Calcul sécurisé - Contrôle continu

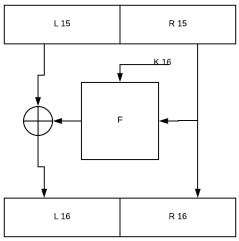
15 mars 2019

Table des matières

1 Question 1

Une attaque par faute contre le DES consiste à introduire une modification sur un bit du chiffré durant l'exécution du chiffrement de manière a compromettre son execution.De ce fait si on dispose d'un chiffré correct et d'un chiffré fauté par l'attaque on peut obtenir des informations sur une partie de la clé utilisée pour le chiffrement. Lors d'une attaque par force brute (recherche exhaustive) sur la clé du DES, la complexité est de 2^{56}. l'objectif de l'attaque par faute est de réduire cette complexitée.

En supposant que l'attaquant est capable d'effectuer une faute sur la valeur de sortie R_{15} du $15_{ème}$ tour une attaque par faute peut être décrite de la façon suivante :



Lors d'une utilisation normal de DES (sans attaque par faute), on obtiendrais les resultats suivant pour L_{16} et R_{16}.

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

Maintenant, si on introduit une faute sur la valeur de sortie R_{15} du $15_{ème}$ tour, on obtient les valeurs suivantes :

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

On a donc une possibilité de retrouver K_{16} en utilisant L_{16} et L^*_{16}.
On va donc utiliser l'operation XOR ($\oplus$) sur L_{16} et L^*_{16} de façon a obtenir :

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

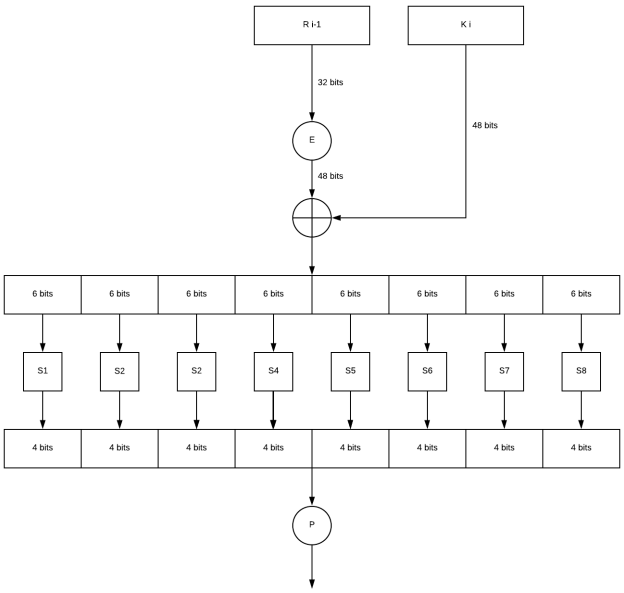
$$= \cancel{L_{15}} \oplus F(R_{15}, K_{16}) \oplus \cancel{L_{15}} \oplus F(R^*_{15}, K_{16})$$

On se retrouve donc avec :

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

Pour continuer l'attaque on va devoir étudier la fonction F plus en détail soit :

F



On constate qie la fonction F prend le bloc R_{i-1} de 32 bits en entrée ainsi que la clé K_i. R_{i-1} passe ensuite par la fonction E qui a pour but d'appliquer une expansion sur le bloc, le passant de 32 à 48 bits.
Après l'expansion, l'opération XOR est appliquée entre R_{i-1} et K_i.
On a donc 48 bits , qui vont être "découpés" en 8 blocs de 6 bits. Chaque bloc va ensuite passer par une S-Box. Les S-Box prennent en entrée 6 bits et en renvoient 4, ce qui permet de ramener la nombre de bit à 32 (soit la taille initiale de R_{i-1}).
Ce bloc de 32 bits va finalement subir une permutation, la sortie de cette permutation étant le resultat renvoyé par la fonction F.
On peut donc écrire l'équation $L_{16} \oplus L_{16}^* = F(R_{15}, K_{16}) \oplus F(R_{15}^*, K_{16})$ de la façon suivante :

$$L_{16} \oplus L_{16}^* = P(S(E(R_{15}) \oplus K_{16})) \oplus P(S(E(R_{15}^*) \oplus K_{16}))$$

On sait également d'après le schéma d'exécution de la fonction F qu'on peut isoler individuellement le résultat de chaque S-box, ce qui permet de transformer cette équation.
$L_{16} \oplus L_{16}^* = F(R_{15}, K_{16}) \oplus F(R_{15}^*, K_{16})$ devient :

$$L_{16} \oplus L_{16}^* =$$

$$P(S_1(E(R_{15}) \oplus K_{16}^{0-5}))||S_2(E(R_{15}) \oplus K_{16}^{6-11}))||...)$$

$$\oplus$$

$$P(S_1(E(R_{15}^*) \oplus K_{16}^{0-5}))||S_2(E(R_{15}^*) \oplus K_{16}^{6-11}))||...)$$

On peut se permettre d'écrire l'équation ainsi car le contenu des différentes S-box est connu, on peut donc à partir des 6 bits d'entrée trouver les bits de sortie correspondants (par ailleurs on peut également retrouver à partir des 4 bits de sortie d'une S-box plusieurs bloc de 6 bits d'entrée possible).

Maintenant on souhaite se débarrasser de P dans notre équation, pour cela il suffit d'appliquer P^{-1} à $L_{16} \oplus L_{16}^*$ soit (possible car la permutation P est connue) :

$$P^{-1}(L_{16} \oplus L_{16}^*) =$$

$$S_1(E(R_{15}) \oplus K_{16}^{0-5}) \oplus S_1(E(R_{15}^*) \oplus K_{16}^{0-5})||S_2(E(R_{15}) \oplus K_{16}^{6-11}) \oplus S_2(E(R_{15}^*) \oplus K_{16}^{6-11})||$$
$$...$$

On peut maintenant mettre en pratique le fait que l'on puisse retrouver les bits d'entrée des 6 box grâce aux bit de sortie pour utiliser cette équation sur les S-box.
On va diviser $P^{-1}(L_{16} \oplus L_{16}^*)$ en 8 blocs de 4 bits. Chaque bloc correspond donc à une sortie de S-box. On aura donc 8 équations :

- $P^{-1}(L_{16} \oplus L_{16}^*)_{0-3} = S_1(E(R_{15}) \oplus K_{16}))_{0-3} \oplus S_1(E(R_{15}^*) \oplus K_{16})_{0-3}$
- $P^{-1}(L_{16} \oplus L_{16}^*)_{4-7} = S_2(E(R_{15}) \oplus K_{16}))_{4-7} \oplus S_2(E(R_{15}^*) \oplus K_{16})_{4-7}$
- ...

K_{16} est la seule valeur inconnue dans ces 8 équations. Chacune de ces équations va nous permettre de retrouver 6 bits de K_{16}, on va donc devoir faire une recherche exhaustive des bits de K_{16} pour chaque S-box afin de retrouver les 48 bits de la sous-clé (6 bits par S-box).
On fera donc 8 recherches de 6 bits pour une complexité de $8 * 2^6$.

2 Question 2

2.1 Décrire précisément ce que vous faites pour retrouver la clé

On a précédemment établi 8 équations qui devrait nous permettre de retrouver K_{16}, chaque équation permettant de trouver 6 bits de la clé rentrant dans la S-box correspondante. On va donc maintenant devoir attaquer chaque S-Box pour retrouver les 6 bits de K_{16}. Cependant comme vu précédemment les 4 bits de sorties d'une S-Box peuvent correspondre à plusieurs entrées de 6 bits différents. On va donc procéder de la façon suivante afin de trouver les 6 bon bits de la clé pour chaque S-Box :

— En premier lieu on va chercher pour chaque S-Box quels sont les chiffrés faux correspondants de la façon suivante :

chiffré juste	1E F4 9F 41 6D D5 57 8A	0001 1110 1111 0100 1001 1111 0100 0001 0110 1101 1101 0101 0101 0111 1000 1010
chiffré faux	1C E5 9F 45 6D D5 57 8E	0001 1100 1110 0101 1001 1111 0100 0101 0110 1101 1101 0101 0101 0111 1000 1110
Permutation IP	58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44, 36, 28, 20, 12, 4, 62, 54, 46, 38, 30, 22, 14, 6, 64, 56, 48, 40, 32, 24, 16, 8, 57, 49, 41, 33, 25, 17, 9, 1, 59, 51, 43, 35, 27, 19, 11, 3, 61, 53, 45, 37, 29, 21, 13, 5, 63, 55, 47, 39, 31, 23, 15, 7	
	L16	R16
chiffré juste permuté	0111 1010 0110 0111 0111 0111 0111 1100	1010 0110 0001 0010 1001 0101 1100 0101
	L16*	R16*
chiffré faux permuté	0111 1010 0110 0101 1111 1111 0111 1110	1010 0110 0001 0010 1001 0101 1100 0100
L16 XOR L16*	0000 0000 0000 0010 1000 1000 0000 0010	
P-1	0101 0000 0000 0000 0000 0000 0000 0011	

— On constate ici au résultat de $L_{16} \oplus L^*_{16}$ que le 1^{er} et le $8^{ème}$ blocs de 4 bits sont différents de 0. On en déduit donc que ce chiffré peut être utilisé pour l'attaque des S-Box 1 et 8.
— La partie précédente nous à permis de définir 8 équations de cette forme :

$$P^{-1}(L_{16} \oplus L^*_{16})_{0-3} = S_1(E(R_{15}) \oplus K_{16})_{0-3} \oplus S_1(E(R^*_{15}) \oplus K_{16})_{0-3}...$$

On a vu comment identifier quels chiffrés faux utiliser contre quels S-BOX, on va à partir de la devoir trouver tout les couples de 6 bits possibles $R_{15} \oplus K_{16}$ et $R^*_{15} \oplus K_{16}$ tel que :

$$S(E(R_{15}) \oplus K_{16})_{i-i+3} \oplus S(E(R^*_{15}) \oplus K_{16})_{i-i+3} = P^{-1}(L_{16} \oplus L^*_{16})_{i-i+3}$$

On va ensuite devoir isoler K_{16}.
Pour ce faire on va :

— Récupérer le chiffré juste C. On notera
$L_{16} = IP(C)_{0-31}$ et $R_{15} = R_{16} = IP(C)_{32-62}$

— On récupère un chiffré faux FC utilisable sur la s-box a attaquer. On notera
$L^*_{16} = IP(FC)_{0-31}$ et $R^*_{15} = R^*_{16} = IP(FC)_{32-62}$

— Chacune de ces variables est connue est fait 32 bits.
On va donc prendre en référence $P^{-1}(L_{16} \oplus L^*_{16})$
O va également appliquer une expansion E (présente dans la fonction F vue précédemment) à R_{15} et R^*_{15} tel que $E\_R_{15} = E(R_{15})$ et $E\_R^*_{15} = E(R^*_{15})$ avec E_R_{15} et $E\_R^*_{15}$ de 48 bits.

— On effectue ensuite une recherche exhaustive de K_{16}.
On note $Ctmp = E\_R_{15} \oplus K_{16}$
et $FCtmp = E\_R^*_{15} \oplus K_{16}$, chacune de ces 2 variables composée de 48 bits.

— Maintenant, on va récupérer 6 bits correspondant à la S-box que l'on souhaite attaquer (les 6 premiers si on souhaite attaquer S_1 et ainsi de suite).
— On utilise ces 6 bits sur la S-Box à attaquer avec Ctmp et FCtmp. Puis on appliquer un XOR sur le resultat obtenue avec Ctmp et FCtmp sur la S-Box.

Si le résultat du XOR correspond à $P^{-1}(L_{16} \oplus L_{16}^*)$ alors on saura que les 6 bits de K_{16} utilisé pour créer Ctmp et FCtmp sont une solutions possible.

— Enfin on réitère ce processus avec plusieurs chiffrés différents sur une même S-box (les chiffrés identifiés précédemment comme étant utilisable sur cette S-Box).
On aura donc plusieurs solutions de 6 bits possibles pour K_{16}. Les 6 bon bits a conservé seront ceux communs a chaque attaque effectuée sur la S-Box.

— De cette façon on a réussi a identifier 6 bits de K_{16} grâce à une S-Box. Il nous suffit donc d'attaquer les 7 autres de la même manière pour réussir à obtenir 8 * 6 bits = 48 bits soit K_{16}.

2.2 Donnez les 48 bits de clé que vous obtenez grâce à cette attaque par fautes

On à pu, grâce à l'attaque décrite précédement identifier K_{16} comme étant :

— directement en sortie du programme :

$$b|8|3a|21|??|d|2a$$

— converti en binaire :

$$001011|001000|111110|110110|100001|?|001101|101010$$

— converti en hexadécimal :

$$?$$

3 Question 3

3.1 Expliquer comment on peut retrouver les 8 bits manquants

3.2 Faites-le, et donner ainsi la valeur complète de la clé qui vous a été assignée.

4 Question 4

5 Question 5