Calcul sécurisé, Attaque par faute sur le DES

Niels Merceron
Numéro d'étudiant: 21801038

Table des matières

Chapitre 1

Attaque par faute sur le DES

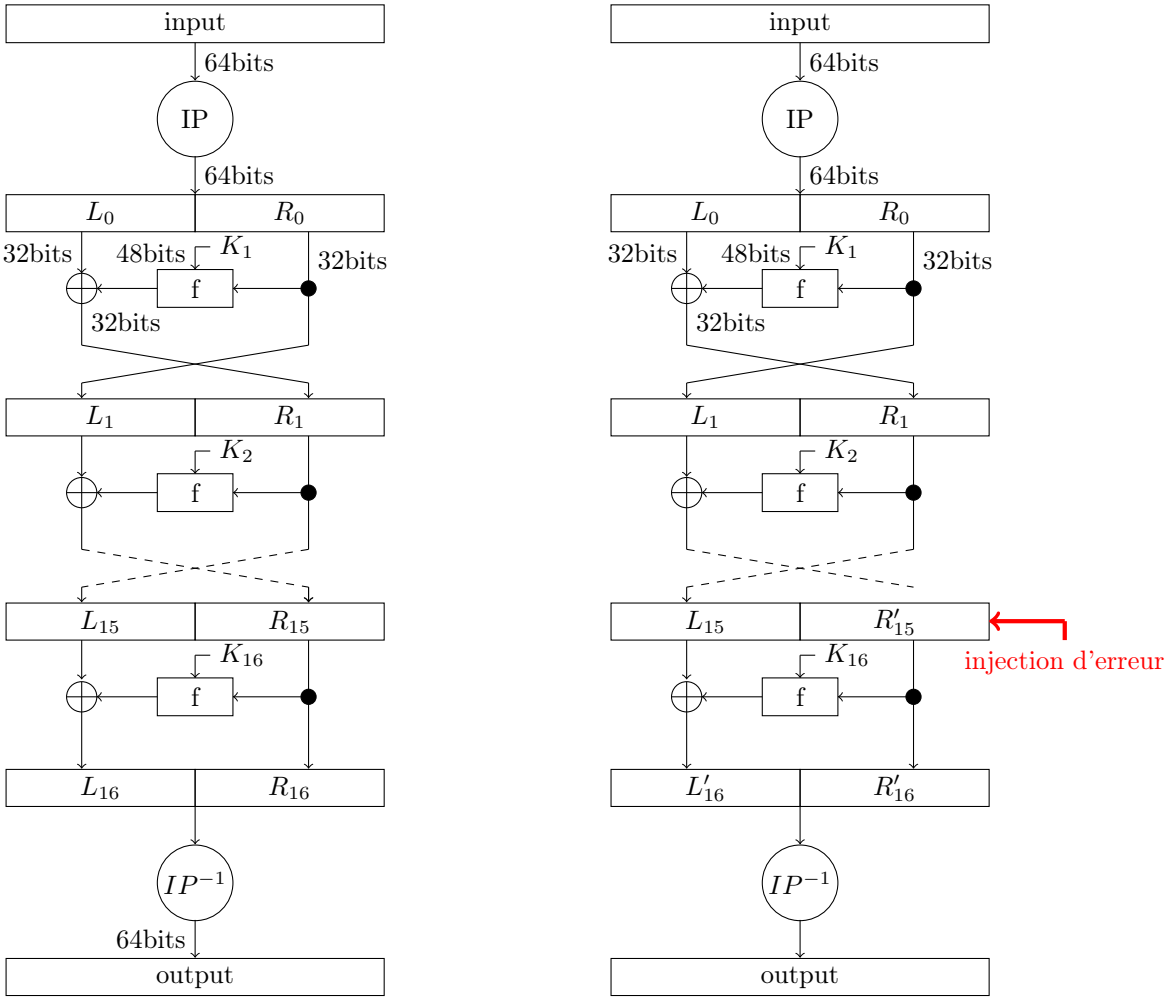

FIGURE 1.1 – Schéma du DES avec et sans faute

1.1 Description du principe d'attaque par 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 a manipulé quelque chose de simple et qui nous donne de l'information sur la clef secrète. Plus précisément, on identifiera quel bit est touché par la faute. Une fois le bit identifié, on regardera quelle 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 clef secrète K.

Chapitre 2

Application concrète

2.1 Description de l'attaque par faute

Pour commencer on veut obtenir R_{16} et L_{16} donc on effectue au chiffré correcte les permutations décrites par IP. On fait la même chose pour obtenir R'_{16} et L'_{16} qui correspondent au chiffré fauté.
On a donc $R_{16} = R_{15}$, $L_{16} = L_{15} \oplus f(R_{15}, K_{16})$ mais aussi $R'_{16} = R'_{15}$ et $L'_{16} = L_{15} \oplus f(R'_{15}, K_{16})$. Pour savoir ou ce situe l'erreur on effectue le calcul suivant $R_{16} \oplus R'_{16} = R_{15} \oplus R'_{15} = R_{15} \oplus R_{15} \oplus \mathcal{E} = \mathcal{E}$. Savoir ou ce 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 K_{16} dedans et la plus réduite possible. On va donc commencer par effectuer :
$L'_{16} \oplus L_{16} = L_{15} \oplus F(R'_{15}, K_{16}) \oplus L_{15} \oplus F(R_{15}, K_{16}) = F(R'_{15}, K_{16}) \oplus F(R_{15}, K_{16})$.

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).

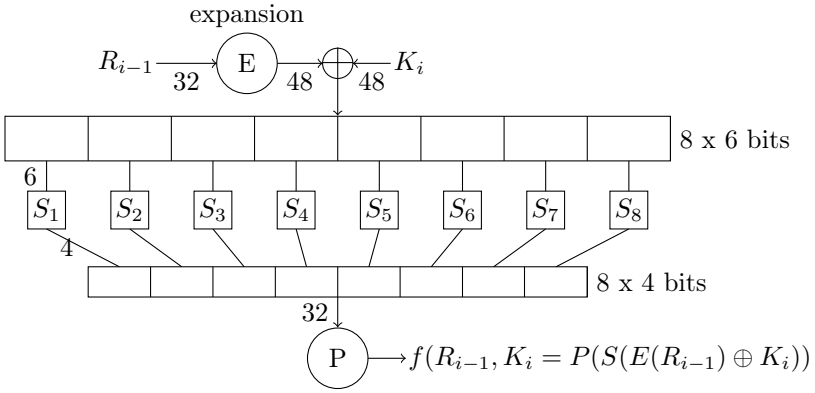

FIGURE 2.1 – Schéma de la fonction F

on obtient donc la formule suivant :

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

De cette formule on peut enlever la permutation P car on connais l'emplacement des bits permutés et la permutation P est linéaire.
on a donc :

$$P^{-1}(L'_{16} \oplus L_{16}) = P^{-1}(P(S(E(R'_{15}) \oplus K_{16})) \oplus P(S(E(R_{15}) \oplus K_{16})))$$

$$P^{-1}(L'_{16} \oplus L_{16}) = S(E(R'_{15}) \oplus K_{16}) \oplus S(E(R_{15}) \oplus K_{16})$$

À ce stade on ne peut pas réduire plus cette formule car S est non linéaire donc je ne peux pas l'enlever comme j'ai fait avec la permutation P.
On va nommer deux variable $\alpha = E(R_{15}) \oplus K_{16}$ et $\alpha' = E(R'_{15}) \oplus K_{16}$. Par force brute, on va déterminer tous les α' passant dans la sbox nous donnant les mêmes bits de sorti avant permutation. Cela représente une force brute sur 2^6valeurs a tester.On effectuera cette tâche pour chaque chiffré fauté en notre possession.
Je vais étendre les α' obtenu jusqu'à 48bits pour pouvoir effectuer les calculs suivant tout en gardant l'emplacement d'alpha parmi les 48bits rajouté.
R'_{15} est connu ($R'_{16} = R'_{15}$) donc je vais calculer l'expansion de R'_{15} puis effectuer un xor entre cette expansion et les α' déterminer précédement. De faire ce calcul me permettra d'avoir 6 bit de la clef K_{16} cependant j'aurai plusieurs candidats (notés γ) car j'ai plusieurs α'.
Dans un premier temps pour déterminer le bon candidat je vais calculer α avec comme clé un γ a l'emplacement de la sbox touché par la faute ce qui nous donnera une clef de 48 bits.j'effectuerai cette opération autant de fois qu'il y a de γ.
Dans un second temps, je passerai le résultat de α dans les sbox puis je vérifie que le résultat obtenu est le bon avant permuation. Si il est bon alors cela veut dire que j'ai trouvé les 6 bons bits de clef (γ) sinon je continue jusqu'a trouver le bon γ me donnant la bonne sortie avant permuation.

Exemple : la faute est situé au 30ème bit.

on a donc :
$S(E(R_{15}) \oplus K_{16}$ = 01 ..... 1010 1110
$E(R'_{15})$ = 10 ..... 1001 0110
α'_1= 00 ..... 0010 0111
α'_2= 00 ..... 0001 0100
$\gamma_1 = E(R'_{15}) \oplus \alpha'_1$= 10 ..... 1011 0001
$\gamma_2 = E(R'_{15}) \oplus \alpha'_2$= 10 ..... 1000 0010
α_{γ_1}= 01 ..... 1000 0101 , α_{γ_1} n'est pas le bon candidat car les 4 derniers bits sont différent de celui de $S(E(R_{15}) \oplus K_{16}$.
α_{γ_2}= 00 ..... 0110 1110 , α_{γ_2} les 4 derniers bit sont équivalent à $S(E(R_{15}) \oplus K_{16}$ donc les bit de 31 à 28 sont égaux a ceux de γ_2.
La compléxité serait de 2^6 par Sbox donc une compléxité de 2^9 au total.
Pour finir après avoir appliquer l'attaque décrite un peu plus haut j'obtiens la clef suivante : B1A6E1869A35

Chapitre 3

Clef complète du DES

Algorithm 1 DES cadencement de clef

Input : 64 bit de clef K=$k_1....k_{64}$.

Output : 16 clef de 48 bit $K_i, 1 \leq i \leq 16$.

1.Définir $v_i, 1 \leq i \leq 16$ tel que : $v_i = 1$ pour i $\in$ 1,2,9,16 ;$v_i = 2$ sinon.
2.T$\leftarrow$PC1(K) ;T est une moitié de 28 bit de (C_0,D_0).(utiliser la table PC1 pour choisir des bit de K :$C_0 = k_{57}k_{49}...k_{36}, D_0 = k_{63}k_{55}...k_4$).
3.Pour i de 1 à 16 calculer K_i de la manière suivante :
$C_i \leftarrow (C_{i-1} \leftarrow v_i), D_i \leftarrow (D_{i-1} \leftarrow v_i), K_i \leftarrow PC2(C_i, D_i)$.(utiliser la table PC2 pour sélectionner 48 bits de la concaténation de $b_1b_2...b_{56}deC_ietD_i : K_i = b_{14}b_{17}...b_{32}$).

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(a)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

(b)

FIGURE 3.1 – table PC1(a) avec C_i en haut, D_i en bas et PC2(b)

Pour retrouver la clef en entier on va calculer le cadencement de clef a l'envers.Plus précisément, on va inverser PC2 et PC1 pour retrouver l'emplacement de nos 48 bits avant cadencement. Une fois cela fait, nous connaissons donc 48 sur 56 bit donc il faut force brute sur 8 bit donc cela représente 2^8 valeurs a testé. si on veut retrouver la clef entière avec les bits de parité, il faudra juste couper la clef de 56 bit trouvé en 8 paquets de 7 bit puis rajouté un bit de telle manière que le nombre de 1 dans le paquet soit impaire.
Une fois cela fait j'obtient donc comme clef : E65875255B64BA40

Chapitre 4

Fautes sur les tours précédents

4.1 Tout est connu

Dans ce cas si on connait tous les états intermédiaires de chaque tour car on peut y accéder physiquement sur la puce ou la carte alors l'attaque décrite un peu plus haut est possible sur tous les tours du DES. Cependant il y a un point qui diverge, pour retrouver la clef totale du DES il faudra appliquer un cadencement adapté au tour concerné.

4.2 On ne connait que l'output

Dans ce cas on ne connaitra que $R_{16}, R'_{16}, L_{16}, L'_{16}, R_0 et L_0$.
équation du DES pour le tour 14 sans faute :
$L_{15} = R_{14} \mid R_{15} = L_{14} \oplus F(R_{14}, K_{15})$
$R_{16} = R_{15} \mid L_{16} = L_{15} \oplus F(R_{15}, K_{16})$
équation du DES pour le tour 14 avec faute sur R_{14} :
$L'_{15} = R'_{14} \mid R'_{15} = L_{14} \oplus F(R'_{14}, K_{15})$
$R_{16} = R'_{15} \mid L'_{16} = L'_{15} \oplus F(R'_{15}, K_{16})$

On remarque que la faute c'est propagé et que donc retracé où a été la faute est compliqué. Il faudra faire un guest de bit fauté pour retracer où est cette faute. De plus si on choisi de faire ça on devra connaitre L_{15} et L'_{15} pour nous faciliter la tâche sinon c'est très compliqué et cela augmenterai grandement la complexité.
Cette recherche exhaustive va élever la complexité de 2^4. on aura donc comme compléxité 2^{13} De plus pour les tours plus haut, il faudra éléver cette attaque au carré car il faudra guest plus de bit fauté. Donc pour le tour 13 on aura 2^{26}. Pour le tour 12 on aura 2^{52}.
Pour le tour 11 on aura 2^{104}.On voit qu'a partir du tour 11 cela ne sert plus a rien de faire cette attaque car pour un ordinateur de nos jours, elle est incalculable dans un temps raisonnable, de plus elle excède grandement la valeur de force brute de recherche de la clef totale (2^{64}).

Chapitre 5

Contre-mesures

5.1 Doublé le calcul du DES

Une solution assez simple qui nous vient en tête est de calculer deux/plusieurs fois l'algorithme du DES pour le même message puis véfirier 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 faute 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 donnerait une complexité très grande. De plus si on voulait mettre cette solution dans une carte à puce on serait très vite limité au nombre de fois que l'on puisse calculer le DES. La complexité de cette solution serait de l'ordre K×O(DES), K étant le nombre de fois que l'ont calcul le DES.

5.2 Diffusion de l'erreur

Une autre solution que j'ai discuté avec mon chargé de td est la diffusion de l'erreur. Plus précisément, le but est de propager l'erreur à tous les bits, plusieurs fois de manières non linéaire pour que le chiffré obtenu par l'attaquant ne puisse lui donner aucune information sur la clef ou tout autre élément utile a l'attaquant. Cependant dans les faits cette solution n'a pas encore été réalisée pour le DES ou autre chiffrement du même type que le DES. Il y a un problème majeur, trouvé une méthode qui diffuse l'erreur quand il y en a une mais ne diffuse rien quand il n'y a pas d'erreur et garde le sens du messsage. À énoncé verbalement c'est simple mais dans les faits aucun modèle ou solution concrète n'a été décrite a ce jour dans la littérature Cryptographique.

5.3 Protection physique

On peut penser a une protection physique qui détecte quand l'intégrité de la carte est forcée et bloque donc le calcul du DES et même on pourrait aller plus loin pour que l'assaillant n'ait aucune information. La carte pourrait totalement effacer ces données stockées dès que son intégrité est mis en péril.