



Algoritmo DES

Data Encryption Standard

Studenti

Marchesani Carmine - 113916

Donici Ionut Bogdan - 109585

Minetti Loris Emanuele - 115346

Link utili

Github

https://github.com/ionutbogdandonici/DES_C.git



Wikipedia

https://en.wikipedia.org/wiki/Data_Encryption_Standard



In poche parole...

- **Data Encryption Standard** (DES, letteralmente "Norma per la crittografia dei dati") è un algoritmo di cifratura scelto come standard dagli Stati Uniti d'America nel 1976 e in seguito diventato di utilizzo internazionale. Si basa su un algoritmo a chiave simmetrica con chiave a 64 bit (ma solo 56 utili poiché 8 sono di controllo).
- La poca sicurezza di questo algoritmo è stata dimostrata nel 1999, rendendo necessario lo sviluppo di altri algoritmi di cifratura. **TripleDes**, ovvero la tripla applicazione con chiavi differenti dell'algoritmo DES, risolve molti problemi del suo genitore.
- **AES** (Advanced Encryption Standard) è l'algoritmo oggi giorno più sicuro e scalabile in materia, ampiamente più utilizzato degli altri due sopracitati.



INPUT

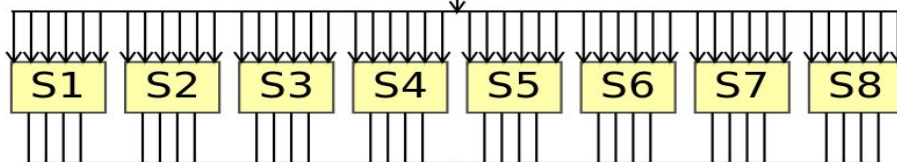
Stringa alfanumerica di 64 bit

Right / Left
block

Half Block (32 bits)

Subkey (48 bits)

E



P

OUTPUT

Stringa cifrata tramite chiave



STEP BY STEP

Messaggio iniziale esadecimale (64bit) , Key (56bit effective)

M □ 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111



L □ 00000001 00100011 01000101 01100111



R □ 10001001 10101011 11001101 11101111

K □ 0001001**1** 0011010**0** 0101011**1** 0111100**1** 1001101**1** 1011110**0** 1101111**1** 1111000**1**

Creazione di 16 Subkeys

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

K 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

PC-1
permutation

K+ 1111000 0110011 0010101 0101 111 0101010 1011001 1001111 0001111

C₀ 1111000 0110011 0010101 0101

D₀ 111 0101010 1011001 1001111 0001111

Definiti C₀ e D₀ ricaviamo 16 blocchi da 28 bit l'uno.

1 ≤ n ≤ 16

Come?

C_n D_n □ C_{n-1} D_{n-1} applicando ricorsivamente una **left shift** (<<) al blocco precedente... **n volte!**

Risultato?

$C_0 = 1111000011001100101010101111$
 $D_0 = 0101010101100110011110001111$
 $C_1 = 1110000110011001010101011111$
 $D_1 = 1010101011001100111100011110$
 $C_2 = 1100001100110010101010111111$
 $D_2 = 0101010110011001111000111101$
 $C_3 = 0000110011001010101011111111$
 $D_3 = 0101011001100111100011110101$
 $C_4 = 0011001100101010101111111100$
 $D_4 = 0101100110011110001111010101$
 $C_5 = 1100110010101010111111110000$
 $D_5 = 0110011001111000111101010101$
 $C_6 = 0011001010101011111111000011$
 $D_6 = 1001100111100011110101010101$
 $C_7 = 1100101010101111111100001100$
 $D_7 = 0110011110001111010101010110$
 $C_8 = 0010101010111111110000110011$
 $D_8 = 1001111000111101010101011001$

$C_9 = 0101010101111111100001100110$
 $D_9 = 0011110001111010101010110011$
 $C_{10} = 0101010111111110000110011001$
 $D_{10} = 1111000111101010101011001100$
 $C_{11} = 0101011111111000011001100101$
 $D_{11} = 1100011110101010101100110011$
 $C_{12} = 0101111111100001100110010101$
 $D_{12} = 0001111010101010110011001111$
 $C_{13} = 0111111110000110011001010101$
 $D_{13} = 0111101010101011001100111100$
 $C_{14} = 1111111000011001100101010101$
 $D_{14} = 1110101010101100110011110001$
 $C_{15} = 1111100001100110010101010111$
 $D_{15} = 1010101010110011001111000111$
 $C_{16} = 1111000011001100101010101111$
 $D_{16} = 0101010101100110011110001111$

Subkey additional permutation

C_1D_1 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110



PC-2
permutation

K_1 000110 110000 001011 101111 111111 000111 000001 110010 _____

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



PC-2 usa solo 48 bit poiché è composta da 56 bit

La permutazione PC-2 viene anch'essa applicata **n volte** alla coppia C_nD_n .


BASTA CON LE CHIAVI!

Torniamo al messaggio.

IP – Initial 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

M 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

 **IP**
permutation

IP 1100 1100 0000 0000 1100 1100 1111 1111 . 1111 0000 1010 1010 1111 0000 1010 1010

L₀ 1100 1100 0000 0000 1100 1100 1111 1111

R₀ 1111 0000 1010 1010 1111 0000 1010 1010

The formula

Ora si procede attraverso 16 iterazioni tale che $1 \leq n \leq 16$ utilizzando la seguente formula.

$$\begin{aligned} L_n &= R_{n-1} \\ R_n &= L_{n-1} + f(R_{n-1}, K_n) \end{aligned}$$

Questa funzione f opera su due blocchi da 32 bit ed una chiave, K_n , da 48 bit per produrre un blocco da 32 bit che rappresenta il risultato della applicazione della **S-Box** ed una permutazione.

Estensione di R_{n-1} da 32 bit a 48 bit

E BIT-SELECTION TABLE					
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

Attraverso l'utilizzo della tabella E prendiamo il blocco R_{n-1} , il quale è da 32 bit, e lo portiamo a 48 bit in modo da poter applicare l'operazione di XOR con la chiave di riferimento K_n .

Il risultato ottenuto viene suddiviso in gruppi da 6 bit ognuno, ai quali viene applicata la funzione S-Box portandoli da 6 bit a 4 bit.

S-BOX

S1																
Row No.	Column Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

La S-Box è una tabella che permette la riduzione di un gruppo di bit da 6 a 4.

Vengono presi il primo e l'ultimo bit dei 6 bit. La loro combinazione indica la riga da prendere in considerazione della tabella S.

I 4 bit interni, invece, rappresentano in formato decimale il valore della colonna di riferimento. Ciò che si trova all'interno della cella data dalla loro intersezione viene preso e trasformato in binario.

Permutation and Next-Step

Con la stringa ottenuta dall'applicazione dell' S-Box viene applicata la permutazione seguendo la tabella **P**.

Otteniamo così la stringa $f = 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011$.

A quest'ultima applichiamo la seguente formula:

$$\begin{aligned} R_1 &= L_0 + f(R_0, K_1) \\ &= 1100\ 1100\ 0000\ 0000\ 1100\ 1100\ 1111\ 1111 \\ &\quad + 0010\ 0011\ 0100\ 1010\ 1010\ 1001\ 1011\ 1011 \\ &= 1110\ 1111\ 0100\ 1010\ 0110\ 0101\ 0100\ 0100 \end{aligned}$$

Al prossimo round otteniamo $L_2 = R_1$, ovvero il blocco appena calcolato. Il prossimo da calcolare sarà $R_2 = L_1 + f(R_1, K_2)$.

Continueremo così fino al 16esimo round.

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

CypherText

Prendendo il blocco L_{16} ed R_{16} , invertiamo il loro ordine e li concateniamo, mettendo R_{16} a sinistra, e L_{16} a destra.

$$R_{16}L_{16} = 00001010\ 01001100\ 11011001\ 10010101\ 01000011\ 01000010\ 00110010\ 00110100$$

A questo punto applicheremo una permutazione seguendo la tabella IP-1 ottenendo così la seguente sequenza di bit:

$$IP^{-1} = 10000101\ 11101000\ 00010011\ 01010100\ 00001111\ 00001010\ 10110100\ 00000101$$

Che in esadecimale corrisponde a:

85E813540F0AB405



IL MESSAGGIO CRIPTATO !!

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

Messaggio:

**1FA230E6EED063AB35C8B50D67A48290927BDC7383F69BA
B**

Key:

133457799BBCDFF1

<http://des.online-domain-tools.com/>

