Introduzione alla cifratura con DES

Novembre 2017

DES la cifratura a chiave segreta

Il 15 maggio 1973 il **NBS** (*National Bureau of Standards*, oggi chiamato *NIST - National Institute of Standards and Technology*) ha lanciato una gara nel *Federal Register* (l'equivalente negli USA della *Gazzetta Ufficiale* in Italia) per la creazione di un algoritmo di cifratura che rispondesse ai criteri seguenti: possedere un alto livello di sicurezza legato ad una chiave di piccole dimensioni che serva da cifratura e da decifratura, essere comprensibile, non dipendere dalla confidenzialità dell'algoritmo, essere adattabile ed economico ed essere efficace e esportabile. Alla fine del 1974, IBM propone «Lucifer», che, grazie alla NSA (National Security Agency), venne modificato il 23 novembre 1976 per dare vita al **DES** (*Data Encryption Standard*). Il DES è stato finalmente approvato nel 1978 dall'NBS. Il DES fu messo in norma dall'*ANSI* (*American National Standard Institute*) con il nome di *ANSI X3.92*, più conosciuto sotto la

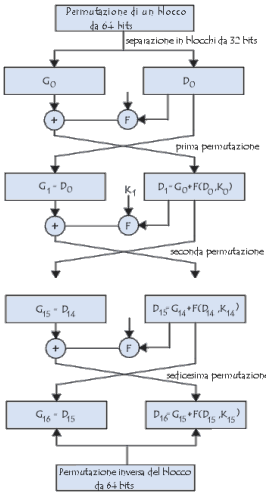
denominazione *DEA* (*Data Encryption Algorithm*).

Principio del DES

Si tratta di un sistema di cifratura simmetrica per blocchi di 64 bit, di cui 8 bit (un byte) servono da test di parità (per verificare l'integrità della chiave). Ogni bit di parità della chiave (1 ogni 8 bit) serve a testare un dei bit della chiave per parità dispari, cioè ogni bit di parità è sistemato in modo da avere un numero dispari di '1' nel gruppo degli 8 bit al quale appartiene. La chiave possiede quindi una lunghezza «utile» di 5 bit, il che significa che solo 56 bit servono in realtà all'algoritmo. L'algoritmo consiste nell'effettuare delle combinazioni, delle sostituzioni e delle permutazioni tra il testo da cifrare e la chiave, facendo in modo che le operazioni possano farsi nei due sensi (per la decifratura). La combinazione tra sostituzioni e permutazioni è detta **codice prodotto**. La chiave è codificata a 64 bit e formata da 16 blocchi di 4 bit, generalmente abbreviati k_1 a k_{16}. Dato che «solo» 56 bit servono effettivamente a codificare, possono esisterne 2^{56} (sia $7.2*10^{16}$) chiavi diverse.

L'algoritmo di DES

Le grandi linee dell'algoritmo sono le seguenti: frazionamento del testo in blocchi da 64 bit (8 byte), permutazione iniziale dei blocchi, divisione dei blocchi in due parti, sinistra e destra, detti *S* e *D*, tappe di permutazione e di sostituzione ripetute 16 volte (dette **round**) e incollare le parti sinistra e destra poi permutazione iniziale inversa:



Frazionamento del testo

Permutazione iniziale

In un primo tempo, ogni bit di un blocco è sottoposto alla permutazione iniziale, che può essere rappresentata dalla matrice di permutazione iniziale (sigla *IP*) seguente:

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

Questa matrice di permutazione indica, percorrendo la matrice da sinistra a destra poi dall'alto in basso, che il 58^{esimo} bit del blocco del testo da 64 bit si trova in prima posizione, il 50^{esimo} in seconda posizione e così via.

Scissione in blocchi da 32 bit

Una volta che la permutazione iniziale è realizzata, il blocco di 64 bit è scisso in due blocchi da 32 bit, siglati rispettivamente **S** e **D** (per sinistra e destra, con la sigla anglosassone *L* e *R* per *Left and Right*). Si nota $\mathbf{S}_0$ e $\mathbf{D}_0$ lo stato iniziale di questi due blocchi:

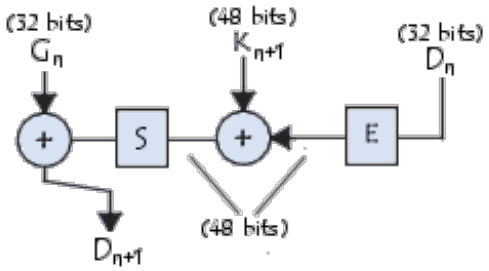
$\mathbf{S}_0$								
	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8

$\mathbf{D}_0$								
	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

È interessante osservare che $\mathbf{S}_0$ contiene tutti i bit che possiedono una posizione pari nel messaggio iniziale, mentre $\mathbf{D}_0$contiene i bit di posizione dispari.

Round

I blocchi $\mathbf{S}_n$ e $\mathbf{D}_n$ sono sottoposti ad una serie di trasformazioni interattive dette *round*, esplicitate in questo schema, e i cui dettagli vengono forniti più in basso:



Funzione d'espansione

I 32 bit del blocco $\mathbf{D}_0$ sono estesi a 48 bit grazie ad una tabella (matrice) detta *tabella d'espansione* (sigla **E**), nella quale i 48 bit sono mischiati e 16 fra loro sono duplicati:

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

Così, l'ultimo bit di $\mathbf{D}_0$ (cioè il 7^{imo}bit del blocco d'origine) diventa il primo, il primo diventa il secondo, ecc. In più, i bit 1,4,5,8,9,12,13,16,17,20,21,24,25,28 e 29 di $\mathbf{D}_0$ (rispettivamente 57, 33, 25, l, 59, 35, 27, 3, 6l, 37, 29, 5, 63, 39, 31 e 7 del blocco d'origine) sono duplicati e disseminati nella matrice.

O esclusivo con la chiave

La matrice che risulta dai 48 bit è detta $\mathbf{D'}_0$ oppure $\mathbf{E}[\mathbf{D}_0]$. L'algoritmo DES esegue quindi un *O esclusivo* tra la prima chiave $\mathbf{K}_1$ e $\mathbf{E}[\mathbf{D}_0]$. Il risultato di questo *O esclusivo* è una matrice di 48 bit che chiameremo $\mathbf{D}_0$ per comodità (non si tratta dello stesso $\mathbf{D}_0$ dell'inizio).

Funzione di sostituzione

$\mathbf{D}_0$ e in seguito scisso in 8 blocchi da 6 bit, sigla $\mathbf{D}_{0i}$. Ciascuno di questi blocchi passa per delle **funzioni di selezione** (dette talvolta *scatole di sostituzione* o *funzioni di compressione*), siglate generalmente $\mathbf{S}_i$. I primi e gli ultimi bit di ogni $\mathbf{D}_{0i}$ determinano (in codice binario) la linea della funzione di selezione, gli altri bit (rispettivamente2, 3, 4 e 5) determinano la colonna. Dato che la sezione della linea si fa su due bit, vi sono 4 possibilità (0,1,2,3). Dato che la sezione della linea si fa su 4 bit, vi sono 16 possibilità (da 0 a 15). Grazie a questa informazione, la funzione di selezione "seleziona" un valore codificato su 4 bit. Ecco la prima funzione di sostituzione, rappresentata da una matrice di 4 per 16:

S_1

	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

Sia $\mathbf{D}_{01}$ uguale a *101110*. I primi e gli ultimi bit danno *10*, cioè 2 in codice binario. I bit 2, 3, 4 e 5 danno *0111*, ovvero 7 in binario. Il risultato della funzione di selezione è quindi il valore situato nelle linea n° 2, nella colonna n° 7. Si tratta del valore *11*, sia in binario *111*. Ciascuno degli 8 blocchi di 6 bit è passato nella funzione di selezione corrispondente, che da a ognuno 8 valori di 4 bit ciascuno. Ecco le altre funzioni di selezione:

S_2

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Ogni blocco di 6 bit è poi sostituito con un blocco di 4 bit. Questi bit sono raggruppati per formare un blocco da 32 bit.

Permutazione

Il blocco di 32 bit ottenuto è infine sottoposto ad una permutazione **P** di cui vediamo la tabella:

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

O esclusivo

L'insieme di questi risultati in uscita da **P** è sottoposto ad un *O esclusivo* con $\mathbf{S}_0$ di partenza (come indicato nel primo schema) per dare D1, mentre lo $\mathbf{D}_0$ iniziale da $\mathbf{S}_1$.

iterazione

L'insieme delle tappe precedenti (*round*) e reiterato 16 volte.

Permutazione iniziale inversa

Alla fine delle iterazioni, i due blocchi $\mathbf{S}_{16}$ e $\mathbf{D}_{16}$sono "reincollati", poi sottomessi alla permutazione iniziale inversa:

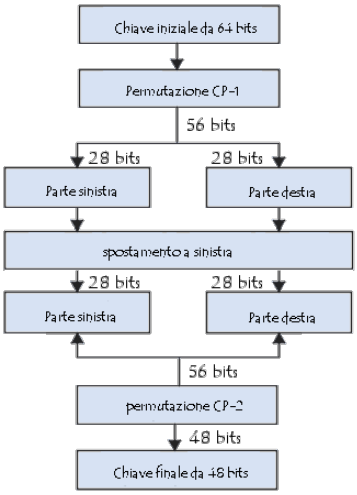
IP-1

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

Il risultato in uscita è un testo codificato a 64 bit.

Generazione delle chiavi

Visto che l'algoritmo di DES presentato qui sopra è pubblico, tutta la sicurezza è basata sulla complessità delle chiavi di cifratura. L'algoritmo qui sotto mostra come ottenere, partendo da una chiave a 64 bit (composta da 64 caratteri alfanumerici qualunque), 8 chiavi diversificate di 48 bit da utilizzare nell'algoritmo di DES:



In un primo tempo i bit di parità della chiave sono eliminati per ottenere una chiave di una lunghezza utile di 56 bit. La prima tappa consiste in una permutazione siglata **CP-1** con la seguente matrice:

CP-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

Questa matrice può infatti essere scritta sotto forma di due matrici $\mathbf{S}_i$ e $\mathbf{D}_i$ (per sinistra e destra) composta ciascuna da 28 bit:

$\mathbf{S}_i$							
	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

$\mathbf{D}_i$							
	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

Osserviamo come $\mathbf{S}_0$ e $\mathbf{D}_0$ il risultato di questa prima permutazione. Questi due blocchi subiscono poi una rotazione verso sinistra, in maniera che i bit in seconda posizione vadano in prima, quelli in terza in seconda, ecc. I bit in prima posizione passano invece all'ultima. I due blocchi da 28 bit sono in seguito raggruppati in un blocco da 56 bit. Questo passa da una permutazione, detta **CP-2**, fornendo un blocco di 48 bit, che rappresenta la chiave $\mathbf{K}_i$:

CP-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

Alcune iterazioni sull'algoritmo permettono di dare le 16 chiavi $\mathbf{K}_1$ a $\mathbf{K}_{16}$ usate nell'algoritmo di DES.

LS															
	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

TDES, un'alternativa al DES

Nel 1990 Eli Biham e Adi Shamir hanno messo a punto la criptanalisi differenziale che ricerca delle COPIE di testi in chiaro e di testi cifrati. Questo metodo funziona fino a un numero di round inferiore a 15, e nell'algoritmo qui sopra vi sono 16 round. In ogni caso, anche se in una chiave di 56 bit vi sono molteplici possibilità, numerosi processori permettono di calcolare più di 10^6 chiavi al secondo, così, se utilizzate parallelamente su un grande numero di terminali, danno la possibilità ad un grande organismo (uno stato ad esempio) di trovare la chiave corretta. Una soluzione a breve termine può consistere nell'incatenare tre codificazioni DES attraverso due chiavi da 56 bit (il che equivale ad una chiave di 112 bit). Questo processo è chiamato **Triplo DES**, sigla *TDES* (a volte *3DEs* o *3-DES*):

Il **TDES** permette di aumentare significativamente la sicurezza del DES, ma ha come inconveniente maggiore di chiedere più risorse per la cifratura e la decifratura. Distinguiamo solitamente diversi tipi di codificazione triplo DES: DES-EEE3: 3 codificazioni DES con 3 chiavi differenti; DES-EDE3: una chiave diversa per ognuna delle 3 operazioni DES (cifratura, decifratura, cifratura); DES-EEE2 e DES-EDE2: una chiave diversa per la seconda operazione (decifratura). Nel 1997 il *NIST* lanciò una nuova consultazione di progetto per l'elaborazione dell'**AES**(*Advanced Encryption Standard*), un algoritmo di cifratura destinato a sostituire il *DES*. Il sistema di cifratura *DES* fu aggiornato ogni 5 anni. Nel 2000 all'ultima versione, dopo un processo di valutazione durato 3 anni, l'algoritmo elaborato congiuntamente da due candidati belgi, *Vincent Rijmen* e *Joan Daemen* fu scelto come nuovo standard dal *NIST*. Questo nuovo algoritmo battezzato **RIJNDAEL** dai suoi inventori, sostituirà da quel momento il DES.

Ulteriori informazioni

RFC 2420 The PPP Triple-DES Encryption Protocol (3DESE).

Foto: © Pixabay.

‹ Precedente

- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15

Successivo ›