

# DES e AES

Paolo D'Arco  
pdarco@unisa.it

Università di Salerno

Elementi di Crittografia

- 1 Data Encryption Standard (DES)
- 2 Advanced Encryption Standard (AES)

# Data Encryption Standard (DES)

Sviluppato negli anni '70 (IBM-NSA). Adottato nel 1977.

È stato scrutinato a fondo. In pratica non sono stati trovati attacchi migliori della ricerca esaustiva.

Il DES è una rete di Feistel a 16 round

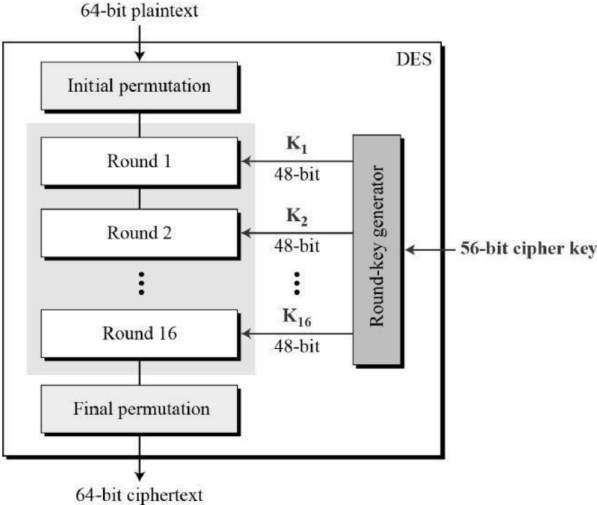
- lunghezza di blocco 64 bit
- lunghezza chiave 56 bit

La funzione di round prende in input una sottochiave di 48 bit ed una stringa di 32 bit.

L'algoritmo di scheduling delle chiavi DES deriva le sottochiavi dalla master key

master key  $K$  (56 bit)  $\Rightarrow K_1, K_2, \dots, K_{16}$  (48 bit ognuna)

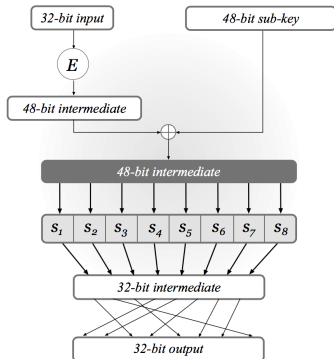
I 48 bit di ogni sottochiave: 24 dalla parte sinistra di  $K$  e 24 dalla parte destra



La funzione di round del DES (DES Mangler function) é costruita usando il paradigma delle SPN

$$\hat{f}_i(K_i, R), \quad K_i \in \{0, 1\}^{48}, \quad R \in \{0, 1\}^{32}$$

La stessa funzione  $\hat{f}_i$  per tutti i round ( $f_i$  differente per le sottochiavi  $K_i$ )



I passi della funzione sono i seguenti

- $R$  viene esteso attraverso la funzione di espansione  $E : R \rightarrow E(R)$  a 48 bit
- $E(R) \oplus K_i$  é una stringa di 48 bit, input per le  $S$ -box

La funzione di espansione é

*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

Le  $S$ -box **non** sono invertibili (gli input sono piú lunghi degli output)

La specifica dell'algoritmo é completamente pubblica. Soltanto la master key é segreta

Le  $S$ -box sono il cuore di  $\hat{f}$  e sono cruciali per la sicurezza del DES

Anche un minimo cambiamento puó introdurre debolezze

Esibiscono le seguenti proprietá:

- 1 ciascuna  $S$ -box é una funzione 4-a-1
- 2 ciascuna riga della tabella contiene tutte le possibili stringhe di 4 bit esattamente **una** volta
- 3 cambiando un bit ad ogni stringa di input di una  $S$ -box, si cambiano sempre **almeno** due bit di output

## La struttura di una S-box

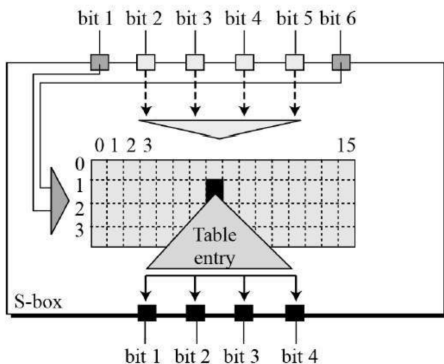
Column Number

Row  
No.

	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  $b_1 \dots b_6$  una generica stringa di input di 6 bit per la S-box.  
 I bit  $b_1 b_6$  indicizzano una riga, i bit  $b_2 b_3 b_4 b_5$  una colonna.



Anche le mixing permutation furono progettate con cura

- i 4 bit di output di qualsiasi *S*-box influenzano l'input di 6 *S*-box nel round successivo



reso possibile dalla presenza della funzione di espansione *E*

La permutazione é

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

La funzione di round del DES fa sí che DES esibisca un forte **effetto valanga**.

Tracciamo le differenze tra i valori intermedi di DES di due input che differiscono in un singolo bit.

$$(L_0, R_0) \quad \text{Assumiamo } R_0 = R'_0 \quad (L'_0, R'_0)$$

↑

Il bit differente é in  $L_0, L'_0$

Dopo il primo round:

$$(L_1, R_1) \quad (L'_1, R'_1)$$

Differenti in un bit in  $R_1, R'_1$  essendo,  $R_1 = L_0 \oplus f(R_0)$ ,  $R'_1 = L'_0 \oplus f(R_0)$

Dopo il secondo round:

$$(L_2, R_2)$$

$$(L'_2, R'_2)$$

$L_2, L'_2$  differiscono in un bit poiché  $L_2 = R_1, L'_2 = R'_1$

Assumiamo inoltre che il bit in cui  $R_1$  ed  $R'_1$  differiscono **non** venga duplicato da  $E$ .

Per la proprietà 3. delle  $S$ -box, l'output dell' $S$ -box in cui il bit è differente è **diverso** in almeno due bit  $\Rightarrow R_2$  ed  $R'_2$  differiscono in **almeno** due bit



$(L_2, R_2)$  ed  $(L'_2, R'_2)$  differiscono in **almeno** tre bit

La mixing permutation diffonde la differenza di due bit in  $R_2$  ed  $R'_2$ .

Ciascuno dei due bit viene usato come input in diverse  $S$ -box.

Dopo il terzo round:

$$(L_3, R_3)$$

$$(L'_3, R'_3)$$

$L_3, L'_3$  differiscono in almeno due bit poiché  $L_3 = R_2, L'_3 = R'_2$

$R_3, R'_3$  differiscono in almeno quattro bit per la prop. 3 della S-box  
(se  $E$  duplica uno o entrambi i bit in cui  $R_2$  ed  $R'_2$  differiscono, allora  $R_3, R'_3$  possono differire in più di 4 bit)



Come per le SPN, abbiamo un effetto esponenziale: dopo 7 round ci aspettiamo che tutti i 32 bit della parte destra siano influenzati dalla differenza di un bit nei due input.

DES ha 16 round. Assicurano che su input simili gli output sembrano indipendenti.

# Attacchi contro DES ridotto

Consideriamo attacchi contro DES ad un round, a due round ed a tre round.

Nessuna di queste varianti può essere pseudocasuale. Non c'è effetto valanga.

Consideriamo attacchi di recupero della chiave di tipo known-plaintext.

*Adv* conosce  $\{(x_i, y_i)\}_i$ , dove  $y_i = DES_K(x_i)$

ONE round. Data la coppia  $(x, y)$  risulta

$$y = (L_1, R_1) \quad x = (L_0, R_0)$$

Poiché  $R_0 = L_1$  ed  $R_1 = L_0 \oplus f_1(R_0)$ , possiamo ricavare  $f_1(R_0) = L_0 \oplus R_1$

$\Rightarrow$  conosciamo una **coppia** input/output per  $f_1$ , ovvero  $(R_0, L_0 \oplus R_1)$ .

# Attacchi contro DES ridotto

Applicando l'inversa della mixing permutation ad  $L_0 \oplus R_1$  otteniamo le stringhe di output delle  $S$ -box

Ci sono 4 possibili valori di input (di 6 bit) **per ognuna** delle  $S$ -box

L'input alle  $S$ -box é l'xor tra  $E(R_0)$  e  $K_1$ .

Essendo  $R_0$  noto, risulta  $E(R_0)$  noto  $\Rightarrow$  possiamo calcolare 4 **possibili valori** per ciascuna porzione di 6 bit di  $K_1$

- semplicemente, **per ognuna** delle  $S$ -box, calcoliamo l'xor tra ognuno dei 4 possibili input alla  $S$ -box e la porzione di  $E(R_0)$  corrispondente

Abbiamo ridotto il numero di valori possibili per la sottochiave  $K_1$  da  $2^{48}$  a  $4^8 = 2^{16}$

Un'altra coppia  $(x', y')$  é sufficiente per individuare quella giusta tra le  $2^{16}$ .

## Attacchi contro DES ridotto

TWO round. Data una coppia  $(x, y)$  sono noti:  $(L_0, R_0)$  ed  $(L_2, R_2)$ .  
D'altra parte

$$L_1 = R_0, \quad R_1 = L_0 \oplus f_1(R_0), \quad L_2 = R_1, \quad \text{ed} \quad R_2 = L_1 \oplus f_2(R_1)$$

per cui conosciamo anche  $L_1$  ed  $R_1$

⇒ disponiamo di una coppia input/output per  $f_1$ , ovvero  $(R_0, L_0 \oplus R_1)$

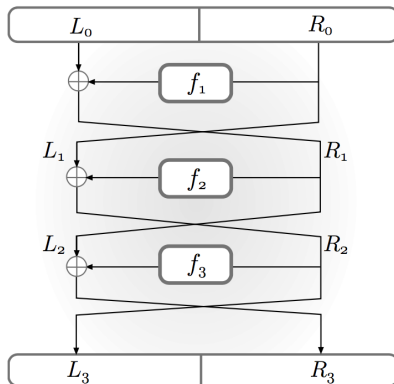
⇒ disponiamo di una coppia input/output per  $f_2$ , ovvero  $(R_1, L_1 \oplus R_2)$

Pertanto, lo stesso metodo usato nel caso precedente può essere utilizzato per determinare  $K_1$  e  $K_2$  in tempo circa  $2 \cdot 2^{16}$

Nota: l'attacco funziona anche se  $K_1$  e  $K_2$  sono chiavi totalmente indipendenti.



THREE round. Più complicato. Diversa strategia di attacco.



# Attacchi contro DES ridotto

Data una coppia  $(x, y)$ , conosciamo  $(L_0, R_0)$  ed  $(L_3, R_3)$ .

Poiché  $L_1 = R_0$  ed  $R_2 = L_3$ , le uniche incognite sono  $R_1$  ed  $L_2$  (che sono uguali).

Pertanto, **non** disponiamo di una coppia input/output per  $f_1, f_2$  ed  $f_3$ , e.g.,

$$R_2 = L_1 \oplus f_2(R_1) \quad \Rightarrow \quad f_2(R_1) = L_1 \oplus R_2 \quad \text{ma } R_1 \text{ non è noto.}$$

Ma conosciamo una **relazione** tra input ed output di  $f_1$  ed  $f_3$ .

Precisamente, risulta

$$f_1(R_0) \oplus f_3(R_2) = (L_0 \oplus R_1) \oplus (L_2 \oplus R_3) = (L_0 \oplus L_2) \oplus (L_2 \oplus R_3) = (L_0 \oplus R_3)$$

Ovvero, l'xor degli output di  $f_1$  ed  $f_3$  è uguale ad  $L_0 \oplus R_3$  (che è noto).

Inoltre,  $R_0$  ed  $R_2$ , input di  $f_1$  ed  $f_3$ , rispettivamente, sono noti.

## Attacchi contro DES ridotto

Possiamo allora determinare gli **input** di  $f_1$  ed  $f_3$  e l'**xor dei loro output**.

Useremo queste informazioni per progettare un attacco di recupero della chiave!

Ricordiamo che la master key in DES  $K$  (56 bit) é  $K = K_L || K_R$ , due metà di 28 bit.

Ciascuna sottochiave  $K_i$  (48 bit) prende 24 bit da  $K_L$  e 24 da  $K_R$ .



$K_L$  influenza i primi 24 bit,  $K_R$  influenza i secondi 24 bit

Proviamo ad indovinare (e a **fissare**) un valore per  $K_L$

Conosciamo  $R_0$  per  $f_1$ . Usiamo  $K_L$  per calcolare l'input per le prime 4 S-box e, quindi, metà dei bit di output di  $f_1$ .

Usando  $R_2$  possiamo calcolare gli stessi bit di output di  $f_3$  per lo stesso valore di  $K_L$

A questo punto possiamo verificare l'ipotesi fatta su  $K_L$

- calcoliamo l'xor tra i bit di output ottenuti valutando  $f_1$  ed  $f_3$
- li confrontiamo con i bit corrispondenti di  $L_0 \oplus R_3$

Indicando con  $\overline{f_1(R_0)}|_{16}$  e  $\overline{f_3(R_2)}|_{16}$  i bit di output calcolati, ed  $\overline{L_0 \oplus R_3}|_{16}$  i bit di  $L_0 \oplus R_3$  nelle **stesse** posizioni, se

$$\overline{f_1(R_0)}|_{16} \oplus \overline{f_3(R_2)}|_{16} \neq \overline{L_0 \oplus R_3}|_{16}.$$

allora l'ipotesi su  $K_L$  è sbagliata e occorre ripetere il processo ipotizzando un nuovo valore per  $K_L$ .

# Attacchi contro DES ridotto

Un'ipotesi corretta **supera sempre** il test. Una scorretta soltanto con prob.  $\approx 2^{-16}$ .

Poiché sono possibili  $2^{28}$  ipotesi per  $K_L$ , alla fine del processo si aspettiamo circa  $2^{28} \cdot 2^{-16} = 2^{12}$  valori possibili per  $K_L$ .

Lo stesso attacco può essere sferrato contro  $K_R$

⇒ in tempo prossimo a  $2 \cdot 2^{28}$  abbiamo circa  $2^{12} \cdot 2^{12} = 2^{24}$  possibili candidati per la master key.

Usando una coppia aggiuntiva  $(x', y')$  è possibile individuare quella giusta


- con una ricerca esaustiva su tutte le  $2^{24}$  possibili master key
- applicando nuovamente l'attacco presentato con le  $2^{12}$  possibili  $K_L$  e le  $2^{12}$  possibili  $K_R$

La complessità dell'attacco é data da

- relativamente all'uso di  $(x, y)$ , circa  $2 \cdot 2^{28}$  passi
- relativamente all'uso di  $(x', y')$ , circa  $2^{24}$  passi per la ricerca esaustiva oppure circa  $2 \cdot 2^{12}$  passi per ripetere l'attacco

In entrambi i casi, non piú di  $2^{30}$  passi, che é molto meno di  $2^{56}$ .

# Sicurezza del DES

15 January	1977	DES is published as a FIPS standard FIPS PUB 46
	1983	DES is reaffirmed for the first time
	1986	<a href="#">Videocipher II</a> , a TV satellite scrambling system based upon DES, begins use by HBO
22 January	1988	DES is reaffirmed for the second time as FIPS 46-1, superseding FIPS PUB 46
July	1991	Biham and Shamir rediscover <a href="#">differential cryptanalysis</a> , and apply it to a 15-round DES-like cryptosystem.
	1992	Biham and Shamir report the first theoretical attack with less complexity than brute force: <a href="#">differential cryptanalysis</a> . However, it requires an unrealistic $2^{47}$ <a href="#">chosen plaintexts</a> .
30 December	1993	DES is reaffirmed for the third time as FIPS 46-2
	1994	The first experimental cryptanalysis of DES is performed using linear cryptanalysis (Matsui, 1994).
June	1997	The <a href="#">DESCHALL Project</a> breaks a message encrypted with DES for the first time in public.
July	1998	The EFF's <a href="#">DES cracker</a> (Deep Crack) breaks a DES key in 56 hours.
January	1999	Together, <a href="#">Deep Crack</a> and <a href="#">distributed.net</a> break a DES key in 22 hours and 15 minutes.
25 October	1999	DES is reaffirmed for the fourth time as FIPS 46-3, which specifies the preferred use of <a href="#">Triple DES</a> , with single DES permitted only in legacy systems.
26 November	2001	The <a href="#">Advanced Encryption Standard</a> is published in FIPS 197
26 May	2002	The AES becomes effective
26 July	2004	The withdrawal of FIPS 46-3 (and a couple of related standards) is proposed in the <i>Federal Register</i> <sup>[19]</sup>
19 May	2005	NIST withdraws FIPS 46-3 (see <a href="#">Federal Register vol 70, number 96</a>  )
April	2006	The FPGA-based parallel machine <a href="#">COPACOBANA</a> of the Universities of Bochum and Kiel, Germany, breaks DES in 9 days at a \$10,000 hardware cost. <sup>[20]</sup> Within a year software improvements reduced the average time to 6.4 days.
Nov.	2008	The successor of <a href="#">COPACOBANA</a> , the RIVYERA machine, reduced the average time to less than a single day.
July	2017	A <a href="#">chosen-plaintext attack</a> utilizing a <a href="#">rainbow table</a> can recover the DES key for a single specific chosen plaintext <code>1122334455667788</code> in 25 seconds. A new rainbow table has to be calculated per plaintext. A limited set of rainbow tables have been made available for download. <sup>[21]</sup>

# Parametri insufficienti oggi

Non solo la lunghezza della chiave é troppo corta per la potenza computazionale attuale.

Anche la lunghezza del blocco in alcune applicazioni é fonte di preoccupazione.

Esempio: la prova di sicurezza che CTR-mode é CPA sicura dipende dalla lunghezza del blocco.

A vince con probabilità  $\frac{2 \cdot q^2}{2^\ell}$  se ottiene  $q$  coppie  $(m, c)$

Pertanto, se  $\ell = 64 \Rightarrow$  per  $q = 2^{30}$  risulta

$$\frac{2 \cdot q^2}{2^\ell} = \frac{2 \cdot (2^{30})^2}{2^\ell} = \frac{2 \cdot 2^{60}}{2^{64}} = \frac{1}{2^3} = 1/8$$

che é una probabilità altissima!



La progettazione del DES é "quasi perfetta". Non si conoscono debolezze strutturali.

I suoi parametri sono solo troppi corti oggi.

In pratica il miglior attacco disponibile é la ricerca esaustiva.

Esistono due tecniche di attacco sofisticate

- crittoanalisi differenziale (Biham e Shamir 1991)
  - attacco chosen plaintext: richiede  $2^{47}$  coppie e rompe in tempo  $2^{37}$
- crittoanalisi lineare (Matsui, 1993)
  - attacco known plaintext: richiede  $2^{43}$  coppie e rompe in tempo  $2^{39}$

Gli attacchi sono piú di interesse teorico che di rilevanza pratica.

## 1 Apportare modifiche interne

- mantenere la funzione di round, incrementare la taglia della master key
- cambiare  $S$ -box ed usare sottochiavi piú lunghe
- svantaggio: **perdiamo** la confidenza che abbiamo guadagnato circa la struttura DES negli anni

## 2 Usare il DES come una scatola nera

- lo consideriamo un cifrario a blocchi perfetto con chiave da 56 bit
- un nuovo cifrario a blocchi lo invoca come subroutine

Sia  $F$  un cifrario a blocchi con chiave di  $n$  bit e lunghezza di blocco di  $\ell$  bit.  
Un nuovo cifrario con chiave di  $2n$  bit può essere definito come segue:

$$F'_{K_1, K_2}(x) \stackrel{\text{def}}{=} F_{K_2}(F_{K_1}(x))$$

Nel caso che  $F$  sia  $DES$ , otterremmo  $2DES$ , con chiave da 112 bit.

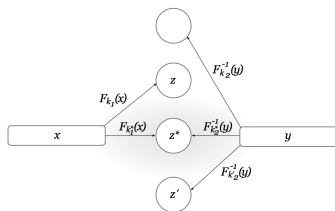
Se la ricerca esaustiva fosse l'attacco migliore, saremmo protetti rispetto ad attacchi di tempo  $\leq 2^{112}$

Purtroppo non è così.

# Attacco contro la cifratura doppia

A dispone di  $(x, y) = (x, F'_{K_1^*, K_2^*}(x))$ . Opera come segue

- Per ogni  $K_1 \in \{0, 1\}^n$ , calcola  $z = F_{K_1}(x)$  e memorizza  $(z, K_1)$  nella lista  $L_1$
- Per ogni  $K_2 \in \{0, 1\}^n$ , calcola  $z = F_{K_2}^{-1}(y)$  e memorizza  $(z, K_2)$  nella lista  $L_2$
- Le coppie  $(z_1, K_1) \in L_1$  e  $(z_2, K_2) \in L_2$  costituiscono un **match** se  $z_1 = z_2$ . Per ciascun match memorizza  $(K_1, K_2)$  nella lista  $S$



# Attacco contro la cifratura doppia

L'attacco richiede

Tempo  $O(n \cdot 2^n)$

Spazio  $O((n + \ell) \cdot 2^n)$

La lista  $S$  contiene  $(K_1, K_2)$  tali che  $F_{K_1}(x) = F_{K_2}^{-1}(y)$

Una coppia  $(K_1, K_2) \neq (K_1^*, K_2^*)$  soddisfa l'equazione con prob.  $2^{-\ell}$ , trattando  $F_K$  ed  $F_K^{-1}$  come funzioni totalmente casuali



La taglia attesa della lista  $S$  è  $2^{2n} \cdot 2^{-\ell} = 2^{2n-\ell}$

Usando altre poche coppie  $(x_i, y_i)$ ,  $S$  viene sfoltita e la chiave  $(K_1^*, K_2^*)$  correttamente individuata.

In conclusione, *2DES* non viene usato.

Vengono usate due varianti

- 1 Tre chiavi  $K_1, K_2$  e  $K_3$  indipendenti

$$F''_{K_1, K_2, K_3}(x) \stackrel{\text{def}}{=} F_{K_3}(F_{K_2}^{-1}(F_{K_1}(x)))$$

- 2 Due chiavi  $K_1$  e  $K_2$  indipendenti

$$F''_{K_1, K_2}(x) \stackrel{\text{def}}{=} F_{K_1}(F_{K_2}^{-1}(F_{K_1}(x)))$$

L'invocazione centrale di  $F_K^{-1}$  è per ragioni di compatibilità, i.e.,

$$K_1 = K_2 = K_3 \quad \Rightarrow \quad 3DES_{K_1, K_2, K_3} = DES_{K_1}$$

D'altra parte, se  $F$  é una SPRP, allora  $F^{-1}$  é una SPRP e quindi non ci sono problemi in generale in questo uso.

**Sicurezza della prima variante:** la chiave é lunga  $3n$  bit ma, per l'attacco precedente contro la cifratura doppia - che si applica anche qui - otteniamo sicurezza rispetto ad attacchi di tempo  $\leq 2^{2n}$ .

**Sicurezza della seconda variante:** la chiave é lunga  $2n$  bit. Al momento non si conoscono attacchi di complessitá migliore di  $2^{2n}$ , dato soltanto un piccolo numero di coppie  $(x, y)$

É una buona scelta in pratica

**Svantaggi:** lunghezza di blocco relativamente piccola e cifrario complessivamente lento (3 invocazioni di  $DES$ ).

# Advanced Encryption Standard (AES)

Cifrario a blocchi, adottato ufficialmente dal NIST nel 2001.

Prescelto tra 15 algoritmi finalisti nella call per sicurezza, efficienza in implementazioni HW e SW e per flessibilità.

Ad oggi, non sono noti attacchi crittoanalitici migliori della ricerca esaustiva

Ha lunghezza di blocco di 128 bit. Può usare chiavi di 128, 192 e 256 bit

La lunghezza della chiave

- influisce sulla schedulazione delle chiavi e sul numero di round
- **non** influisce sulla struttura di alto livello di ciascun round

AES é essenzialmente una rete SPN

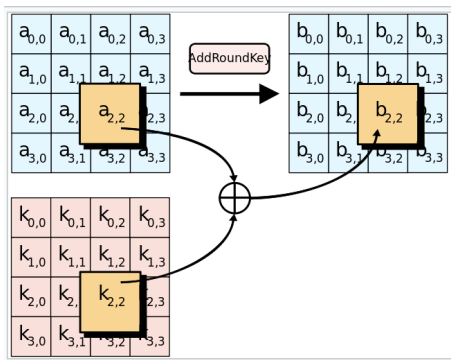
Un array di 16 byte di taglia  $4 \times 4$  (lo stato di AES) viene modificato durante la computazione.



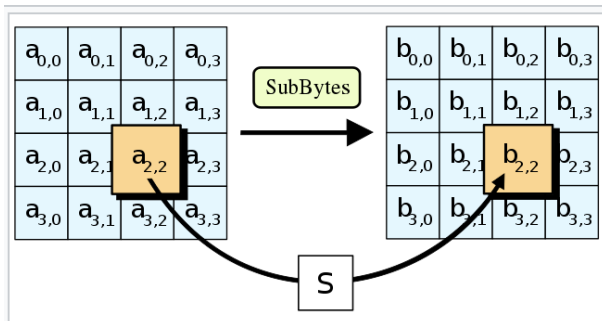
# AES: Struttura

Lo stato iniziale corrisponde all'input. La struttura di ogni round é costituita da 4 passi.

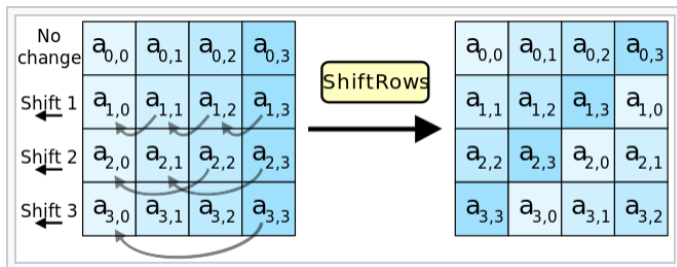
**Passo 1: AddRoundKey.** Una sottochiave di round di 128 bit viene "aggiunta" tramite xor allo stato



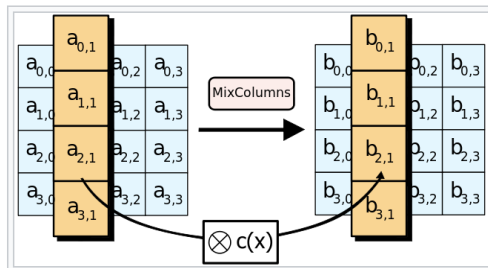
**Passo 2: SubBytes.** Ciascun byte dello stato viene sostituito con un altro byte, in accordo ad una tabella di lookup  $S$ .



**Passo 3: ShiftRows.** I byte di ciascuna riga vengono traslati (shift) a sinistra. Riga  $i \rightarrow$  shift di  $i$  posizioni.



**Passo 4: MixColumns.** Una trasformazione invertibile viene applicata ai 4 byte di ogni colonna (moltiplicazione matriciale).



**Proprietá:** se due input differiscono in  $b > 0$  byte, allora applicando la trasformazione, i due output differiscono in almeno  $5 - b$  byte.

Nel round iniziale viene effettuato soltanto il passo **AddRoundKey**.

Nei round intermedi i passi sono **SubBytes**, **ShiftRows**, **MixColumns** e **AddRoundKey**.

Nel round finale, **MixColumns** non viene eseguito.

Si noti che i passi 3 e 4 corrispondono al mixing step in una rete SPN.

Circa il numero di round, una chiave da

- 128 bit  $\Rightarrow$  10 round    192 bit  $\Rightarrow$  12 round    256 bit  $\Rightarrow$  14 round

*AES* é una scelta eccellente d'uso in tutti gli schemi che richiedono una permutazione pseudocasuale forte (SPRP).

Federal Information Processing Standards (FIPS).

Riferimenti: DES

[https://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard)

<https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>

Riferimenti: AES

[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf>