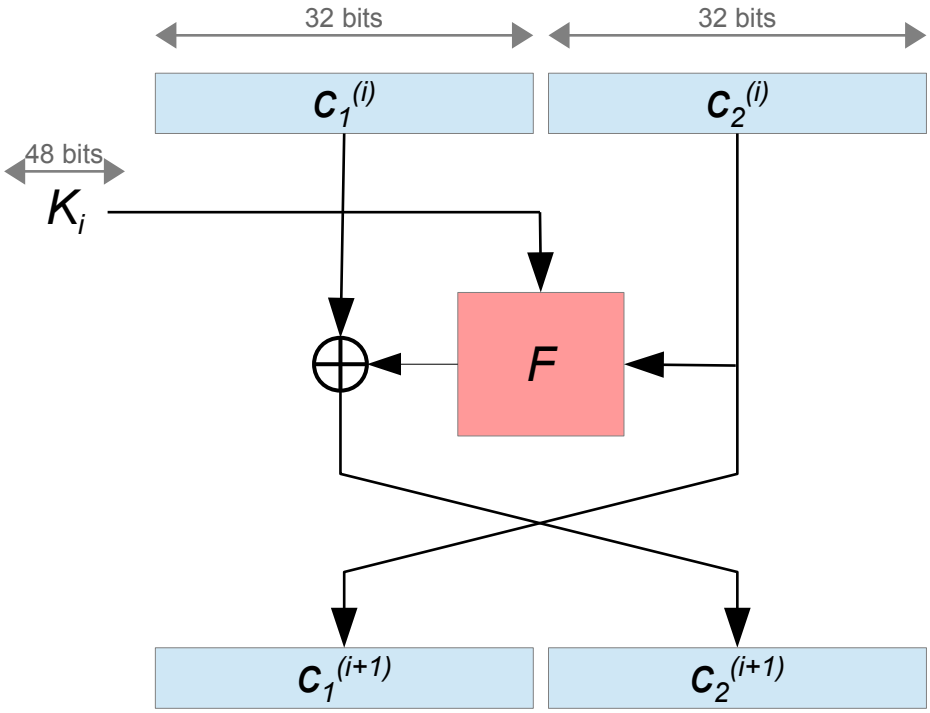
Description du chiffrement DES

Schéma général – Schéma de Feistel & Fonction F :

- L'algorithme DES chiffre des blocs de 64 bits
- Il utilise une clé *K* de 56+8 bits (8 groupes de 7 bits + 1 bit de parité).
- Il exécute sur chacun des blocs de 64 bits, 16 rondes d'un schéma de Feistel.
- Chaque ronde utilise une clé K_i partielle de 48 bits calculée à partir de la clé principale *K*.

Rappel : une ronde du schéma de Feistel



- Avant le schéma de Feistel, une permutation *IP* est appliquée sur le bloc de 64 bits :

IP =

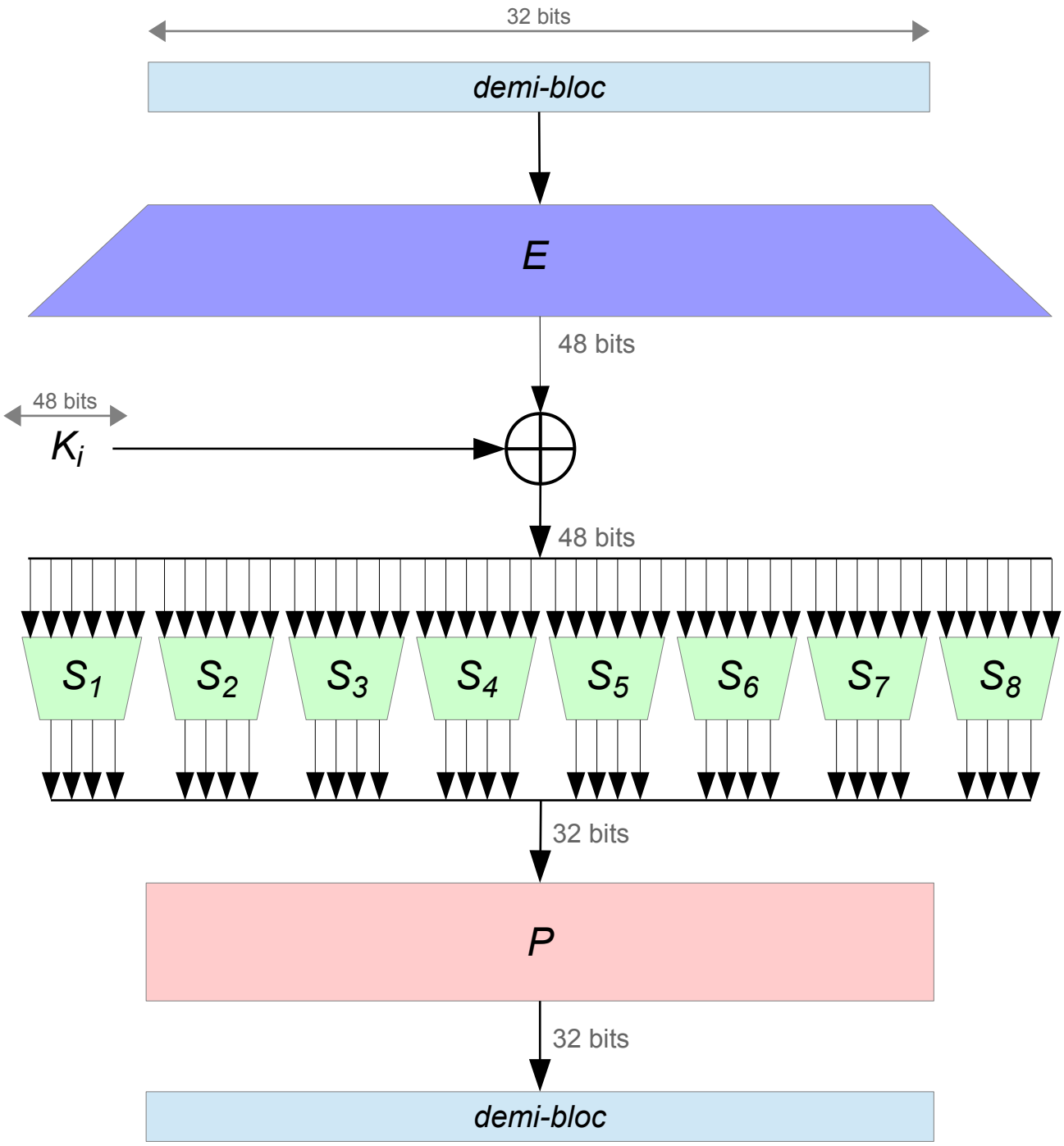
58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

- Après le schéma de Feistel, la permutation IP^{-1} est appliquée sur le bloc de 64 bits :

IP^{-1} =

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

- Description de la fonction *F* qui traite un demi bloc de 32 bits avec une clé de 48 bits :



- *E* est une fonction d'expansion (faisant passer le bloc de 32 bits à 48 bits). Les 32 bits $b_1 \ldots b_{32}$ sont répétés de la manière suivante :

$$
E = \begin{matrix}
b_{32} & b_1 & b_2 & b_3 & b_4 & b_5 \\
b_4 & b_5 & b_6 & b_7 & b_8 & b_9 \\
b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} \\
b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} \\
b_{16} & b_{17} & b_{18} & b_{19} & b_{20} & b_{21} \\
b_{20} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\
b_{24} & b_{25} & b_{26} & b_{27} & b_{28} & b_{29} \\
b_{28} & b_{29} & b_{30} & b_{31} & b_{32} & b_1
\end{matrix}
$$

Les boites de substitution – S-box :

- Les boites S_1 à S_8 sont des boites de substitution.

  Les bits sont pris par groupe de 6 : $b_1b_2b_3b_4b_5b_6$.

  Pour chaque S-box, il existe un tableau à 4 lignes et 16 colonnes que l'on lit ainsi : à chaque entrée $b_1b_2b_3b_4b_5b_6$, on associe la valeur contenue dans la case de la ligne b_1b_6 et de la colonne $b_2b_3b_4b_5$.

  Par exemple, pour déterminer S_1(**011011**) il faut lire la ligne **01** et la colonne **1101** du tableau S-box 1 ; c'est-à-dire **0101**.

- S-box 1 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	**1101**	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	**0101**	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

- S-box 2 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1111	0001	1000	1110	0110	1011	0011	0100	1001	0111	0010	1101	1100	0000	0101	1010
01	0011	1101	0100	0111	1111	0010	1000	1110	1100	0000	0001	1010	0110	1001	1011	0101
10	0000	1110	0111	1011	1010	0100	1101	0001	0101	1000	1100	0110	1001	0011	0010	1111
11	1101	1000	1010	0001	0011	1111	0100	0010	1011	0110	0111	1100	0000	0101	1110	1001

- S-box 3 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1010	0000	1001	1110	0110	0011	1111	0101	0001	1101	1100	0111	1011	0100	0010	1000
01	1101	0111	0000	1001	0011	0100	0110	1010	0010	1000	0101	1110	1100	1011	1111	0001
10	1101	0110	0100	1001	1000	1111	0011	0000	1011	0001	0010	1100	0101	1010	1110	0111
11	0001	1010	1101	0000	0110	1001	1000	0111	0100	1111	1110	0011	1011	0101	0010	1100

- S-box 4 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0111	1101	1110	0011	0000	0110	1001	1010	0001	0010	1000	0101	1011	1100	0100	1111
01	1101	1000	1011	0101	0110	1111	0000	0011	0100	0111	0010	1100	0001	1010	1110	1001
10	1010	0110	1001	0000	1100	1011	0111	1101	1111	0001	0011	1110	0101	0010	1000	0100
11	0011	1111	0000	0110	1010	0001	1101	1000	1001	0100	0101	1011	1100	0111	0010	1110

- S-box 5 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

- S-box 6 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1100	0001	1010	1111	1001	0010	0110	1000	0000	1101	0011	0100	1110	0111	0101	1011
01	1010	1111	0100	0010	0111	1100	1001	0101	0110	0001	1101	1110	0000	1011	0011	1000
10	1001	1110	1111	0101	0010	1000	1100	0011	0111	0000	0100	1010	0001	1101	1011	0110
11	0100	0011	0010	1100	1001	0101	1111	1010	1011	1110	0001	0111	0110	0000	1000	1101

- S-box 7 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0100	1011	0010	1110	1111	0000	1000	1101	0011	1100	1001	0111	0101	1010	0110	0001
01	1101	0000	1011	0111	0100	1001	0001	1010	1110	0011	0101	1100	0010	1111	1000	0110
10	0001	0100	1011	1101	1100	0011	0111	1110	1010	1111	0110	1000	0000	0101	1001	0010
11	0110	1011	1101	1000	0001	0100	1010	0111	1001	0101	0000	1111	1110	0010	0011	1100

- S-box 8 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1101	0010	1000	0100	0110	1111	1011	0001	1010	1001	0011	1110	0101	0000	1100	0111
01	0001	1111	1101	1000	1010	0011	0111	0100	1100	0101	0110	1011	0000	1110	1001	0010
10	0111	1011	0100	0001	1001	1100	1110	0010	0000	0110	1010	1101	1111	0011	0101	1000
11	0010	0001	1110	0111	0100	1010	1000	1101	1111	1100	1001	0000	0011	0101	0110	1011

- *P* est la permutation suivante :

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

Générateur des clés partielles :

- Chaque ronde du réseau de Feistel utilise une clé K_i calculée à partir d'une clé utilisateur de 56 bits + 8 bits de parité. Pour construire ces clés partielles, deux permutations sont nécessaires ainsi que des rotations de un ou deux bits vers la gauche :

  PC-1 est une permutation/compression sur 64 bits qui retourne 56 bits :

  PC-1 = 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
  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

  Les positions 8, 16, 24, 32, 40, 48, 56 et 64 sont ignorés (ce sont les bits de parité).

  PC-2 est une permutation/compression sur 56 bits qui retourne 48 bits :

  PC-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

  Les positions 9, 18, 22, 25, 35, 38, 43 et 54 sont ignorés.

  R_1, R_2, R_9 et R_{16} sont des rotations de un bit vers la gauche (*left shift x1*) sur des vecteurs de 28 bits.

  R_3, R_4, R_5, R_6, R_7, R_8, R_{10} , R_{11}, R_{12}, R_{13}, R_{14} et R_{15} sont des rotations de deux bits vers la gauche (*left shift x2*) sur des vecteurs de 28 bits.

- Finalement, le schéma suivant décrit la génération des clés partielles K_i :