Chapitre II Data Encryption Standard (DES)

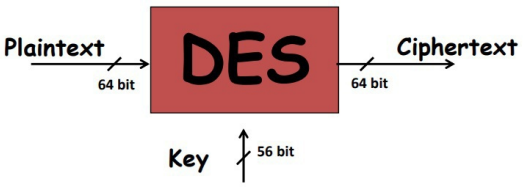


Table des matières

Introduction à DES

- DES est un algorithme de chiffrement par bloque. Taille du bloque chiffré = 64bits, taille de la clé *primaire* K = 56 bits(2^{56} valeurs possibles de la clé) ;
- Inventé par IBM avec NSA (National Security Agency) et standardisé en 1977.
  Cassé par attaque brute force en 2000 avec l'implémentation hardware Copacobana.

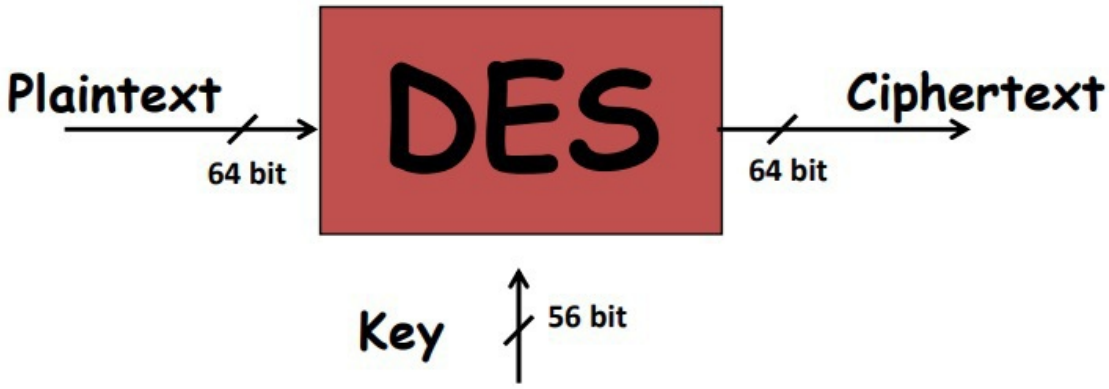


Figure 1. DES

Fonction de chiffrement DES

II

DES se compose des étapes suivantes :

- Permutation initiale
- 16 rounds de chiffrement
- Permutation finale (inverse de la permutation initiale)

- $P(x) = L_0R_0$
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $y = IP^{-1}(R_{16}L_{16})$

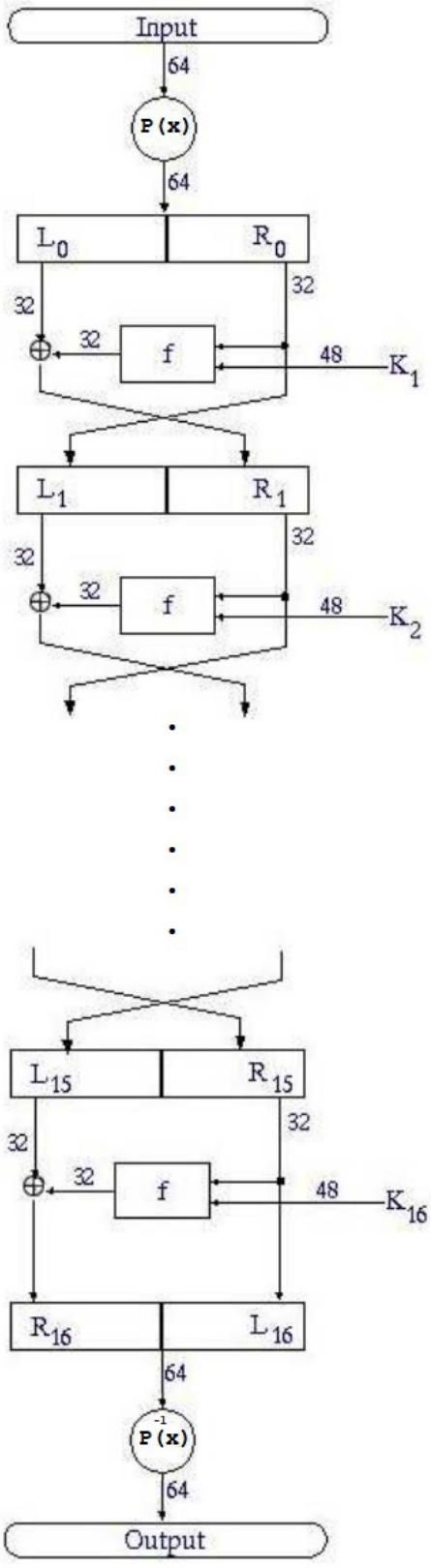


Figure 2. Fonctionnement global de DES

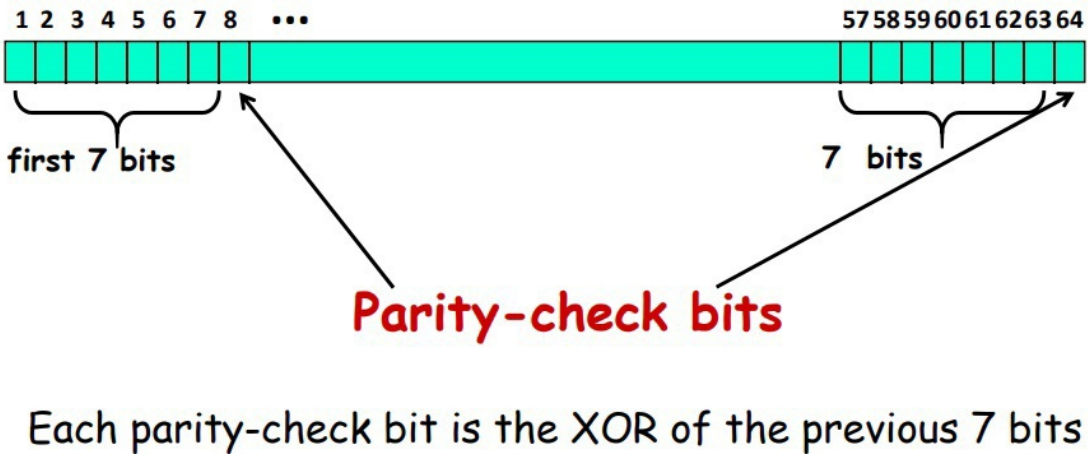


Figure 3. Parity check dans les clé DES

- Avant de commencer la procédure de chiffrement, la première phase qui doit être exécutée consiste à générer 16 sous-clés (sub-keys) depuis la clé primaire. La taille de chaque clé est de 48bits (voir la section génération de clés). Dans plusieurs références, la clé primaire est d'une taille de 64 bits, mais dans chaque octet de la clé, le 8em bit n'est pas utilisé car celui-ci comporte la valeur de vérification de parité.
- Pour commencer le traitement d'un bloque de texte de 64 bits, celui ci est introduit à une fonction de permutation P(X) [où X est le bloque de bits]. Cette phase consiste simplement à changer les positions des bits du texte claire.
  Par exemple : Comme montré dans la figure 4, le bit numéro 1 sera déplacé vers la position 40 et le bit numéro 2 à la position 8.

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

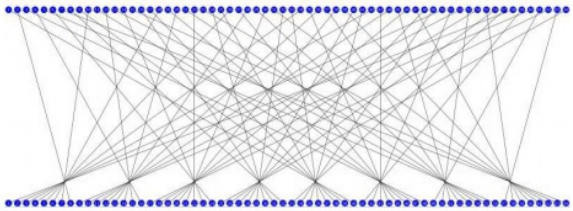


Figure 4. Permutation initiale

La permutation finale est exactement l'inverse de cette permutation initial :

Par exemple : Comme montré dans la figure 5, le bit numéro 40 sera déplacé vers la position 1 et le bit numéro 8 à la position 2.

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

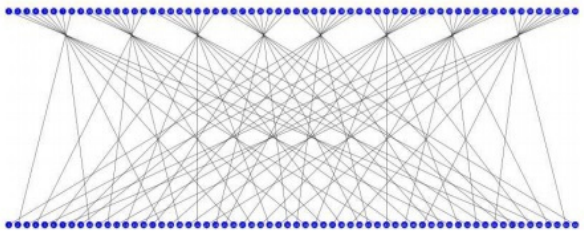


Figure 5. Permutation finale

- Après la permutation initiale, le bloque de 64 bit est divisé en deux bloques de 32 bits (L_0, R_0).
- A chaque round i , $R_i = L_{i-1}$ XOR $F(R_{i-1},K_i)$ et $L_i = R_{i-1}$(La partie droite du round précédant est copiée dans la partie gauche, et la partie gauche est chiffrée avec le résultat de la fonction F).
- A la fin des 16 rounds, L_{16} et R_{16} sont inversées : $L_{16} = L_{15}$ XOR $F(R_{15},K_{16})$ et $R_{16} = R_{15}$

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

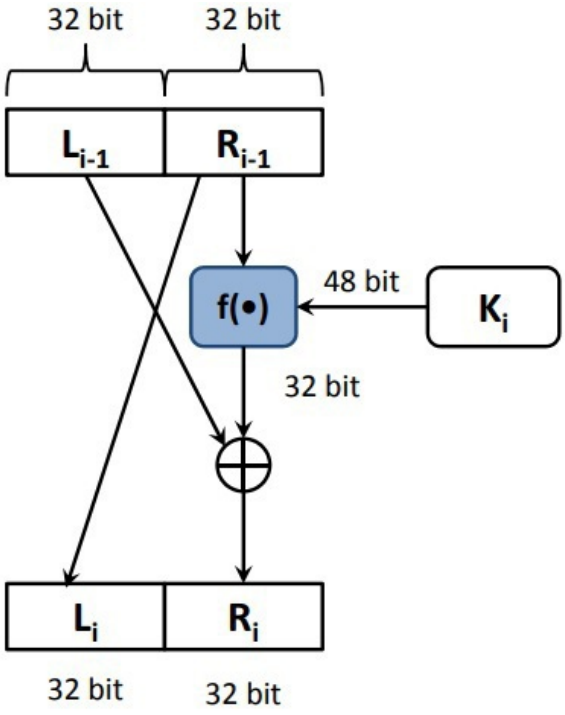
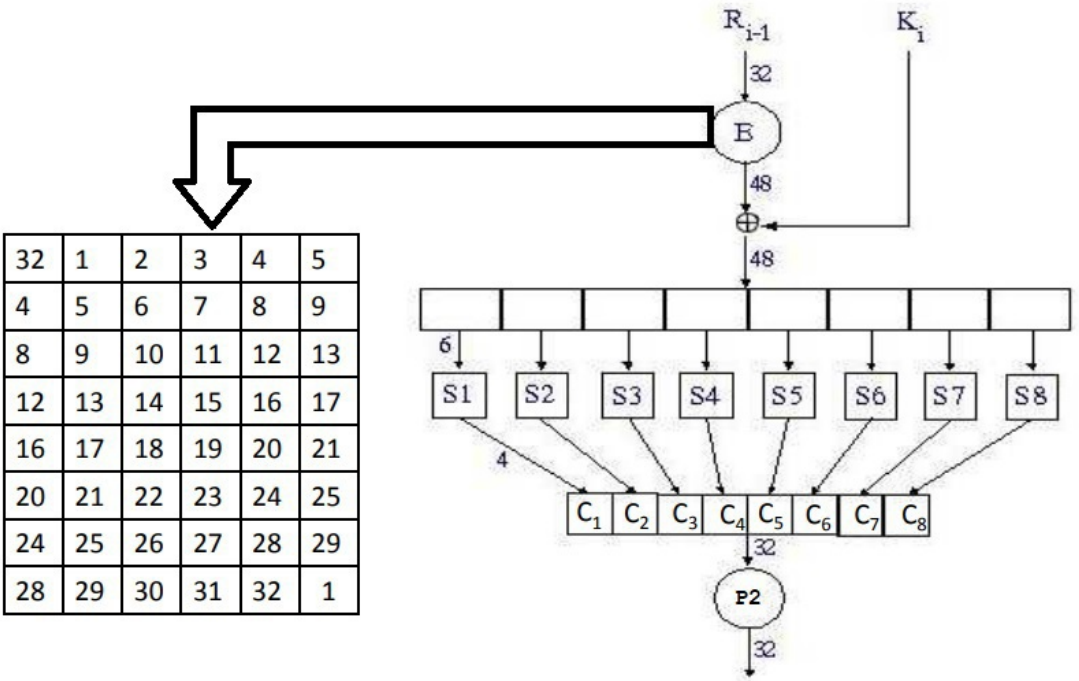


Figure 6. Round i de DES

1. La fonction F



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

Figure 6. Fonction F

- La fonction F de DES est la partie la plus complexe de ce système. Cette fonction prend en paramètre , K_i et R_{i-1}.
- La première étape de cette fonction consiste à étendre le bloque R_{i-1} de 32 bits à 48 bits (*pour que le XOR avec la clé dérivé K_i soit possible*). Pour faire ce-ci, le bloque de 32 bits est divisé en 8 bloques de 4 bits. Par la suite, de chaque bloque de 4bits, le bit le plus à droit et le bit le plus à gauche sont dupliqués. Ainsi, 2 de chaque 4 bits sont dupliqués et la taille devient de 32 + 32 /2 = 48 bits (*voir le tableau à gauche pour les positions des bits dupliquées*).
- Après l'extension de la clé, la valeur en résultat est combinée avec la valeur de la clé du round par un XOR.

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

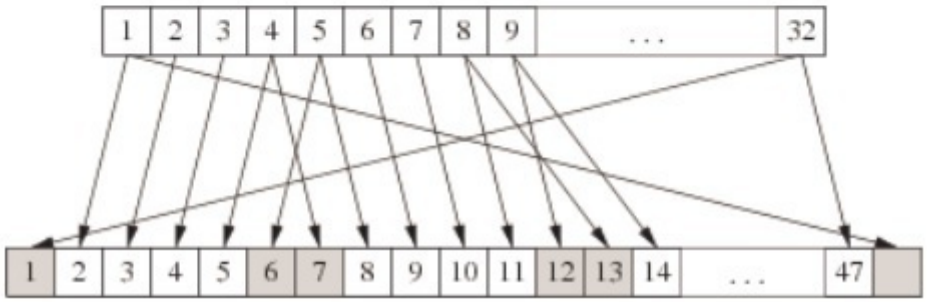


Figure 7. Fonction d'extension

- Le bloque devient de taille 48 bits (8 bloques de 6 bits). Par la suite, chaque bloque de 6bits est passé en paramètre à l'entré du Sbox correspondant (le premier bloque de 6bits sera traité par Sbox1, le deuxième bloque sera passé au Sbox 2, etc). Il existe 8 différents Sbox , chacune prend 6 bits en entré et renvoie 4 bits en sortie.

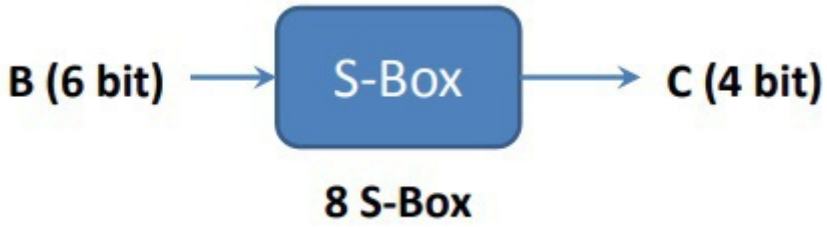


Figure 8. SBox

- Chaque Sbox est une simple table de correspondance de 4 lignes et 16 colonnes. Pour obtenir la sortie correspondante à une entré de 6bits :

  1. Le premier et le dernier bit sont pris pour obtenir l'indice de la ligne
  2. Les 4 bits au milieu sont pris pour obtenir la valeur de la colonne
  3. La valeur dans la case est renvoyé en sortie

Par exemple pour B=101111 (indice ligne= 11 = 3, indice colonne=0111= 7). La sortie correspondante

pour un tel bloque de 6bits = 7

S_1	1	2	3	...	7								15 Colonne
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
Ligne																

S(i, j) <16

Figure 8. Sbox 1

- Par la suite les sorties de chaque Sbox sont regroupées (8 x 4 bits = 32 bits !) dans un seul bloque.
- La dernière étape de l'exécution d'un round consiste à une permutation fixe sur le bloque de 32 bits.

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

Figure 9. Permutation finale du round

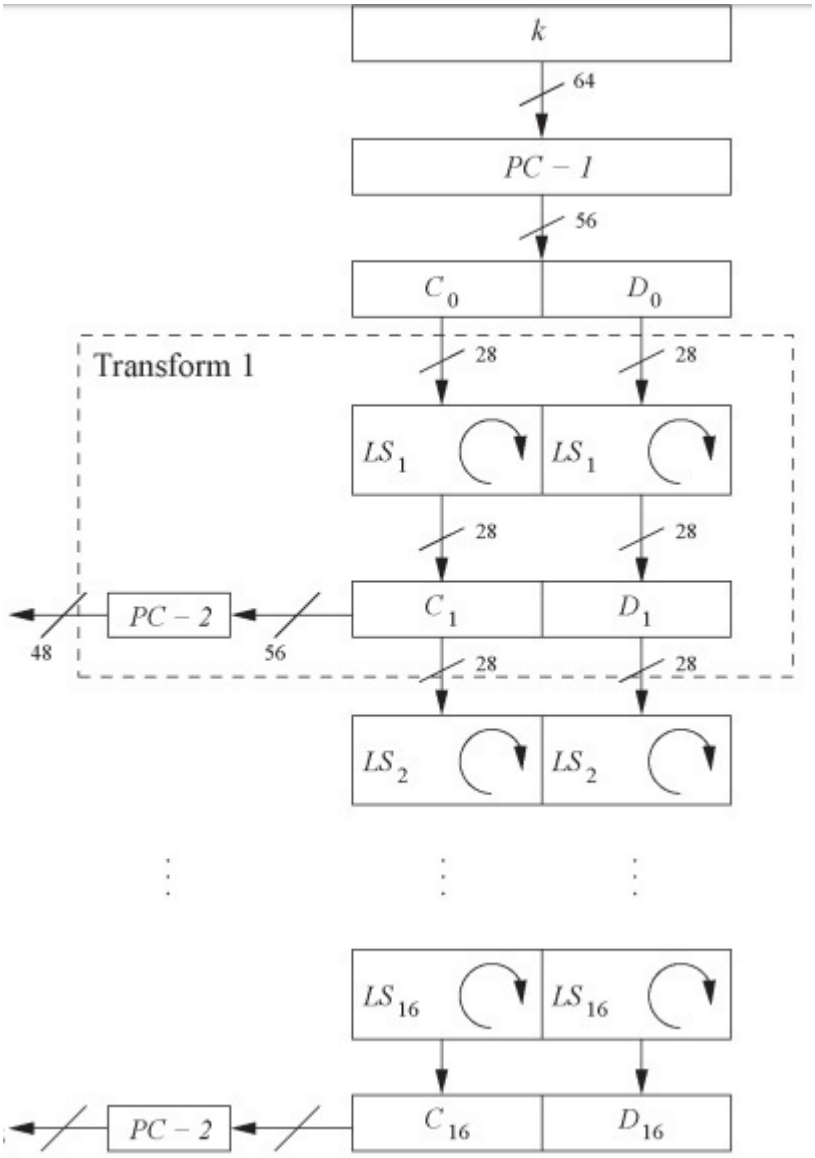
Fondamental

La fonction d'extension, Sbox et la permutation du round assurent la propriété *Diffusion* (voir chapitre 1). Ces trois fonctionnalités assurent qu'après 5 rounds, le changement d'un seul bit dans le texte clair affectera tout le bloque)

Génération de clés

Pour le chiffrement et déchiffrement, les 16 clés dérivées doivent être générés en premier (une clé pour chaque round).

1. La première étape (*PC -1 = Permuted Choice*) consiste à enlever les 8 bits de parité de la clé primaire de 64 bits et de permuter les positions des 56 bits restants. Cette table de sélection est définie d'une manière fixe.
2. Le bloque de 56 bits en résultat est divisé en deux parties de 28 bits C_0 et D_0.
3. A chaque itération les deux parties de la clé sont décalés par un ou deux bits à gauche (décalés par un bit dans round 1,2,9, et 16, décalés par 2 bits dans tous les autres itérations)
4. Après la fin de décalage, la clé est passée en entré de (*PC -2 = Permuted Choice*) qui prend 48 bits des 56 bit ($C_i + D_i$). Cette table de sélection est fixe comme PC1



Left						
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
Right						
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

PC 1 (Permuted Choice 1)

PC 2 (Permuted Choice 2)

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