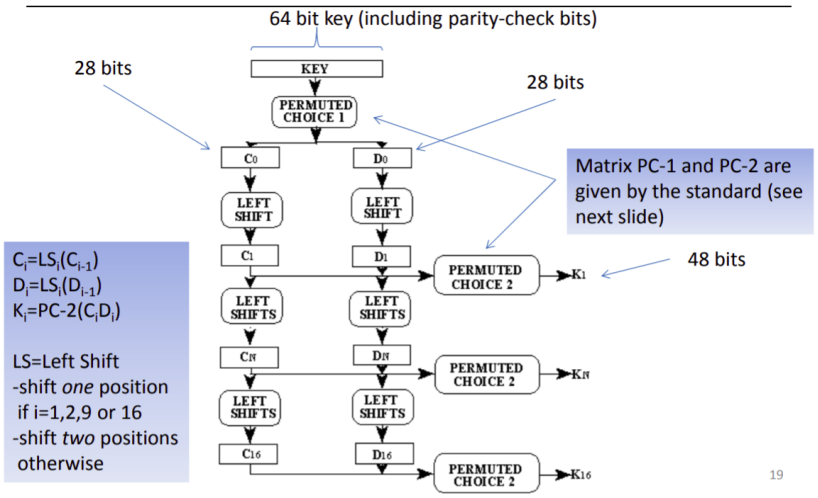
Pregunta 1 a)

En Key schedule se parte con una llave original de 64 bits la cual contiene 8 bits de paridad, quedando 56 bits. Estos 56 bits pasan por PC1 y son divididos en 2 mitades de 28 bits cada una Ci-1 y Di-1, luego se les aplica 1 o 2 Left Shift (según la ronda) obteniendo Ci y Di a los que se les aplica PC2 para obtener un ki en el que se seleccionan 48 bits. Ci y Di pasan a la siguiente ronda y se repite este proceso 16 veces. Esta información es explicada en la Figura siguiente (Fuente: Laboratorio de recherche en informatique, Université Paris Saclay, https://www.lri.fr/~fmartignon/documenti/systemesecurite/4-DES.pdf)



En PC1 se omiten los bits de paridad (8, 16, 4, 32, 40, 48, 56, 64) que no son escogidos y no aparecen en p1. Esta permutación se muestra en la siguiente figura para Left y Right (Fuente: Laboratorio de recherche en informatique, Université Paris Saclay)

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

En el caso de PC2 se seleccionan 48 bits de los 56 del último estado de la llave. De esos 48 bits, 24 son para cada lado (Izquierdo/Derecho). La transformación se muestra en la siguiente imagen ((Fuente: Laboratorio de recherche en informatique, Université Paris Saclay)

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

Pregunta 1b)

Si suponemos que DES realiza solo 3 rondas y tenemos un mensaje de texto plano m. Ignorando las Permutaciones inicial y final de DES obtenemos:

$m = L0||R0$

$c = Enc(m, k) = L3||R3$

A partir de la 1° ronda de DES obtenemos:

$L1 = R0$ y $R1 = L0 \oplus f1(R0)$

Luego de la 2° ronda de DES obtenemos:

$L2 = R1$ y $R2 = L1 \oplus f2(R1)$

Finalmente de la 3° ronda de DES se obtiene:

$L3 = R2$ y $R3 = L2 \oplus f3(R2)$

Entonces $L2 = f3(R2) \oplus R3 = f3(L3) \oplus R3$

Y como $R1 = L2$ se obtiene:

$R1 = f3(L3) \oplus R3$ y de la 1° ronda tenemos que $R1 = L0 \oplus f1(R0)$.

L0 y R0 los conocemos directamente del mensaje. Luego en un ataque de texto plano al obtener el mensaje cifrado c podemos obtener L3 y R3. A la vez, sabemos que por definición de fi:

$R1 = f(k3, L3) \oplus R3$ **y** $R1 = L0 \oplus f(k1, R0)$**.**

Y de *key-schedule* sabemos que teniendo un k inicial podemos obtener k1 y k3. Y con ello, evaluar ambas expresiones anteriores y comparar si son o no iguales.

Sin embargo, con fuerza bruta la llave podría tener 2^56 opciones. Pero desde *key-schedule* y tal como se indica en el enunciado, sabemos que se puede tratar cada mitad de la llave de forma independiente, teniendo cada una 2^28 opciones.

De lo anterior, mandando un mensaje de texto plano m y obteniendo c, y luego usando fuerza bruta para obtener un k con el que evaluar las expresiones de R1 destacadas es posible recuperar la llave k en menos de 2^30 aplicaciones de la red S/P.