrm{IP}^{-1} \circ \mathrm{IP} \\
&= id
\end{aligned}$$

□

I.4 Constantes du DES

Le chiffrement DES possède donc plusieurs constante. Parmi elles, on compte notamment :

- les permutations initiale et finale (inverse de l'initiale) notées `IP` et `RFP` ;
- deux permutations `PC1` et `PC2` servant à la création des 16 sous-clefs K_r de 48 bits.
- `E` qui sert à obtenir 48 bits à partir de 32, en dupliquant deux bits sur quatre. On l'utilise à l'entrée du réseau de Feistel.
- une permutation `P` qui s'effectue en sortie du réseau de Feistel.
- des décalages pré-déterminés lors de la création des sous-clefs. On appelera le tableau les représentants `shifts`.

I.5 Avantages et inconvénients

Avantages :

- Le chiffrement et le déchiffrement sont très rapides, les algorithmes de chiffrement symétrique sont généralement beaucoup moins complexes que les algorithmes de chiffrement asymétrique.

Inconvénients :

- Le chiffrement symétrique n'assure que la confidentialité des données, et pas l'intégrité ou l'authenticité.
- Une clef symétrique étant une entité personnelle, pour communiquer avec deux personnes différentes, il faudra deux clefs différentes, une pour chaque interlocuteur.
- L'utilisation d'une clef unique présente un problème : il faut pouvoir communiquer la clef de manière sûre à la personne avec laquelle on souhaite dialoguer.
- Il est nécessaire de garantir la confidentialité de cette clef. Les échanges qui suivront reposent sur celle-ci.
- Avec un algorithme de chiffrement par bloc, on ne peut commencer à chiffrer et à déchiffrer un message que si l'on connaît la totalité d'un bloc. Ceci occasionne naturellement un délai dans la transmission et nécessite également le stockage successif des blocs dans une mémoire tampon.
- Actuellement, le DES n'est plus utilisé à cause de sa faible capacité de clefs et de son manque de réactivité. Une attaque des données transmises à un destinataire serait systématiquement réalisée en un laps de temps raisonnable

Nota Bene.
À la fin des années 1990, le chiffrement DES est devenu trop faible pour résister à une attaque exhaustive. L'algorithme **Triple DES** (ou 3DES) fut alors adopté comme solution provisoire de remplacement. Il consiste en trois applications successives du DES, mais avec simplement deux clefs différentes (on utilise la même clef pour le premier et le dernier chiffrement du DES). En tout, on utilise donc une clef de $2 \times 56 = 112$ bits. Ce dernier possède une propriété intéressante vis à vis de son prédecesseur : en effet, tout chiffré DES peut être décrypté par un système 3DES, tandis qu'un chiffré 3DES, sous condition d'avoir utilisé 2 fois la même clé pour le chiffrer, peut être déchiffré par un système DES. En conséquence, la transition entre les 2 systèmes a pu se faire efficacement, les anciens systèmes DES pouvaient toujours être employés avec le nouveau standard.

II Cryptanalyse différentielle DES

Principe II.1.
L'attaque est une attaque à clair choisi : c'est-à-dire que l'attaquant possède la fonction de chiffrement, qu'il peut utiliser à sa guise comme une boîte noire. Aussi, l'attaquant peut générer autant de couples (clair, chiffré) qu'il le souhaite.

Dans cette attaque, on s'intérresse à la propagation des différences Δx de deux textes clairs (x, x') au fur et à mesure de l'algorithme.

À chaque étape, l'attaquant fait des statistiques et attribue des probabilités aux clefs possibles, en fonction des changements qu'elles ont apportés sur le couple de valeurs. Après avoir travaillé sur un grand nombre de couples clairs/chiffrés, la clef ayant la plus grande probabilité est gardée.

Explications de l'attaque (Cryptanalyse du schéma de Feistel et attaque sur le dernier tour du DES)**.**

➢ Recherche de caractéristiques différentielles.
On fabrique les tables des différences d'une S-box donnée.

On peut alors observer qu'en fixant une différence en entrée et en sortie, seules quelques valeurs sont possibles à l'entrée de la S-box.
En connaissant la valeur d'entrée dans le réseau de Feistel, et en sachant que la clef de tour est ajoutée avant de passer dans les S-box, l'ensemble des entrées possibles d'une S-box permet d'observer un ensemble de clefs possibles.

Le principe est alors de compter le nombre d'apparitions d'une clef dans l'ensemble des clefs possibles. Celle qui, avec un grand nombre d'échantillons est la plus probable a de grandes chances d'être la bonne.

En reprenant les explications de [Gér10] et [Wan08] on obtient un algorithme pour la cryptanalyse différentielle d'un réseau de Feistel :

- N : nombre de couples clairs/chiffrés
- Pour toutes les valeurs de clefs possibles k
  - $t_k := 0$

- Fin Pour
- Pour i allant de 1 à N et $k \in {\mathbb{F}_2}^{n_k}$
  - Si $P\left(S(E({c_1}^{i,g}) \oplus k) \oplus S(E({c_2}^{i,g}) \oplus k)\right) = {\delta_r}^g \oplus {c_1}^{i,d} \oplus {c_2}^{i,d}$, Alors $t_k := t_k + 1$
  - Fin Si
- Fin Pour
- Retourner $k_{r+1} = argmax_k \ \{t_k\}$

${c_1}^{i,g}$ (respectivement ${c_1}^{i,d}$) représente la première partie du chiffré à gauche (resp. droite) au tour i et ${c_2}^{i,g}$ (respectivement ${c_2}^{i,d}$) représente la deuxième partie du chiffré à gauche (resp. droite) au tour i.

➢ Attaque sur le dernier tour d'un chiffrement à 16 tours.

Pour attaquer $r + 1$ tours de chiffrement, on utilise une différentielle sur les r premiers tours. Il faut alors déchiffrer les chiffrés sur un tour avec tous les candidats possibles pour la sous-clef du dernier tour.

Références

[BS92] Eli Biham and Adi Shamir. Differential cryptanalysis of the full 16-round des. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '92, page 487–496, Berlin, Heidelberg, 1992. Springer-Verlag.

[Gér10] Benoît Gérard. *Cryptanalyses statistiques des algorithmes de chiffrement à clef secrète.* Theses, Université Pierre et Marie Curie - Paris VI, December 2010.

[Hey02] Howard M. Heys. A tutorial on linear and differential cryptanalysis. *Cryptologia*, 26(3) :189–221, July 2002.

[Wan08] Meiqin Wang. Differential cryptanalysis of reduced-round present. In Serge Vaudenay, editor, *Progress in Cryptology – AFRICACRYPT 2008*, pages 40–49, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.