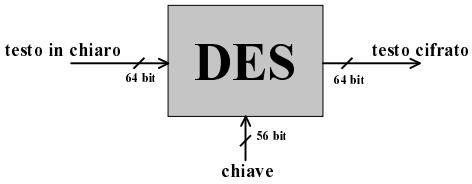
Data Encryption Standard (DES)

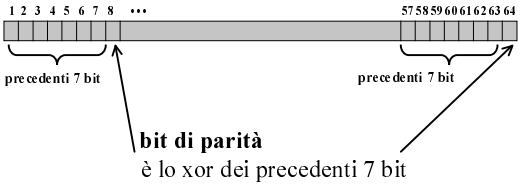
- 15 maggio 1973, richiesta pubblica per uno standard della NBS, oggi NIST (27 agosto 1974, seconda richiesta)
- Modifica di Lucifer, sviluppato da IBM (chiave da 128 a 56 bit) reso noto nel 1975
- 1976: due workshop
- Standard pubblicato 15 gennaio 1977
- Riaffermato per successivi 5 anni nel 1983, 1987, 1992
- DES challanges
- Advanced Encryption Standard (AES)

Data Encryption Standard

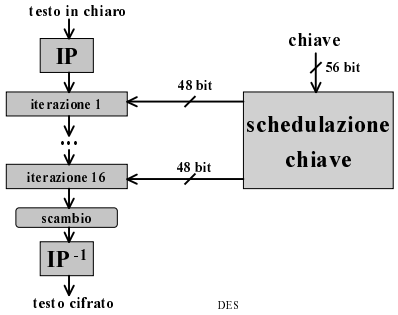


Lunghezza della Chiave

Nello standard DES la chiave è lunga 64 bit
8 byte di cui l'ottavo bit è di parità

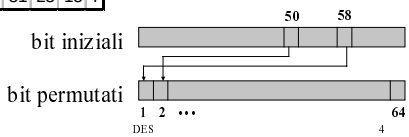


Struttura del DES



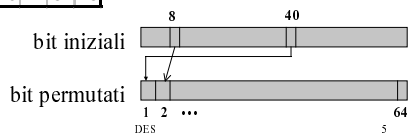
Permutazione Iniziale IP

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

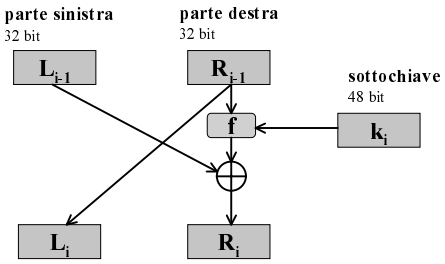


Permutazione Inversa IP^{-1}

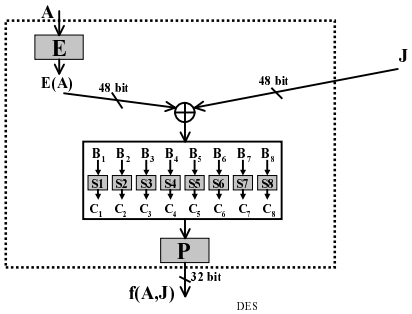
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



Singola Iterazione



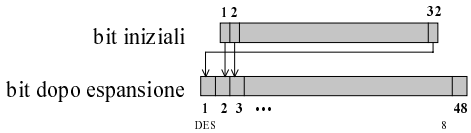
La funzione f



Espansione 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

16 bit sono duplicati



S-box

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	0110	0111	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	7	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	10	2	4	9	1	7	5	11	3	14	10	0	6	13

Box S1

6 bit in input specificano un elemento della tabella la cui conversione binaria dà i 4 bit output

Funzionamento delle S-box

input **101110**

primo ed ultimo bit **10**

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	0110	0111	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	7	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	10	2	4	9	1	7	5	11	3	14	10	0	6	13

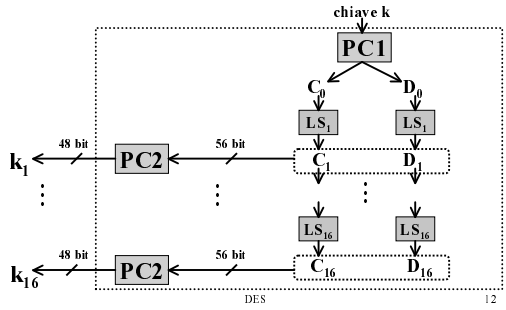
Box S1

output 11 in binario = **1011**

Proprietà delle S-box [NBS, 1976]

- Ogni riga è una permutazione degli interi 0,..,15
- Nessuna S-box è una funzione affine o lineare dei suoi input
- Cambiando un solo bit di input ad una S-box variano almeno due bit nell'output
- Per ogni S-box S e per ogni input x a 6 bit:
  $S(x)$ e $S(x \oplus 001100)$ differiscono in almeno due bit
- Per ogni S-box, per ogni input x e per ogni bit d,g,
  $S(x) \neq S(x \oplus 11dg00)$
- Per ogni S-box, se fissiamo un bit di input e osserviamo il valore di un fissato bit di output, il numero degli input per i quali il bit di output vale 0 è circa uguale al numero degli input per i quali tale bit vale 1

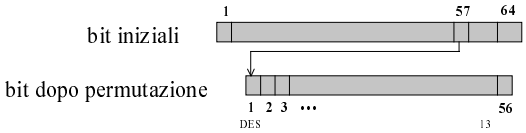
Schedulazione delle chiavi



Permutazione 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

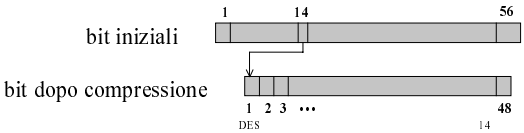
i bit in posizione 8, 16, 24, 32, 40, 48, 56, 64 sono di parità e non compaiono



Compressione-permutazione 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	26	41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32

8 bit soppressi: in posizione 9, 18, 22, 25, 35, 38, 43 e 54



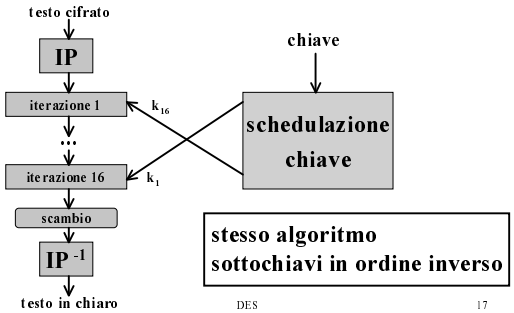
Schedulazione degli shift a sinistra

iterazione	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Totale shift nelle 16 iterazioni = 28 posizioni

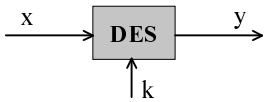
Decifratura DES

Decifratura DES

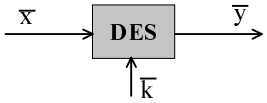


Proprietà del complemento

Se



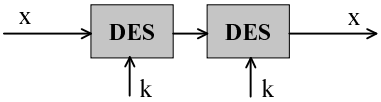
allora



$\bar{\cdot}$ è il complemento bit per bit

Chiavi deboli

k è una chiave debole se per ogni x

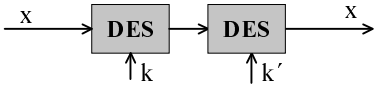


Ci sono 4 chiavi deboli

chiave debole	C_0	D_0
0101 0101 0101 0101	0^{28}	0^{28}
FEFE FEFE FEFE FEFE	1^{28}	1^{28}
1F1F 1F1F OEOE OEOE	0^{28}	1^{28}
E0E0 E0E0 F1F1 F1F1	1^{28}	0^{28}

Chiavi semideboli

k,k′ è una coppia di chiavi semideboli se per ogni x



Ci sono 6 coppie di chiavi semideboli

C_0	D_0	k	k'	C_0	D_0
$\{01\}^{14}$	$\{01\}^{14}$	01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01	$\{10\}^{14}$	$\{10\}^{14}$
$\{01\}^{14}$	$\{10\}^{14}$	1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E	$\{10\}^{14}$	$\{01\}^{14}$
$\{01\}^{14}$	0^{28}	01E0 01E0 01F1 01F1	E001 E001 F101 F101	$\{10\}^{14}$	0^{28}
$\{01\}^{14}$	1^{28}	1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E	$\{10\}^{14}$	1^{28}
0^{28}	$\{01\}^{14}$	011F 011F 010E 010E	1F01 1F01 OE01 0E01	0^{28}	$\{10\}^{14}$
1^{28}	$\{01\}^{14}$	E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1	1^{28}	$\{10\}^{14}$

Crittoanalisi differenziale

- Eli Biham e Adi Shamir [1990]
- Già conosciuto da Coppersmith quando fu progettato !?

numero round	chosen plaintext	known plaintext
8	2^{14}	2^{38}
9	2^{24}	2^{44}
10	2^{24}	2^{43}
11	2^{31}	2^{47}
12	2^{31}	2^{47}
13	2^{39}	2^{52}
14	2^{39}	2^{51}
15	2^{47}	2^{56}
16	2^{47}	2^{55}

numero messaggi in chiaro

Crittoanalisi differenziale e lineare

Attacco *known-plaintext* oppure *chosen-plaintext*

Metodo di attacco	known plaintext	chosen plaintext	complessità spazio	complessità tempo
precomputazione esaustiva	-	1	2^{56}	1
ricerca esaustiva	1	-	trascurabile	2^{55}
crittoanalisi lineare	2^{43} (85%)	-	messaggi	2^{43}
	2^{38} (10%)	-	messaggi	2^{50}
crittoanalisi differenziale	-	2^{47}	messaggi	2^{47}
	2^{55}		messaggi	2^{55}

percentuale di successo

Ricerca esaustiva

- Numero chiavi DES = $2^{56} \approx 7{,}2056 \cdot 10^{16}$
- Un computer a 500 Mhz che testa una chiave per ciclo di clock impiega

  144.115.188 secondi ≈ 834 giorni ≈ 2 anni e 3 mesi

  per provare $2^{55} \approx 3{,}6 \cdot 10^{16}$ chiavi

DES challanges

- 10.000 dollari al primo che rompe la *challenge* se rotta entro il 25% del miglior tempo precedente
- **Giugno 1997**: 39 giorni, testato 24% delle 2^{56} chiavi, Rocke Verser scrisse e distribuì un client di ricerca, 70.000 computer, trovata da M. K. Sanders (Pentium 90 MHz, 16M)
- **Luglio 1998**: 56 ore, Deep Crack, EFF, 250.000 dollari
- **Gennaio 1999**: 22 ore 15 minuti testando 245 miliardi di chiavi al secondo, Distributed.Net 100.000 computer e EFF

Modalità operative del DES

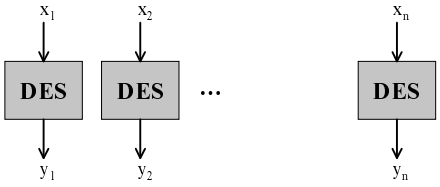
Come cifrare testi più lunghi di 64 bit?

- Electronic codebook chaining (ECB)
- Cipher block chaining (CBC)
- Cipher feedback (CFB)
- Output feedback (OFB)

NBS FIPS PUB 46, DES modes of operation, National Bureau of Standards, 1977

Electronic codebook chaining (ECB)

messaggio in chiaro $x = x_1x_2\ldots x_n$ (diviso in n blocchi di 64 bit)

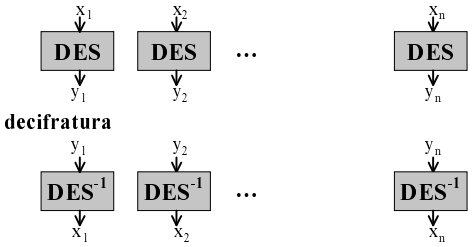


messaggio cifrato $y = y_1y_2\ldots y_n$

Electronic codebook chaining (ECB)

decifratura

cifratura



decifratura

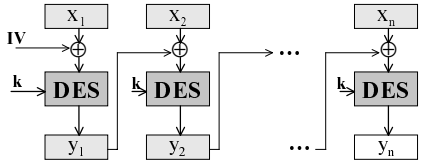
Electronic codebook chaining (ECB)

- Se la lunghezza del messaggio non è multiplo di 64?
  Possibile soluzione: Padding con 100…00
- L'ECB è il metodo più veloce
- Eventuali errori non si propagano

- Non c'è dipendenza tra i blocchi
  - Possibili attacchi di sostituzione
  - Ridondanza testo in chiaro

Cipher Block Chaining (CBC)

messaggio in chiaro $x=x_1x_2\ldots x_n$ (diviso in n blocchi di 64 bit)



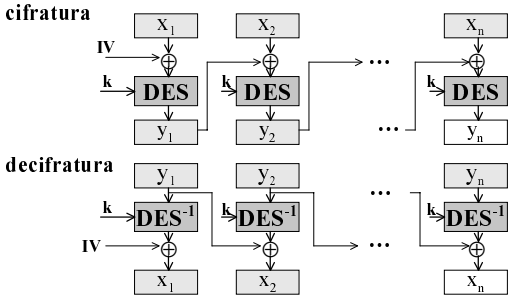
messaggio cifrato $y = y_1y_2\ldots y_n$

vettore di inizializzazione IV di solito pubblico,
(potrebbe anche essere scelto a caso e tenuto nascosto)

Cipher Block Chaining (CBC)

decifratura

cifratura



DES 30

Cipher Block Chaining (CBC)

- Meno veloce dell'ECB
- Propagazione errori

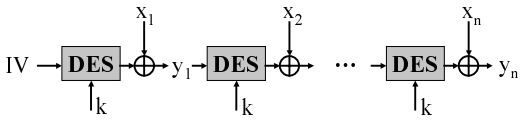
- C'è dipendenza tra i blocchi

Non possibili attacchi di sostituzione

DES 31

Cipher feedback (CFB)

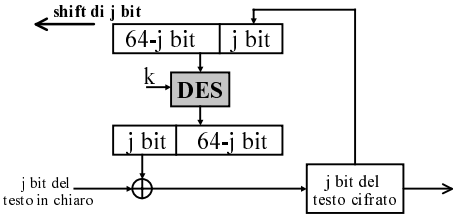
messaggio in chiaro $x=x_1x_2...x_n$ (diviso in n blocchi di 64 bit)



messaggio cifrato $y = y_1y_2\ldots y_n$

DES 32

j-bit Cipher feedback

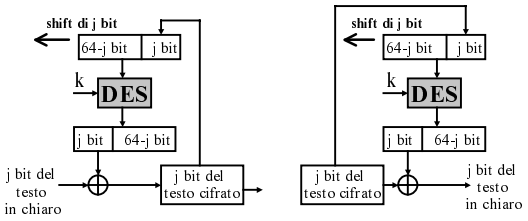


Si inizia cifrando IV

DES 33

j-bit Cipher feedback

decifratura



DES 34

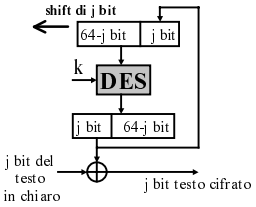
j-bit Cipher feedback

- j può essere scelto a piacimento, ad es. j=8

- Si possono utilizzare j bit cifrati senza aspettarne 64
- Più lento al decrescere di j

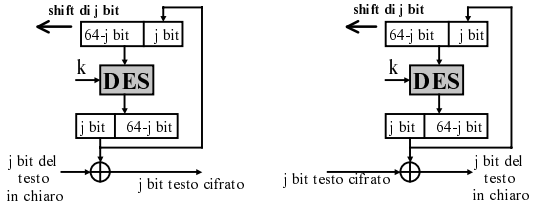
DES 35

j-bit Output feedback



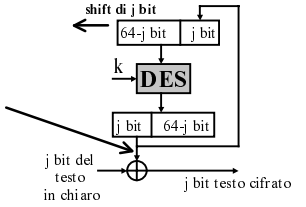
Si inizia cifrando IV

j-bit Output feedback
decifratura



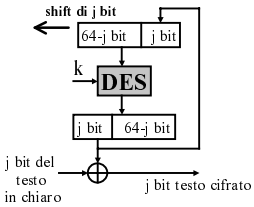
Cifratura **Decifratura**

j-bit Output feedback



I valori input allo xor possono essere precomputati

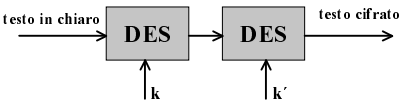
j-bit Output feedback



Se la stessa chiave e lo stesso IV vengono usati per diversi OFB, la *keystream* è la stessa!

IV deve essere cambiato se si usa la stessa chiave

DES Doppio

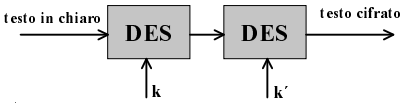


Cifratura

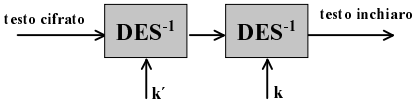
lunghezza blocco = 64 bit
chiave (k, k´) lunga 56+56 = 112 bit

DES Doppio

Cifratura



Decifratura



Sicurezza DES doppio

Quanto è "sicuro" il DES doppio?

DES ≡ DES doppio ?

E' possibile che per ogni (k, k´) esiste k´´ tale che

$DES_{k''}(\cdot) = DES_{k'}(DES_k(\cdot))$

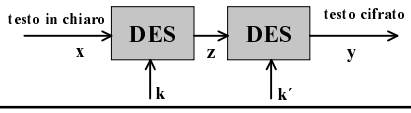
DES non forma un gruppo

- Ci sono $(2^{64})! > 10^{347.380.000.000.000.000.000} > 10^{10^{20}}$ permutazioni per i 2^{64} input
- Ci sono solo 2^{56} permutazioni definite dal DES

L'insieme delle 2^{56} permutazioni definite dalle 2^{56} chiavi DES non è chiuso per composizione (dimostrato solo nel 1992)

|Gruppo generato da composizione di DES| $> 10^{2499}$

DES Doppio: attacco *meet in the middle*



Known Plaintext Attack
Input: x, y = $DES_{k'}(DES_k(x))$
Costruisci tabella

chiave	testo cifrato
k"	$DES_{k"}(x)$
…	…

for $k_2 \in \{0,1\}^{56}$
  do $z = DES^{-1}_{k_2}(y)$
    if per qualche k_1, (k_1, z) è nella tabella
      then return la chiave è (k_1,k_2)

DES Doppio: attacco *meet in the middle*

Known Plaintext Attack
Input: x, y = $DES_{k'}(DES_k(x))$
Costruisci tabella

chiave	testo cifrato
k"	$DES_{k"}(x)$
…	…

for $k_2 \in \{0,1\}^{56}$
  do $z = DES^{-1}_{k_2}(y)$
    if per qualche k_1, (k_1, z) è nella tabella
      then return la chiave è (k_1,k_2)

Complessità spazio: 2^{56} righe nella tabella

Complessità tempo: 2^{57} cifrature + 2^{56} ricerche in tabella

O(1) se tabella hash
56 se array ordinato

DES Doppio: attacco *meet in the middle*

L'output (k_1,k_2) è sicuramente la chiave cercata?

DES Doppio: attacco *meet in the middle*

Dato x, y qual'è il numero medio di chiavi (k_1,k_2) tali che

$y = DES_{k_2}(DES_{k_1}(x))$

DES Doppio: attacco *meet in the middle*

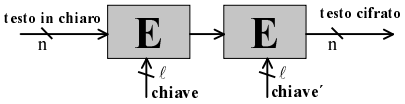
Dato x, y, qual è il numero medio di chiavi (k_1,k_2) tali che

$y = DES_{k_2}(DES_{k_1}(x))$

Fissato x, ci sono 2^{112} chiavi e 2^{64} testi cifrati y

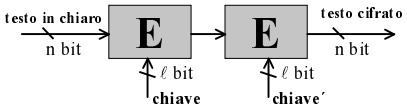
$$\frac{\#\text{chiavi}}{\#y \text{ per fissato } x} = \frac{2^{112}}{2^{64}} = 2^{48}$$

Doppia cifratura



Cifrario a blocchi *casuale*: dati n ed ℓ, scegli a caso 2^{ℓ} permutazioni tra le $(2^n)!$ possibili su 2^n elementi, ed associale con le 2^{ℓ} chiavi

Doppia cifratura



Dato x, y il numero medio di chiavi (k_1,k_2) tali che

$y = E_{k_2}(E_{k_1}(x))$

è

$$2^{2\ell-n}$$

DES Doppio: attacco *meet in the middle*

Known Plaintext Attack
Input: x, $y = DES_k(DES_k(x))$
x′, $y' = DES_k(DES_k(x'))$
Costruisci tabella
for $k_2 \in \{0,1\}^{56}$
do $z = DES^{-1}_{k_2}(y)$
if per qualche k_1, (k_1, z) è nella tabella
e $y' = DES_{k_2}(DES_{k_1}(x'))$
then return la chiave è (k_1,k_2)

chiave	testo cifrato
k''	$DES_{k''}(x)$
…	…

DES Doppio: attacco *meet in the middle*

Dato x, y, x′, y′ qual è il numero medio di chiavi (k_1,k_2) tali che

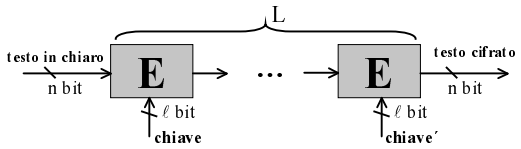
$y = DES_{k_2}(DES_{k_1}(x))$

$y' = DES_{k_2}(DES_{k_1}(x'))$

Fissati x, x′, ci sono 2^{112} chiavi e 2^{128} testi cifrati y, y′

$$\frac{\#\text{chiavi}}{\#y,y' \text{ per fissati } x,x'} = \frac{2^{112}}{2^{128}} = 2^{-16}$$

Cascata con L-stadi di un cifrario



Dati $x_i,y_i,\dots,x_t,y_t$ il numero medio di chiavi $(k_1,k_2,\dots,k_L)$ tali che

$$y_i = E_{k_L}(\dots E_{k_2}(E_{k_1}(x_i))\,)$$

è

$$2^{L\ell - tn}$$

DES Doppio: attacco *meet in the middle*

- Tradeoff tempo-memoria
- Indovino i primi s bit di k, $0 \le s \le 56$
- 2^s tabelle di 2^{56-s} righe

Complessità spazio: 2^{56-s} righe nella tabella

Complessità tempo: $\underbrace{2^s 2^{56-s}}_{2^{56}}$ cifrature + $\underbrace{2^s 2^{56}}_{2^{56+s}}$ ricerche in tabella

SPAZIO * TEMPO $\approx 2^{112}$

DES Doppio: attacco *meet in the middle*

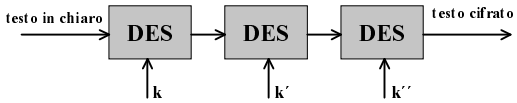
Known Plaintext Attack
Input: $x, y = DES_k(DES_k(x))$
$x', y' = DES_k(DES_k(x'))$
for $u \in \{0,1\}^s$
Costruisci tabella per $v \in \{0,1\}^{56-s}$

chiave	testo cifrato
k'' = u v	$DES_{k''}(x)$
…	…

for $k_2 \in \{0,1\}^{56}$
do $z = DES^{-1}_{k_2}(y)$
if per qualche k_1, (k_1, z) è nella tabella
e $y' = DES_{k_2}(DES_{k_1}(x'))$
then return la chiave è (k_1,k_2)

Complessità spazio: 2^{56-s} righe nella tabella
Complessità tempo: $2^s 2^{56-s}$ cifrature + $2^s 2^{56}$ ricerche in tabella
SPAZIO * TEMPO $\approx 2^{112}$

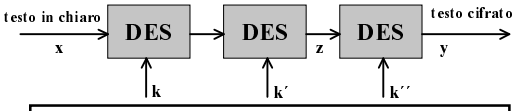
DES Triplicato



Cifratura

- lunghezza blocco = 64 bit
- chiave (k, k´,k´´) lunga 56 + 56 + 56 = 168 bit

DES Triplicato: attacco *meet in the middle*



Known Plaintext Attack
Input: $x, y = DES_{k''}(DES_k(DES_k((x)))$
Costruisci tabella
for $k_3 \in \{0,1\}^{56}$
do $z = DES^{-1}_{k_3}(y)$
if per qualche k_1,k_2, (k_1k_2, z) è nella tabella
then return la chiave è (k_1,k_2,k_3)

chiave	testo cifrato
(k''',k''')	$DES_{k'''}(DES_{k''''}(x))$
…	…

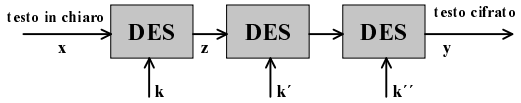
DES Triplicato: attacco *meet in the middle*

Known Plaintext Attack
Input: $x, y = DES_{k''}(DES_k(DES_k((x)))$
Costruisci tabella
for $k_3 \in \{0,1\}^{56}$
do $z = DES^{-1}_{k_3}(y)$
if per qualche k_1,k_2, (k_1k_2, z) è nella tabella
then return la chiave è (k_1,k_2,k_3)

chiave	testo cifrato
(k''',k''')	$DES_{k'''}(DES_{k''''}(x))$
…	…

Complessità spazio: 2^{112} righe nella tabella
Complessità tempo: $2^{112}+2^{56}$ cifrature + 2^{56} ricerche in tabella

DES Triplicato: attacco *meet in the middle*



Known Plaintext Attack
Input: x, y = $DES_{k''}(DES_{k'}(DES_k((x)))$
Costruisci tabella
for $k_3,k_2 \in \{0,1\}^{56} \times \{0,1\}^{56}$
do $z = DES^{-1}_{k_2}(DES^{-1}_{k_3}(y))$
if per qualche k_1, (k_1, z) è nella tabella
then return la chiave è (k_1,k_2,k_3)

chiave	testo cifrato
k'''	$DES_{k'''}(x)$
…	…

DES Triplicato: attacco *meet in the middle*

Known Plaintext Attack
Input: x, y = $DES_{k''}(DES_{k'}(DES_k((x)))$
Costruisci tabella
for $k_3,k_2 \in \{0,1\}^{56} \times \{0,1\}^{56}$
do $z = DES^{-1}_{k_2}(DES^{-1}_{k_3}(y))$
if per qualche k_1, (k_1, z) è nella tabella
then return la chiave è (k_1,k_2,k_3)

chiave	testo cifrato
k'''	$DES_{k'''}(x)$
…	…

Complessità spazio: 2^{56} righe nella tabella
Complessità tempo: $2^{56}+2^{112}$ cifrature + 2^{112} ricerche in tabella

DES Triplo: attacco *meet in the middle*

Complessità *Known Plaintext Attack* $\approx 2^{112}$

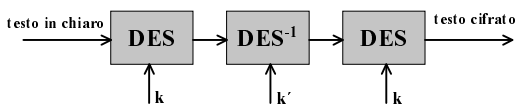
Ricerca esaustiva su tutte le chiavi $\approx 2^{112}$

DES Triplicato: attacco *meet in the middle*

Complessità *Known Plaintext Attack* $\approx 2^{112}$

"Equivalente" ad un cifrario con una chiave di 112 bit, e non 168 bit

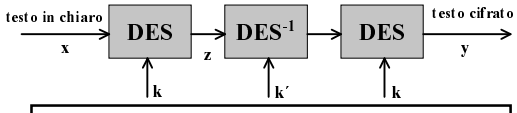
DES Triplo



Cifratura

- lunghezza blocco = 64 bit
- chiave (k, k´) lunga 56+56 = 112 bit
- spesso chiamato $EDE_{k,k'}$ (acronimo per Encrypt Decrypt Encrypt)
- adottato negli standard X9.17 e ISO 8732

DES Triplo: attacco *meet in the middle*



Known Plaintext Attack
Input: x, y = $DES_k(DES^{-1}_{k'}(DES_k((x)))$
Costruisci tabella
for $k_1,k_2 \in \{0,1\}^{56} \times \{0,1\}^{56}$
do $z = DES_{k_2}(DES^{-1}_{k_1}(y))$
if (k_1, z) è nella tabella
then return la chiave è (k_1,k_2)

chiave	testo cifrato
k''	$DES_{k''}(x)$
…	…

DES Triplo: attacco *meet in the middle*

Known Plaintext Attack
Input: x, $y = DES_k(DES^{-1}_{k'}(DES_k((x)))$
Costruisci tabella
for $k_1,k_2 \in \{0,1\}^{56} \times \{0,1\}^{56}$
do $z = DES_{k_2}(DES^{-1}_{k_1}(y))$
if (k_1, z) è nella tabella
then return la chiave è (k_1,k_2)

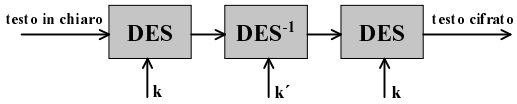
chiave	testo cifrato
k'	$DES_{k'}(x)$
…	…

Complessità spazio: 2^{56} righe nella tabella
Complessità tempo: $2^{56}+2^{112}$ cifrature + 2^{112} ricerche in tabella

DES 66

Compatibilità DES Triplo e DES

Se k = k´ il DES triplo

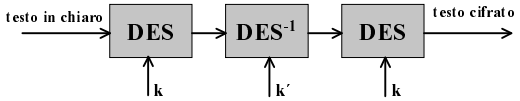


è equivalente al semplice DES

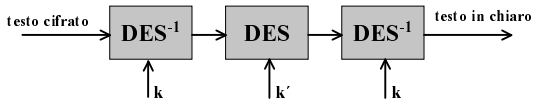
DES 67

Decifratura DES Triplo

Cifratura



Decifratura



DES 68

Altri cifrari

- IDEA (International Data Encryption Algorithm) [1990]
- SAFER (Secure And Fast Encryption Routine)
  SAFER K-64 [1994], SAFER K-128 [1995]
- RC5 [1995]

cifrario	bit chiave	bit testo
IDEA	128	64
SAFER K-64	64	64
SAFER K-128	128	64
RC5	<256 byte	32,64,128

- Madryga, NewDES, FEAL, REDOC, LOKI, Khufu, Knafre, RC2, MMB, GOST, Blowfish, …
- … AES

DES 69