

RC2: mashing round

$R[0], R[1], R[2], R[3]$

Mash R[0] — $K[R[0] \& 63]$

Mash R[1] — $K[R[1] \& 63]$

Mash R[2] — $K[R[2] \& 63]$

Mash R[3] — $K[R[3] \& 63]$

$R[0], R[1], R[2], R[3]$

AES 6

Mash R[i]

$R[i] \leftarrow R[i] + K[R[i-1] \& 63]$

AES 7

RC2: mashing round

$R[0] \leftarrow R[0] + K[R[3] \& 63]$

$R[1] \leftarrow R[1] + K[R[0] \& 63]$

$R[2] \leftarrow R[2] + K[R[1] \& 63]$

$R[3] \leftarrow R[3] + K[R[2] \& 63]$

AES 8

RC2: espansione chiave

chiave nei byte $L[0], \dots, L[T-1]$ $1 \leq T \leq 128$

$P[0, \dots, 127] \leftarrow$ espansione binaria π

for $i=T$ **to** 127 **do**

$L[i] \leftarrow P[L[i-1] + L[i-T]]$

$L[128-T] \leftarrow P[L[128-T]]$

for $i=127-T$ **downto** 0 **do**

$L[i] \leftarrow P[L[i+1] \oplus L[i+T]]$

sottochiavi byte $L[0], L[1], \dots, L[127]$

ovvero word 16 bit $K[0], \dots, K[63]$

$K[i] = L[2i] + 256 \cdot L[2i+1]$

AES 9


Tabella P

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	d9	78	f9	c4	19	dd	b5	ed	28	e9	fd	79	4a	a0	d8	9d
10:	c6	7e	37	83	2b	76	53	8e	62	4c	64	88	44	8b	fb	a2
20:	17	9a	59	f5	87	b3	4f	13	61	45	6d	8d	09	81	7d	32
30:	bd	8f	40	eb	86	b7	7b	0b	f0	95	21	22	5c	6b	4e	82
40:	54	d6	65	93	ce	60	b2	1c	73	56	c0	14	a7	8c	f1	dc
50:	12	75	ca	1f	3b	be	e4	d1	42	3d	d4	30	a3	3c	b6	26
60:	6f	bf	0e	da	46	69	07	57	27	f2	1d	9c	bc	94	43	03
70:	f8	11	c7	f6	90	ef	3e	e7	06	c3	d5	2f	c8	66	1e	d7
80:	08	e8	ea	de	80	52	ee	f7	84	aa	72	ac	35	4d	6a	2a
90:	96	1a	d2	71	5a	15	49	74	4b	9f	d0	5e	04	18	a4	ec
a0:	c2	e0	41	6e	0f	51	cb	cc	24	91	af	50	a1	f4	70	39
b0:	99	7c	3a	85	23	b8	b4	7a	fc	02	36	5b	25	55	97	31
c0:	2d	5d	fa	98	e3	8a	92	ae	05	df	29	10	67	6c	ba	c9
d0:	d3	00	e6	cf	e1	9e	a8	2c	63	16	01	3f	58	e2	89	a9
e0:	0d	38	34	1b	ab	33	ff	b0	bb	48	0c	5f	b9	b1	cd	2e
f0:	c5	f3	db	47	e5	a5	9c	77	0a	a6	20	68	fe	7f	c1	ad

AES 10

RC2

Esercizio:
Decifratura?



AES 11

RC5

Ron Rivest [1994]

- ❑ Algoritmo semplice
- ❑ Orientato alla parola macchina
- ❑ Usa operazioni comuni dei processori
- ❑ Parametrizzato
 - Lunghezza parola macchina
 - Numero iterazioni
 - Lunghezza chiave
- ❑ Usa poca memoria (smart card e altre device)
- ❑ Rotazioni data-dependent

AES 12

RC5-w/r/b

$w = 16, 32, 64$
 $r = 0, 1, \dots, 255$
 $b = 0, 1, \dots, 255$

AES 13

RC5

Operazioni su parole di w bit:

- $a+b$ somma modulo 2^w
- $a-b$ sottrazione modulo 2^w
- $a \oplus b$ XOR bit a bit
- $a \ll b$ shift a sinistra di a di un numero di bit dato dai log w bit meno significativi di b
- $a \gg b$ shift a destra di a di un numero di bit dato dai log w bit meno significativi di b

AES 14

RC5: cifratura

Input: testo in chiaro (A,B)
 Chiave schedulata: $S[0, \dots, 2r+1]$

```

A ← A + S[0]
B ← B + S[1]
for i = 1 to r do
    A ← ((A ⊕ B) ≪ B) + S[2i]
    B ← ((B ⊕ A) ≪ A) + S[2i+1]
    
```

Output: testo cifrato (A,B)

AES 15

RC5: decifratura

```

A ← A + S[0]
B ← B + S[1]
for i = 1 to r do
    A ← ((A ⊕ B) ≪ B) + S[2i]
    B ← ((B ⊕ A) ≪ A) + S[2i+1]
    
```

cifratura

```

for i = r downto 1 do
    B ← ((B - S[2i+1]) ≫ A) ⊕ A
    A ← ((A - S[2i]) ≫ B) ⊕ B
    B ← B - S[1]
    A ← A - S[0]
    
```

decifratura

AES 16

RC5: schedulazione chiave

Chiave $K [0, \dots, b-1]$ di b byte

Se 8b non è multiplo di w padding con 00...0

$L [0, \dots, c-1]$ è un array di $c = \lceil 8b/w \rceil$ parole di w bit

↓

Mixing function

↓

$S [0, \dots, 2r+1]$ chiave schedulata

AES 17

RC6: schedulazione chiave

Inizializzazione array S

```

S[0] = P_w
for i = 1 to 2r+1 do
    S[i] ← S[i-1]+Q_w
X ← Y ← 0
i ← j ← 0
do 3·max(c,2r+1) times
    X ← S[i] ← (S[i]+X+Y) « 3
    Y ← L[j] ← (L[j]+X+Y) « (X+Y)
    i ← (i+1) mod (2r+1)
    j ← (j+1) mod c
    
```

3 passi sul più grande array
più passi sul più piccolo

AES 18

Costanti magiche

P_w = espansione binaria numero di Nepero
 $e = 2.71828182459045...$ (decimale) $P_w = \text{Odd}[(e-2)2^w]$

Q_w = espansione binaria rapporto aureo $Q_w = \text{Odd}[(\phi-1)2^w]$
 $\phi = (1+\sqrt{5})/2 = 1.61803398874989...$ (decimale)

w	16 bit	32 bit	64 bit
P_w	b7 e1	b7 e1 51 63	b7 e1 51 62 8a ed 2a 6b
Q_w	9E 37	9E 37 79 b9	9E 37 79 b9 7f 4a 7c 15

AES 19

International Data Encryption Algorithm (IDEA)

Xuejia Lai, James Massey 1990, 1991

testo in chiaro (64 bit) → IDEA → testo cifrato (64 bit)

chiave (128 bit) ↑

- ☐ Più efficiente su processori a 16 bit
- ☐ Usato nel PGP

AES 20

Blowfish

Bruce Schneier 1993, 1994

testo in chiaro (64 bit) → Blowfish → testo cifrato (64 bit)

chiave (32,...,448 bit) ↑

- ☐ velocità cifratura: 18 cicli di clock per byte su processori a 32 bit
- ☐ memoria usata <5K

AES 21

Altri cifrari a blocchi

SAFER (Secure And Fast Encryption Routine)
 SAFER K-64 [1994], SAFER K-128 [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, MMB, GOST, ...

... AES

AES 22