

# Calcul sécurisé, Attaque par faute sur le DES

Niels Merceron  
Numéro d'étudiant: 21801038



# Table des matières

<b>1</b>	<b>Attaque par faute sur le DES</b>	<b>2</b>
1.1	Description du principe d'attaque par faute . . . . .	2
<b>2</b>	<b>Application concrète</b>	<b>3</b>
2.1	Description de l'attaque par faute . . . . .	3
<b>3</b>	<b>Retrouver la clef complète du DES</b>	<b>4</b>
<b>4</b>	<b>Fautes sur les tours précédents</b>	<b>5</b>
4.1	attaque sur le 14ème tour . . . . .	5
4.2	attaque sur le 13ème tour . . . . .	5
4.3	généralisation sur le nième tour . . . . .	5
<b>5</b>	<b>Contre-mesures</b>	<b>6</b>
5.1	Doublé le calcul du DES . . . . .	6
5.2	rajouté de l'aléatoire/calcul imbriqué . . . . .	6
<b>6</b>	<b>Annexe</b>	<b>7</b>

# Chapitre 1

## Attaque par faute sur le DES

### 1.1 Description du principe d'attaque par faute

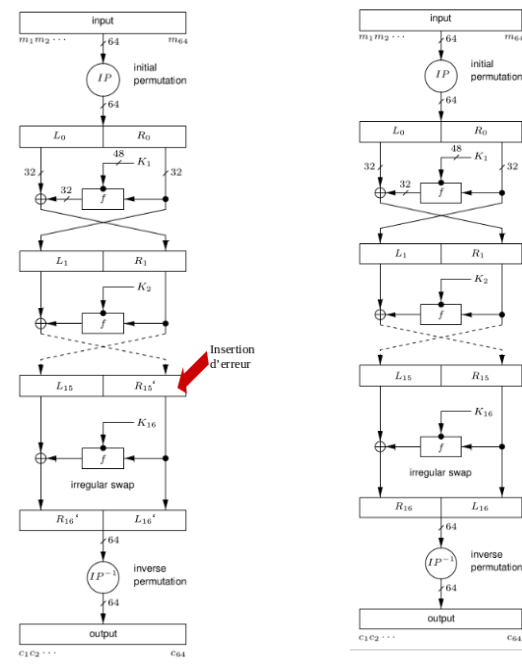


Schéma du DES avec et sans faute

On va injecter une faute à l'aide d'un laser dans un tour (dans notre cas le 15ème), de cette faute résultera un chiffré différent de celui sans faute. De ces deux chiffrés on les comparera en effectuant divers calculs dessus pour arriver à manipuler quelque chose de simple et qui nous donne de l'information sur la clé secrète. Plus précisément, on identifiera quel bit est touché par la faute. Une fois le bit identifié on regardera quel fonction sont touchées par ce bit fauté. Puis on effectuera le calcul des fonctions touchées avec la version non fauté et fauté. Puis on comparera ces deux fonctions pour trouver des informations sur la clé secrète  $K$ .

## Chapitre 2

# Application concrète

### 2.1 Description de l'attaque par faute

Pour commencer on veut obtenir R16 et L16 donc on effectue au chiffé correcte les permutations décrite par IP. On fait la même chose pour obtenir R16' et L16' qui correspondent au chiffé fauté.

On a donc  $R16 = R15$ ,  $L16 = L15 \oplus f(R15, K16)$  mais aussi  $R16' = R15'$  et  $L16' = L15 \oplus f(R15', K16)$ . Pour savoir où se situe l'erreur on effectue le calcul suivant  $R16 \oplus R16' = R15 \oplus R15' = R15 \oplus R15 \oplus \mathcal{E} = \mathcal{E}$ . Savoir où se situe l'erreur nous aidera plus tard.

Une fois cela fait on passe à la partie plus technique. Notre but est de trouver une formule avec ce que nous connaissons, avec K16 dedans et la plus réduite possible. On va donc commencer par effectuer :

$$L16' \oplus L16 = L15 \oplus f(R15', K16) \oplus L15 \oplus f(R15, K16) = f(R15', K16) \oplus f(R15, K16).$$

une fois cela fait on va écrire explicitement la fonction f. Cette fonction f consiste en une expansion(E) puis on xor le résultat avec  $K_i$  (dans notre cas  $K_{16}$ ) ensuite le résultat du xor passe dans les Sbox et pour finir passe dans une permutation(P). on obtient donc la formule suivante :

$$L16' \oplus L16 = P(S(E(R'_{15}) \oplus K_{16})) \oplus P(S(E(R_{15}) \oplus K_{16}))$$

On obtient donc comme partie de clef : B1A6E1869A35

## Chapitre 3

# Retrouver la clef complète du DES

Pour retrouver les 8 bits manquants on peut force brute sur  $2^8$  valeur et calculer le Des avec chaque clef créé a l'aide des  $2^8$

On obtient donc comme clef : E65875255B64BA40

## Chapitre 4

# Fautes sur les tours précédents

4.1 attaque sur le 14ème tour

4.2 attaque sur le 13ème tour

4.3 généralisation sur le nième tour

## Chapitre 5

# Contre-mesures

### 5.1 Doublé le calcul du DES

Une solution assez simple qui nous vient en tête est de calculé deux/plusieurs fois l'algorithme du DES pour le même message puis vérifier si les résultats obtenus à la fin est identique. Si les résultats sont identique alors il n'y a pas eu d'injection de faute, sinon une faute a été introduite on ne renvoie rien. Cependant cette méthode est partiellement efficace car on peut partir du principe que l'attaquant a plusieurs outils similaires pour faire la fautes au même endroit a chaque fois et donc que l'injection de faute soit un succès. Il faudrait donc calculer  $n+1$  fois le DES ( $n$  étant le nombre d'outils qu'a l'attaquant) pour espérer s'en protéger ce qui donnerai une complexité très grande. De plus si on voulait mettre cette solution dans une carte a puce on serait très vite limité au nombre de fois que l'ont puisse calculer le DES. La complexité de cette solution serait de l'ordre  $K \times O(\text{DES})$ ,  $K$  étant le nombre de fois que l'ont calcul le DES.

### 5.2 rajouté de l'aléatoire/calcul imbriqué

## Chapitre 6

## Annexe