Control Continu Calcul Sécurisé, Avril 2021

Alhadj Adam Mamadou 21920273

Master 1 Informatique SeCReTs

Table des matières

1 I : Attaque par faute sur le DES

Une attaque par faute consiste à changer le résultat d'un sous calcul afin d'obtenir une information secrète. Cette attaque est physique qui provoque une erreur volontairement. pour cela il faut modifier la valeur de certains bits afin d'agir physiquement sur les composants électroniques.

Dans notre cas l'attaque se produit à la sortie R_{15} du 15^e tour, Du coup la valeur R_{15} va être changer par l'attaquant.

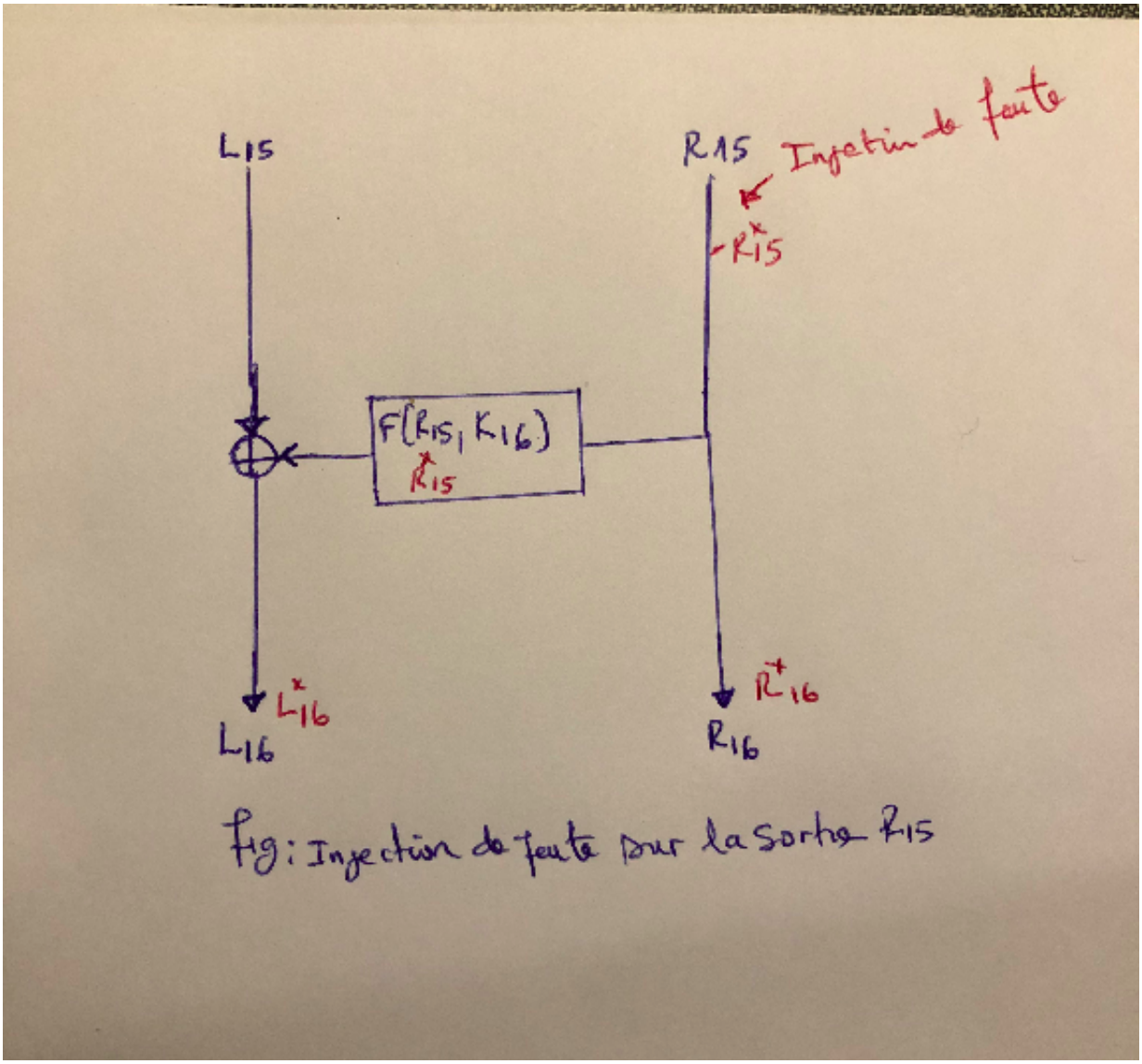


fig : Injection de faute sur la sortie R_{15}

Après l'injection de faute l'attaquant est capable de retrouver la clé secrète de la victime à partir de sous clé K_{15}. A partir de message clair ainsi que le chiffré correspondant puis les 32 messages chiffré obtenus, il est bien facile à déterminer la clé qui a permis de chiffré le message.

2 II : Application concrète

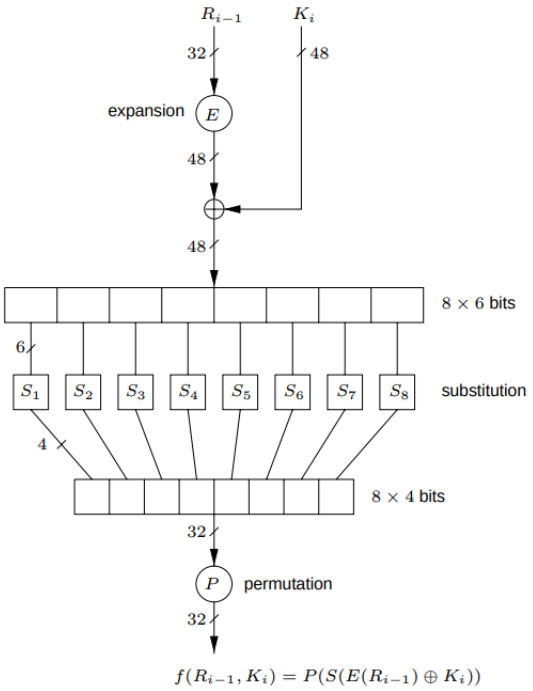
2.1 Question 1

Cette attaque par faute sur le dernier tour du DES comporte plusieurs étapes :

* Tout d'abord, l'objectif de cette attaque consiste à modifier 1 bit sur les 32 bits qui composent R_{15} afin d'obtenir une sortie faussée que nous noterons $R_{15}*$ .La figure ci-haute permet de voir les répercussions de l'injection d'une telle faute à la sortie du 15 ème tour du DES. Après la propagation, on obtient donc un L_{16} et un R_{16} fauté que nous noterons $L_{16}*$ et $R_{16}*$ .Nous allons donc étudier les résultats obtenus à la sortie du dernier tour à l'aide des formules connues suivantes :

$L_{16} = L_{15} \oplus f(K_{16}, R_{15})$ et $R_{16} = R_{15}$ pour le chiffré juste.
$L_{16}* = L_{15} \oplus f(K_{16}, R_{15}*)$ et $R_{16}* = R_{15}*$ pour les chiffrés faux.
Du coup on va effectuer un XOR entre L_{16} et $L_{16}*$ pour éliminer L_{15} et on obtient :
$L_{16} \oplus L_{16}* = f(K_{16}, R_{15}) \oplus f(K_{16}, R_{15}*)$

* L'étape suivante consiste à appliquer cette fonctionnalité sur chaque boite S-Box afin d'établir les équations pour cela on voit comme marche le DES grâce à l'image ci-dessous :



Pour trouver $f(K_{16}, R_{15})$ = P(S(E(R_{15})) $\oplus$K$_{16}$)) et $f(K_{16}, R_{15}*)$ = P(S(E($R_{15}*$))$\oplus$K$_{16}$))
Il va falloir appliquer l'expansion E à R_{15} et $R_{15}*$ afin de transformer le message constituer de 32 bits à 48 bits avant de faire XOR avec le K_{16}.

On récupère donc la position pour les 32 chiffrés faux et on va ensuite regarder où ce bit est propagé à travers la permutation d'expansion

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Ceci va donc nous permettre de cibler exactement quelle SBOX est affectée par quel fauté.

Pour chaque X-Box après la permutation inverse P^{-1} on obtient :

$P^{-1}(L_{16} \oplus L_{16}*)_{b_1 \to b_4} = S_1(E(R_{15}) \oplus K_{16})_{b_1 \to b_4} \oplus S_1(E(R_{15}*) \oplus K_{16})_{b_1 \to b_4}$
$P^{-1}(L_{16} \oplus L_{16}*)_{b_5 \to b_8} = S_2(E(R_{15}) \oplus K_{16})_{b_5 \to b_8} \oplus S_2(E(R_{15}*) \oplus K_{16})_{b_5 \to b_8}$
—
—
—
—
—
$P^{-1}(L_{16} \oplus L_{16}*)_{b_{29} \to b_{32}} = S_8(E(R_{15}) \oplus K_{16})_{b_{29} \to b_{32}} \oplus S_8(E(R_{15}*) \oplus K_{16})_{b_{29} \to b_{32}}$

* l'étape suivante consiste à éliminer les équations dont $P^{-1}(L_{16} \oplus L_{16}*)_{b_x \to b_y}$ vaut 0, ainsi que celle dont $S_i(E(R_{15}) \oplus K_{16})_{b_x \to b_y} = S_i(E(R_{15}*) \oplus K_{16})_{b_x \to b_y}$.

* En suite faire une attaque exhaustive sur la sortie de chaque boîte-S de $S_i(E(R_{15}) \oplus K_{16})_{b_x \to b_y}$: pour chaque élément, on déduit les possibles valeurs d'entrée de la boîte-S $E(R_{15}) \oplus K_{16}$. (En effet, avec la configuration non linéaire des boîtes-S, on a 4 valeurs d'entrée possibles par valeur de sortie.) Ainsi, on en déduit une possible K_{16}.

pour chaque K_{16} possible, calculer $S_i(E(R_{15}*) \oplus K_{16})_{b_x \to b_y}$ et regarder si $P^{-1}(L_{16} \oplus L_{16}*)_{b_x \to b_y} = S_i(E(R_{15}) \oplus K_{16})_{b_x \to b_y} \oplus S_i(E(R_{15}*) \oplus K_{16})_{b_x \to b_y}$.

* l'étape suivante consiste Pour chaque boîte-S, il y a une liste de clés candidates pour chaque chiffré faux qui agit sur la portion de sous clé. Trouver pour chaque boîte-S l'intersection des portions de clés candidates. Il y aura donc 1 portion de clé (sur 6 bits) par boîte-S. Concaténer les 8×6 bits pour obtenir la sous clé K_{16}.

La complexité pour trouver K_{16} correspond à la recherche exhaustive sur la boite-S ce qui donne :
2^6 choix pour les 8 S-Box. Et 32 chiffrés fautés dans notre cas à tester.

$O(32 \times 8 \times 2^4 \times 4)$.
Donc on a une complexité de $O(2^5 \times 2^3 \times 2^4 \times 2^2) = O(2^{14})$ pour trouver K_{16}.

2.2 Question 2

* Après avoir obtenu les différentes portions de 6 bits sur chaque S-box, la clé k16 que j'ai trouvé :
Pour chaque fois on prend la clé trouvé par le S-box et on concatène avec la prochaine clé trouvé.

Clé K de 6bits Choisie dans la SBOX 1 : 11 donc on concat avec l'ancien K

K16 actuelle = 11

Clé K de 6bits Choisie dans la SBOX 2 : 38 donc on concat avec l'ancien K

K16 actuelle = 478

Clé K de 6bits Choisie dans la SBOX 3 : 11 donc on concat avec l'ancien K

K16 actuelle = 11E11

Clé K de 6bits Choisie dans la SBOX 4 : 25 donc on concat avec l'ancien K

K16 actuelle = 478465

Clé K de 6bits Choisie dans la SBOX 5 : 33 donc on concat avec l'ancien K

K16 actuelle = 11E11973

Clé K de 6bits Choisie dans la SBOX 6 : 3E donc on concat avec l'ancien K

K16 actuelle = 478465CFE

Clé K de 6bits Choisie dans la SBOX 7 : 17 donc on concat avec l'ancien K

K16 actuelle = 11E11973F97

Clé K de 6bits Choisie dans la SBOX 8 : 1 donc on concat avec l'ancien K

K16 actuelle = 478465CFE5C1

$K16_{48}(binaire)$= 010001111000010001100101110011111111100101110000001
$K16_{48}(Hexa)$= $478465CFE5C1$

3 III : Retrouver la clé complète du DES

3.1 Question 1

Nous avons obtenu la clé secrète K_{16} qui contient 48 bits. En analysant le schéma de création des 16 sous-clés (de 48 bits chacune) à partir de la clé secrète (de 64 bits avec les 8 bits de parité), on peut en déduire la clé secrète. Cela comporte plusieurs étapes :

* La première étape est effectuer une permutation inverse $PC_2^{-1}(K_{16})$ afin d'obtenir C_{16} et D_{16}. Il est important de noter que lors de cette permutation inverse, 8 bits sont inconnus. En effet, on passe de 48 bits à 56 bits. Il y a donc 8 bits qu'on ne peut déduire à partir de K_{16}.

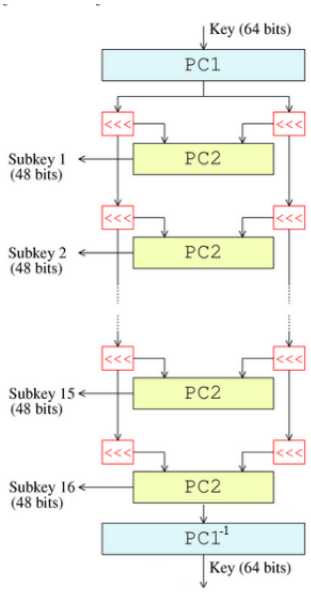
* Létape 2 consiste à déduire C_0 et D_0 à partir de C_{16} et D_{16}.
En effet, $C_0 = C_{16}$ et $D_0 = D_{16}$ puisque la somme des shifts circulaires donne 28 qui correspond à la taille des blocs C_i et D_i.

* L'étape 3 consiste à effectuer une permutation inverse $PC_1^{-1}(C_0||D_0)$ afin d'obtenir la clé finale sur 64 bits. Les 8 bits inconnus sont mélangés dans la clé finale. En plus on aura 8 bits supplémentaires rajoutés correspondant aux bits de parité. Au finale on aura donc une clé de 64 bits.

* L'étape 4 consiste à retrouver les 8 bits inconnus par une recherche exhaustive : 2^8 cas possibles.

* L'étape 5 consiste à déduire les 8 bits de parité en s'assurant que chaque octet possède un nombre impair de bits à 1.

On aura donc une complexité pour trouver K_{16} puis K est de $O(2^{14} + 2^8)O(2^{14})$

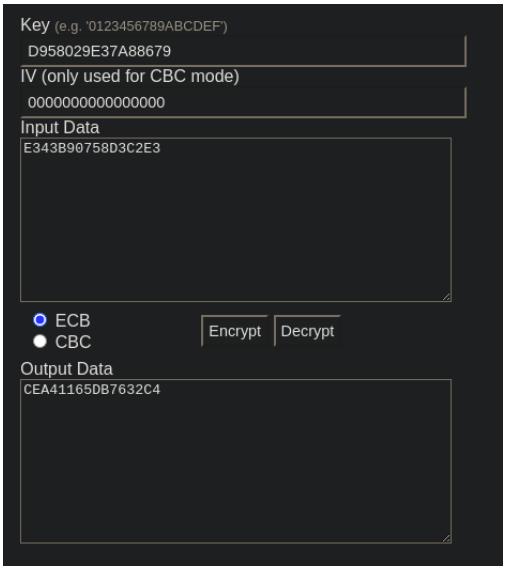


permutation PC2 :

3.2 Question 2

Après avoir effectuer les différentes étapes définis ci-haut voici la clé k obtenu en Binaire puis en Hexadécimale :
K_{64} =11011001 01011000 00000010 10011110 00110111 10101000 10000110 01111001
$K_{64} = D9\ 58\ 02\ 9E\ 37\ A8\ 86\ 79$

Ainsi le resultat obtenu avec le lien fournis :



4 IV : Fautes sur les tours précédents

Faute provoquée sur la valeur de sortie R_{14} du 14^e tour. Dans ce cas on va obtenir l'équation suivante :

$L_{15} = L_{14} \oplus f(K_{15}, R_{14})$ et $R_{15} = R_{14}$ pour le chiffré juste.
$L_{15}* = L_{14} \oplus f(K_{15}, R_{14}*)$ et $R_{15}* = R_{14}*$ pour les chiffrés faux.

En suite on fare :
$P^{-1}(R_{15} \oplus R_{15*}) = S_i(E(R_{14} \oplus K_{15}) \oplus S_i(E(R_{14*} \oplus K_{15})$ ainsi de suite c'est exactement le même principe qu'on l'a fait avec l'attaque précédente mais a chaque fois la complexité diffère jusqu'à ce que l'attaque ne soit plus réaliste.
Or on avait defini que la complexité pour trouvé la clé avec le tour de R_{15} est $O(2^{14})$.
Pour la sortie du R_{14} du 14^e tour on aura une complexité de $O(2^{28})$ et ainsi de suite et chaque tour supplemetaire on multiplie avec une complexite d'un tour qui est $O(2^{14})$ jusqu'à obtenir la complexité que l'attaque soit non réaliste qui est $O(2^{80})$.
Maintenant denifissons les complexité jusqu'à obtenir la complexité que l'attaque soit non réaliste.

Pour le 15^e tour on aura : $O(2^{14})$, cette attaque est réaliste.
Pour le 14^e tour on aura : $O(2^{28})$, cette attaque est réaliste.
Pour le 13^e tour on aura : $O(2^{42})$, cette attaque est réaliste.
Pour le 12^e tour on aura : $O(2^{56})$, cette attaque est réaliste.
Pour le 11^e tour on aura : $O(2^{70})$, cette attaque est réaliste.
Pour le 10^e tour on aura : $O(2^{84})$, cette attaque n'est pas réaliste car on a une complexité qui est pus de 2^{80} .

A partir du 10^e tour l'attaque n'est plus réalisable car la complexité depasse le seuil qui $O(2^{80})$.

5 V : Contre-mesures

Dans ce cas on aura deux grandes solutions possible :
La première solution est de protéger physiquement le support sur lequel le DES est implémenté.
par exemple, poser une matière sur les composants électroniques qui protégerait une injection de fautes par l'utilisation d'un laser. Cette matière pourrait endommager les circuits électronique si quelqu'un tenterait d'enlever cette matière. Cela n'est pas toujours faisable car certains appareils on besoin d'un contact extérieur pour fonctionner (par exemple une carte bancaire).

La deuxième consiste la protection dans l'implémentions elle même.
par exemple refaire le calcul du chiffrement une deuxième fois (ou plus) et comparer le résultat avec ce qui a été obtenu précédemment. Si le résultat n'est pas le même, alors il a été fauté. Cette contre-mesure rallonge le temps de calcul autant de fois qu'on refait le calcul. Cela peut être un problème dans certains cas.