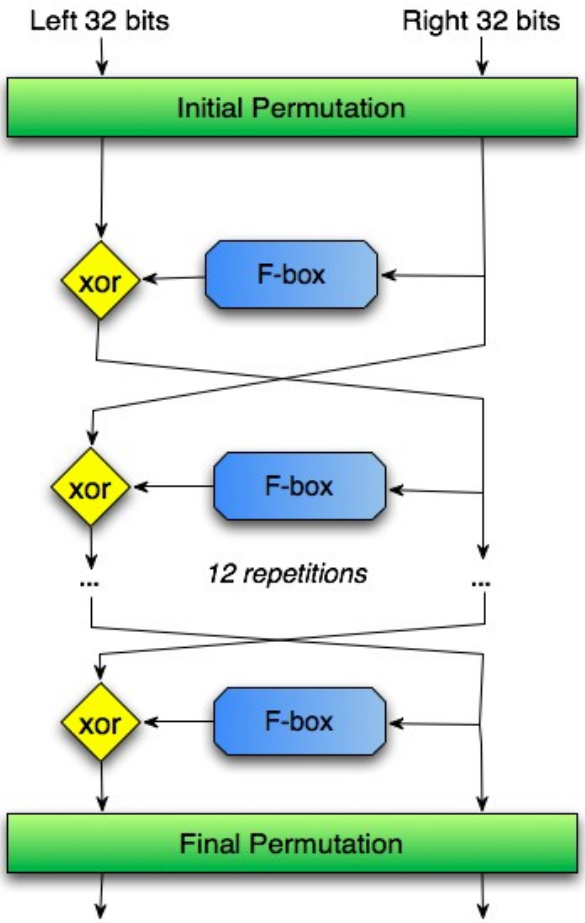
Rapport De Projet

Attaque par faute sur DES

Slimani Arezki 21502933



Question 1 :

Description De l'attaque par faute contre le DES :

L'attaque par faute contre le DES est l'une des attaques qui permet d'obtenir la clef de chiffrement d'un message chiffré, sa particularité est qu'il est plus rapide que certain algorithme de recherche tel que la recherche exhaustive qui a une complexité dans le pire des cas de 2^{56} (la clef fait 64 bit dont 8 bits de parité, donc 56 bits utiliser).

Le concept de l'attaque par faute sur le DES consiste a perturber le comportement du circuit afin de modifier l'exécution correcte du chiffrement. Les fautes son injecter dans le circuit par différents moyens tel que : le laser (sur les cartes a puces), des impulsions lumineuses, la perturbation de l'alimentation, champs magnétiques etc.

Passons maintenant a la pratique. On suppose que l'attaquant est capable d'effectuer une faute sur la valeur de sortie R_{15} du 15^e tour de Feistel. Concrètement cela veut dire que l'attaquant doit changer un seul bit parmi les 32 bits de R_{15}, et celui ci doit procéder de cette manière pour les 32 bits, Observons de plus près ce qu'induit l'injection d'une faute à la sortie de R_{15} qu'on notera R^*_{15}(voir le Figure 1):

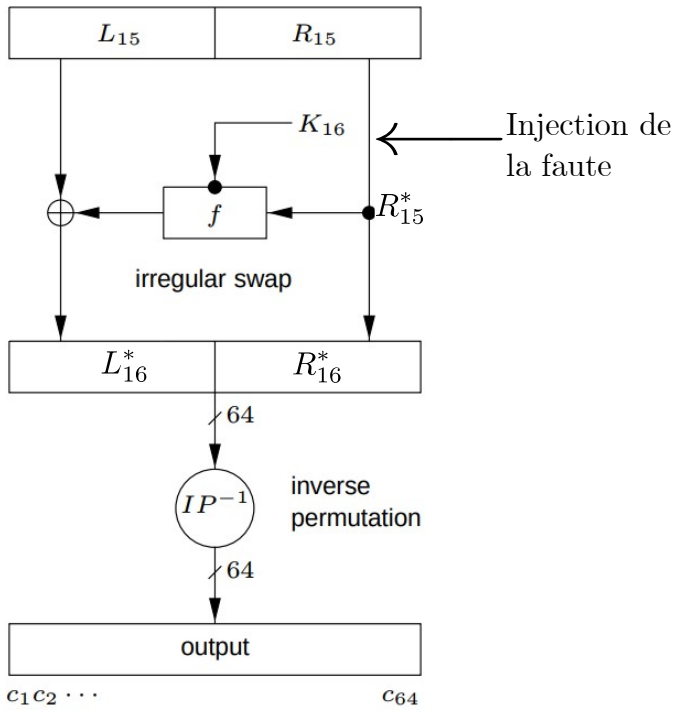


Figure 1: Injection d'une faute sur R_{15}

Comme vous pouvez le remarquer sur le schéma 1, l'injection de la faute sur R_{15} a induit une faute sur R^*_{16} et L^*_{16}, passons maintenant à la partie mathématique et regardons ce que l'injection de faute nous permet d'avoir comme information, essayant de comparer les sorties obtenue avec et sans injection de faute :

1) sans injection de faute :

$$R_{16} = R_{15}$$
$$L_{16} = L_{15} \oplus f(R_{15}, K_{16})$$

2) avec injection de faute :

$$R^*_{16} = R^*_{15}$$
$$L^*_{16} = L_{15} \oplus f(R^*_{15}, K_{16})$$

Si on analyse bien ces deux cas, on vois bien que la partie qui nous permettrai d'obtenir un jour la clef K_{16} est celle de gauche gauche (avec L_{16} et L^*_{16}) car la clef K_{16} n'apparaît que dans ces deux équations. Si on analyse bien L_{16} et L^*_{16} on peut remarquer que L_{15} apparaît dans les deux, donc dans un premier temps, nous allons essayer de le faire disparaître de l'équation, pour cela nous devant faire un XOR de L_{16} et L^*_{16} comme suis :

$$\begin{array}{c} 0 \\ || \\ L_{16} \oplus L^*_{16} = (L_{15} \oplus L_{15}) \oplus f(R_{15}, K_{16}) \oplus f(R^*_{15}, K_{16}) \\ \Downarrow \\ L_{16} \oplus L^*_{16} = f(R_{15}, K_{16}) \oplus f(R^*_{15}, K_{16}) \end{array}$$

Essayant maintenant d'approfondir nos recherche on s'intéressant a la fonction interne f pour essayer de trouver une solution au problème (voir la Figure 2):

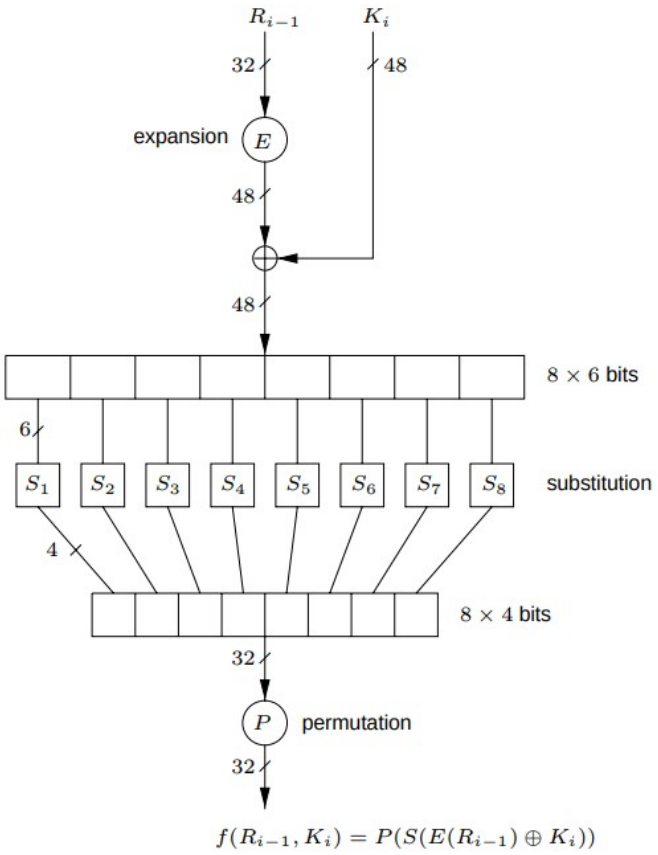


Figure 2: Fonction interne f du DES

Comme on peut le voir sur la figure 2, il y a l'équation de la fonction interne f écrite en bas, reprenons la et analysons la en détail pour les deux cas (avec faute et sans faute) :

$$f(R_{15}, K_{16}) = P(Sbox(E(R_{15}) \oplus K_{16}))$$
$$f(R^*_{15}, K_{16}) = P(Sbox(E(R^*_{15}) \oplus K_{16}))$$

Commençons par le début, la fonction f prend en paramètre R_{15} qui fait 32 et une clef K_{16} de 48 bit, avant de faire un XOR entre R_{15} et K_{16} on doit d'abord faire passer R_{15} de 32 à 48 bit par le biais d'une expansion noté E, maintenant que les deux élément son à 48 bits ont peut leur appliquer un XOR, c'est la que la partie la plus intéressante commence. Ces 48 bits obtenue devrons passer dans des boites de substitution (S-box), il y en a 8 au total, donc 8 paquet de 6 bits serons former avant de passer dans les S-box, toute les S-box son différentes, au final les 8 paquets recevrons un traitement différents, ces S-box dont on reparlera après (car ce sont la cible principal de l'attaque) prennent 6bit en entré et en renvoi 4, au total on aura 32 bits en sortie, voici la répartition des 48 sur ces S-box :

$$Sbox_1(E(R_{15}) \oplus k_{16\ \text{bits}\ 1\to6})$$
$$Sbox_2(E(R_{15}) \oplus k_{16\ \text{bits}\ 7\to12})$$
$$\dots$$
$$\dots$$
$$Sbox_8(E(R_{15}) \oplus k_{16\ \text{bits}\ 43\to48})$$

Ceci dit, il reste encore une permutation P avant de renvoyer le résultat final qui fait 32 bits. La permutation appliquer a la fin de la fonction nous gênent un peut pour mener a bien notre attaque, pourquoi ? Tout simplement parce que les 32 bits n'aurons plus le même ordre qu'a la sortie des S-box, et comme on le verra plus tard dans le rapport, pour trouver des morceaux de la clef K_{16} nous devrons surveiller le résultat d'une ou plusieurs S-box en particulier (les S-box concerner dépendent de l'emplacement du bit fauté).

Pour ce débarrasser de la permutation P nous devant appliquer la permutation inverse de celle ci qu'on notera l'inverse P^{-1} on a :

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

D'après la propriété d'une permutation P, $P(a \oplus b) = P(a) \oplus P(b)$:

$$\begin{gathered} P^{-1}(L_{16} \oplus L^*_{16}) = P^{-1}(P(Sbox(E(R_{15}) \oplus K_{16})) \oplus P(Sbox(E(R^*_{15}) \oplus K_{16}))) \\ \Updownarrow \\ P^{-1}(L_{16} \oplus L^*_{16}) = P^{-1}(P(Sbox(E(R_{15}) \oplus K_{16}))) \oplus P^{-1}(P(Sbox(E(R^*_{15}) \oplus K_{16}))) \\ \Updownarrow \\ P^{-1}(L_{16} \oplus L^*_{16}) = Sbox(E(R_{15}) \oplus K_{16}) \oplus Sbox(E(R^*_{15}) \oplus K_{16}) \end{gathered}$$

Si on découpe cette équation en 8 selon le nombre de S-box, nous avons 8 nouvelle équation suivante :

$$\begin{gathered} P^{-1}(L_{16} \oplus L^*_{16\ \text{bits } 1\to4}) = Sbox_1(E(R_{15}) \oplus K_{16})_{\ \text{bits } 1\to4} \oplus Sbox_1(E(R^*_{15}) \oplus K_{16})_{\ \text{bits } 1\to4} \\ P^{-1}(L_{16} \oplus L^*_{16\ \text{bits } 5\to8}) = Sbox_1(E(R_{15}) \oplus K_{16})_{\ \text{bits } 5\to8} \oplus Sbox_1(E(R^*_{15}) \oplus K_{16})_{\ \text{bits } 5\to8} \\ \dots \\ \dots \\ \dots \\ P^{-1}(L_{16} \oplus L^*_{16\ \text{bits } 29\to32}) = Sbox_1(E(R_{15}) \oplus K_{16})_{\ \text{bits } 29\to32} \oplus Sbox_1(E(R^*_{15}) \oplus K_{16})_{\ \text{bits } 29\to32} \end{gathered}$$

Comme vous pouvez le constater, on a 8 équation différentes a résoudre, et la seul inconnu dans toute ces équation son les 6 bits de K_{16}, donc la seul chose qu'il reste a faire est de tester toute les valeur possible des 6 bits de K_{16} en faisant une recherche exhaustive sur chacune des S-box, ce qui a pour complexité 2^6 pour chacune, au total on aura 8 x 2^6 test a faire, donc la complexité de cette attaque est de 2^6 x 2^3 ce qui fait 2^9 pour trouver la clef K_{16}.