1. (a)

$$\begin{array}{r} \text{keystream: } 10010100 \\ \underline{\text{plaintext: } 01100001} \\ \text{stream cipher: } 11110101 \end{array}$$

(b)

$$\begin{array}{r} \text{keystream: } 10010100 \\ \underline{\text{plaintext: } 01110000} \\ \text{stream cipher: } 11100100 \end{array}$$

(c)

$$\begin{array}{r} \text{keystream: } 10010100 \\ \underline{\text{plaintext: } 01110011} \\ \text{stream cipher: } 11100111 \end{array}$$

(d)

$$\begin{array}{r} \text{keystream: } 10010100 \\ \underline{\text{plaintext: } 01110101} \\ \text{stream cipher: } 11100001 \end{array}$$

2. (a)

$$\begin{aligned} S_1(x_1 = 000000) \oplus S_1(x_2 = 000001) &\neq S_1(x_1 = 000000 \oplus x_2 = 000001) \\ 14_{(10)} \oplus 00_{(10)} &\neq S_1(000001) \\ 1110_{(2)} \oplus 0000_{(2)} &\neq 00_{(10)} \\ 1110_{(2)} &\neq 0000_{(2)} \end{aligned}$$

(b)

$$\begin{aligned} S_1(x_1 = 111111) \oplus S_1(x_2 = 100000) &\neq S_1(x_1 = 111111 \oplus x_2 = 100000) \\ 13_{(10)} \oplus 04_{(10)} &\neq S_1(011111) \\ 1101_{(2)} \oplus 0100_{(2)} &\neq 08_{(10)} \\ 1001_{(2)} &\neq 1000_{(2)} \end{aligned}$$

(c)

$$\begin{aligned} S_1(x_1 = 101010) \oplus S_1(x_2 = 010101) &\neq S_1(x_1 = 101010 \oplus x_2 = 010101) \\ 06_{(10)} \oplus 12_{(10)} &\neq S_1(111111) \\ 0110_{(2)} \oplus 1100_{(2)} &\neq 13_{(10)} \\ 1010_{(2)} &\neq 1101_{(2)} \end{aligned}$$

3. (a) Showing my steps:

i.

plaintext (64-bit)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

key (64-bit)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

ii.

plaintext after IP (64-bit)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

iii.

Key after $PC - 1$ (56-bit)

C_0 (28-bit)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0				

D_0 (28-bit)

				0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

iv.

Since this is round 1, rotate both halves LEFT one bit

$$\begin{array}{c} C_0 \text{ (28-bit)} \\ \begin{array}{cccccccc} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&&&& \end{array} \end{array} \qquad \begin{array}{c} D_0 \text{ (28-bit)} \\ \begin{array}{cccccccc} &&&&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{array} \end{array}$$

v.

Stack the halves such that C_0 is on top of D_0. Permute $PC-2$

$$\begin{array}{c} C_0, D_0 \text{ (56-bit)} \\ \begin{array}{cccccccc} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{array} \end{array} \overset{PC-2}{\Longrightarrow} \begin{array}{c} k_1 \text{ (48-bit)} \\ \begin{array}{cccccc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{array} \end{array}$$

vi.

Creating the f-Function.

A. Take R_0 (32-bit) and expansion permute E

$$\begin{array}{c} E_1(R_0) \text{ (48-bit)} \\ \begin{array}{cccccc} 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \\ 0&0&0&0&0&0 \end{array} \end{array}$$

B. Compute $E_1(R_0) \oplus k_1$ to obtain another 48-bit

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

C. Process each row r_i of matrix in previous step in S_i for another 32-bit

$$\begin{aligned}
S_1 &= 1110 \\
S_2 &= 1111 \\
S_3 &= 1010 \\
S_4 &= 0111 \\
S_5 &= 0010 \\
S_6 &= 1100 \\
S_7 &= 0100 \\
S_8 &= 1101
\end{aligned}$$

D. Send previous step's 32-bit to permutation P for f-Function (32-bit)

$$\begin{array}{c}
S \text{ box results (32-bit)} \\
\begin{matrix}
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 1
\end{matrix}
\end{array}
\xrightarrow{P}
\begin{array}{c}
f \text{ (32-bit)} \\
\begin{matrix}
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0
\end{matrix}
\end{array}$$

vii.

$$\begin{array}{c} L_0 \text{ (32-bit)} \\ \begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \end{array} \oplus \begin{array}{c} f \text{ (32-bit)} \\ \begin{matrix} 1&1&0&1&1&1&0&0 \\ 1&1&0&1&1&0&0&0 \\ 1&1&0&1&1&0&1&1 \\ 1&0&1&1&1&1&0&0 \end{matrix} \end{array}$$

The resulting operation is R_1

$$\begin{array}{c} R_1 \text{ (32-bit)} \\ \begin{matrix} 1&1&0&1&1&1&0&0 \\ 1&1&0&1&1&0&0&0 \\ 1&1&0&1&1&0&1&1 \\ 1&0&1&1&1&1&0&0 \end{matrix} \end{array}$$

viii.

Thus, our output after the first round is

$$\begin{array}{c} L_1 \text{ (32-bit)} \\ \begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \end{array} \qquad \begin{array}{c} R_1 \text{ (32-bit)} \\ \begin{matrix} 1&1&0&1&1&1&0&0 \\ 1&1&0&1&1&0&0&0 \\ 1&1&0&1&1&0&1&1 \\ 1&0&1&1&1&1&0&0 \end{matrix} \end{array}$$

(b) Showing my steps:

i.

$$\begin{array}{c} \text{plaintext (64-bit)} \\ \begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \end{matrix} \end{array} \qquad \begin{array}{c} \text{key (64-bit)} \\ \begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \end{array}$$

ii.

$$\begin{array}{cccccccc}
\multicolumn{8}{c}{\text{plaintext after } IP \text{ (64-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

iii.

Key after $PC-1$ (56-bit)

$$\begin{array}{cccccccc}
\multicolumn{8}{c}{C_0 \text{ (28-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & & &
\end{array}
\qquad
\begin{array}{cccccccc}
\multicolumn{8}{c}{D_0 \text{ (28-bit)}} \\
 & & & & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

iv.

Since this is round 1, rotate both halves LEFT one bit

$$\begin{array}{cccccccc}
\multicolumn{8}{c}{C_0 \text{ (28-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & & & &
\end{array}
\qquad
\begin{array}{cccccccc}
\multicolumn{8}{c}{D_0 \text{ (28-bit)}} \\
 & & & & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

v.

Stack the halves such that C_0 is on top of D_0. Permute $PC-2$

$$\begin{array}{cccccccc}
\multicolumn{8}{c}{C_0, D_0 \text{ (56-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\xRightarrow{PC-2}
\begin{array}{cccccc}
\multicolumn{6}{c}{k_1 \text{ (48-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

vi.

Creating the f-Function.

A. Take R_0 (32-bit) and expansion permute E

$$\begin{array}{cccccccc}
\multicolumn{8}{c}{R_0 \text{ (32-bit)}} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\overset{E}{\Longrightarrow}
\begin{array}{cccccc}
\multicolumn{6}{c}{E_1(R_0) \text{ (48-bit)}} \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

B. Compute $E_1(R_0) \oplus k_1$ to obtain another 48-bit

$$\begin{array}{cccccc}
\multicolumn{6}{c}{E_1(R_0) \text{ (48-bit)}} \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array}
\oplus
\begin{array}{cccccc}
\multicolumn{6}{c}{k_1 \text{ (48-bit)}} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

The resulting 48-bits are

$$\begin{array}{cccccc}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

C. Process each row r_i of matrix in previous step in S_i for another 32-bit

$$\begin{aligned} S_1 &= 0011 \\ S_2 &= 1111 \\ S_3 &= 1010 \\ S_4 &= 0111 \\ S_5 &= 0010 \\ S_6 &= 1100 \\ S_7 &= 0100 \\ S_8 &= 0001 \end{aligned}$$

D. Send previous step's 32-bit to permutation P for f-Function (32-bit)

$$\begin{array}{c} S \text{ box results (32-bit)} \\ \begin{array}{cccc} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \end{array} \xRightarrow{P} \begin{array}{c} f \text{ (32-bit)} \\ \begin{array}{cccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \end{array}$$

vii.

$$\begin{array}{c} L_0 \text{ (32-bit)} \\ \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \end{array} \oplus \begin{array}{c} f \text{ (32-bit)} \\ \begin{array}{cccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \end{array}$$

The resulting operation is R_1

$$\begin{array}{c} R_1 \text{ (32-bit)} \\ \begin{array}{cccccccc} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array} \end{array}$$

viii.

Thus, our output after the first round is

L_1 (32-bit)

1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

R_1 (32-bit)

1	1	0	1	0	1	0	0
0	1	0	1	1	0	0	0
0	1	0	1	1	0	1	1
1	0	0	1	1	1	1	0

There are 6 different bits compared to part (a). The bits are different at positions $L_1 = 1$ and $R_1 = 5, 9, 17, 27, 31$.

4. (a)

initial key k_0 (64-bit)

1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

After $PC - 1$,

k_0 $PC - 1$ (56-bit)

0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Split into halves (28-bit)

C_0 (28-bit)

0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0				

D_0 (28-bit)

				0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Execute LS_1; 1 bit

C_1 (28-bit)								D_1 (28-bit)							
0	0	0	0	0	0	1	0					0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0					0	0	0	0	0	0	0	0

Execute LS_2; 1 bit

C_2 (28-bit)								D_2 (28-bit)							
0	0	0	0	0	1	0	0					0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0					0	0	0	0	0	0	0	0

Execute LS_3; 2 bit

C_3 (28-bit)								D_3 (28-bit)							
0	0	0	1	0	0	0	0					0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0					0	0	0	0	0	0	0	0

Execute $LS_{4,5,\ldots,16}$ such that rounds $i = 1, 2, 9, 16$ shift left 1 bit, else shift 2 bits. The following bits are 1 for each k_i:

$$\begin{aligned} k_1 &= 7 & k_9 &= 21 \\ k_2 &= 6 & k_{10} &= 19 \\ k_3 &= 4 & k_{11} &= 17 \\ k_4 &= 2 & k_{12} &= 15 \\ k_5 &= 28 & k_{13} &= 13 \\ k_6 &= 26 & k_{14} &= 11 \\ k_7 &= 24 & k_{15} &= 9 \\ k_8 &= 22 & k_{16} &= 8 \end{aligned}$$

(b) Observing Table for $PC - 2$, we have

$PC - 2$ (48-bit)

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

$\Rightarrow$

$PC - 2$ (48-bit)

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

With the table rearranged, it is a bit easier to see which S_i Box is affected each k_i,

$$\begin{aligned}
k_1 &= S_4 & k_9 &= S_2 \\
k_2 &= S_2 & k_{10} &= S_3 \\
k_3 &= S_3 & k_{11} &= S_1 \\
k_4 &= S_4 & k_{12} &= S_2 \\
k_5 &= S_2 & k_{13} &= S_4 \\
k_6 &= S_3 & k_{14} &= S_1 \\
k_7 &= S_1 & k_{15} &= \text{bit-9 does not carry over} \\
k_8 &= \text{bit-22 does not carry over} & k_{16} &= S_3
\end{aligned}$$

Observation: Since the keys only changed in C_i, only the first four S Boxes are affected. For bits that a lost during $PC-2$, I assume this does not have an effect on the Boxes.