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

1.1 Description du principe d'attaque par faute

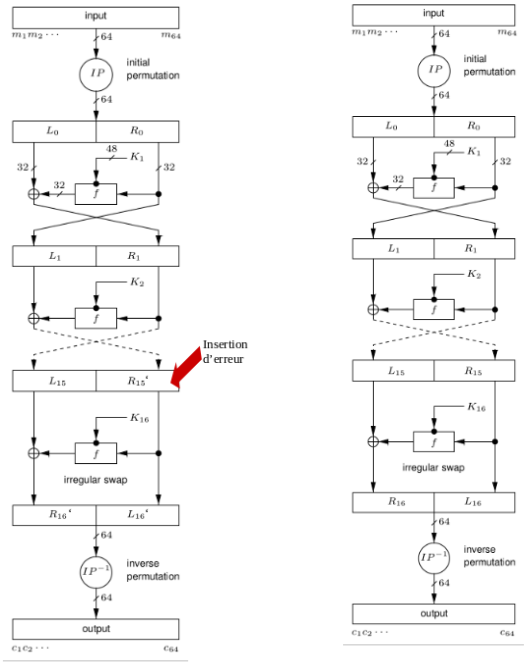

Schéma du DES avec et sans faute

On va injecté une faute a 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 calcul dessus pour arriver a manipulé quelque chose de simple et qui nous donne de l'information sur la clef secrète. Plus précissement, on identifera quel bit est touché par la faute. Une fois le bit identifié on regardera quel fontion sont touchées par ce bit fauté. Puis on effectura le calcul des fonctions touchés 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 R16 et L16 donc on effectue au chiffré correcte les permutations décrite par IP. On fait la même chose pour obtenir R16' et L16' qui correspondent au chiffré 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 ou ce situe l'erreur on effectue le calcul suivant $R16 \oplus R16' = R15 \oplus R15' = R15 \oplus R15 \oplus \mathcal{E} = \mathcal{E}$. Savoir ou ce situe l'erreur nous aidera plus tard.

Une fois cela fait on passe a 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 $L16 \oplus L16'$ pour supprimer L15 de l'équation :
$L16' \oplus L16 = L15 \oplus f(R15', K16) \oplus L15 \oplus f(R15, K16) = f(R15', K16) \oplus f(R15, K16)$.

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 suivant :

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

La permutation P est linéaire donc on peut appliquer P^{-1} a notre formule pour enlever P. on obtient donc la formule suivante :

$$P^{-1}(L16' \oplus L16) = P^{-1}(P(S(E(R'_{15}) \oplus K_{16})) \oplus P(S(E(R_{15}) \oplus K_{16})))$$
$$P^{-1}(L16' \oplus L16) = S(E(R'_{15}) \oplus K_{16}) \oplus S(E(R_{15}) \oplus K_{16})$$

Une fois cela fait on peut remarquer qu'on ne peut pas plus réduire notre équation car les Sbox (S) ne sont pas linéaire. Nous allons donc force brute 2^6 valeurs sur la ou les deux Sbox celon l'emplacement de l'erreur déterminée un peu plus haut.
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 véfirier si les résultats obtenu a 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 compléxité très grande. De plus si on voulais 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 compléxité 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 propagé l'erreur a 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 fait cette solution n'a pas encore été réalisé pour le DES ou autre chiffrement du même type que le DES. Il y a un problème majeur avec cette méthode il faut déjà détecter la faute ce qui n'est pas trop compliqué en soit mais de cette détection découle le fait d'appliquer ou non la diffusion de la faute. A 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.

Chapitre 6

Annexe