

# ADVANCED ENCRYPTION STANDARD

KULIAH KRIPTOGRAFI DAN KEAMANAN JARINGAN

# SEJARAH AES

- **3DES bukan merupakan kandidat yang bagus untuk digunakan dalam waktu lama.**
- **NIST pada 1997 mengeluarkan call for proposals untuk Advanced Encryption Standard (AES) baru:**
  - Harus memiliki keamanan yang sama atau lebih baik daripada 3DES
  - Memperbaiki efisiensi secara signifikan
  - Harus berupa symmetric block cipher dengan panjang blok 128 bit
  - Mendukung penggunaan kunci 128, 192, dan 256 bit.
  - Kriteria evaluasi meliputi evaluation keamanan, efisiensi secara komputasi, penggunaan memori, kesesuaian hardware dan software, dan fleksibilitas.
- **NIST mempublikasikan standard final (FIPS PUB 197) pada November 2001.**
  - Rijndael sebagai algoritma AES
  - Dibuat oleh 2 orang kriptografer dari Belgia: Dr. Joan Daemen dan Dr. Vincent Rijmen.

# OVERVIEW AES

- **Operasi dilakukan untuk byte (8-bit).**
  - Operasi aritmetika penjumlahan, perkalian, pembagian menggunakan  $GF(2^8)$ .
  - Irreducible polynomial:  $m(x) = x^8 + x^4 + x^3 + x + 1$
- **Menggunakan blok dengan ukuran 128 bit**
- **Panjang kunci bisa 128, 192, atau 256 bit → 128 bit merupakan panjang yang biasanya dipakai.**
- **Block direpresentasikan sebagai matriks persegi dari bytes.**
  - Block dikopi ke State array, yang kemudian dimodifikasi pada setiap tahap enkripsi atau dekripsi.
  - Setelah tahap terakhir, State dikopi ke matriks output.
- **Kunci 128-bit direpresentasikan sebagai matriks persegi dari byte.**
  - Kunci di-expand menjadi array dari key schedule words: setiap word terdiri dari 4 byte dan jumlah total key schedule adalah 44 words untuk kunci 128-bit.

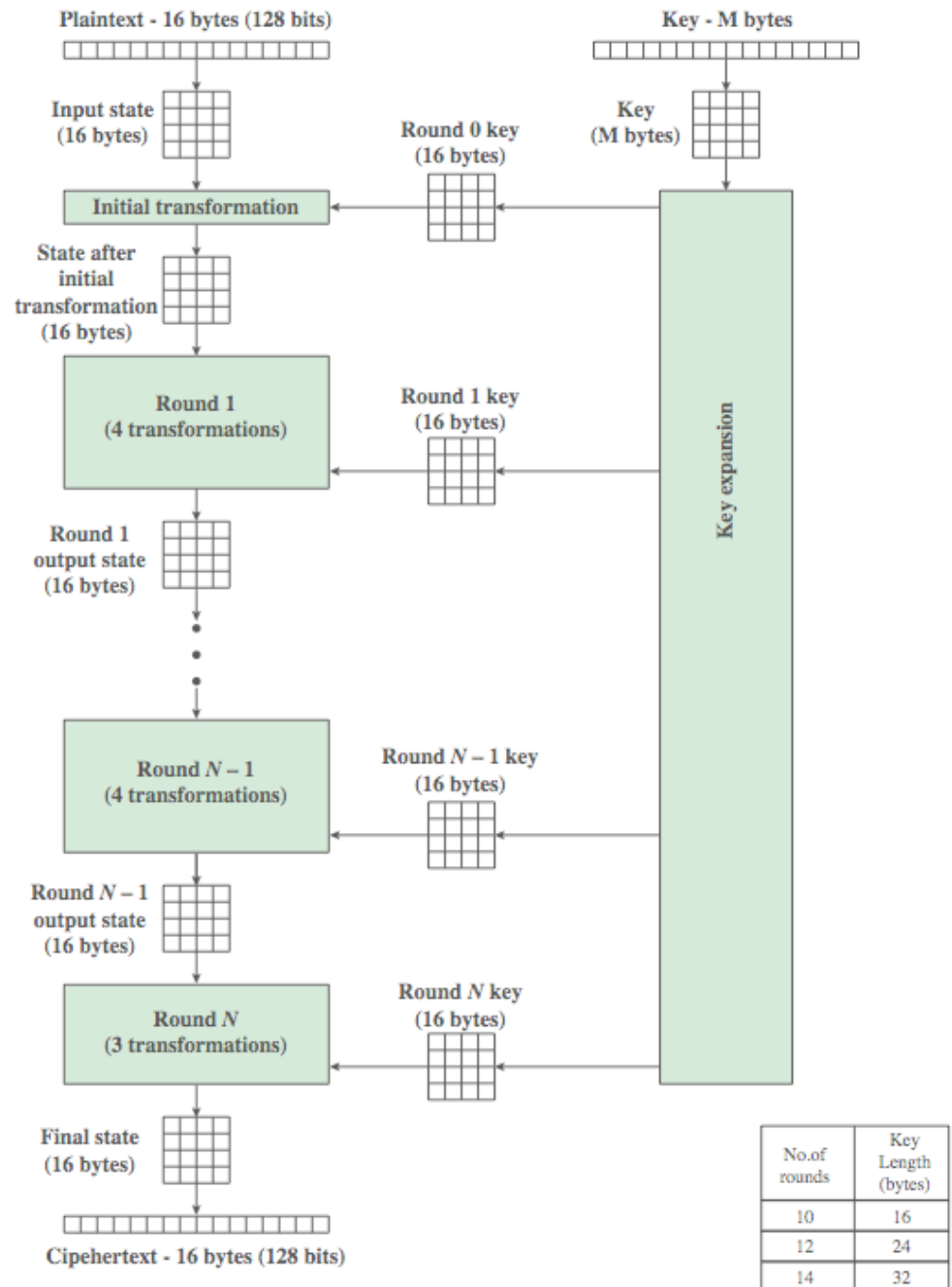
- **Pengurutan bytes dalam matriks berdasarkan kolom.**
  - Contoh: 4 byte pertama pada 128-bit plaintext input ditempatkan pada kolom pertama pada matriks, 4 byte kedua menempati kolom kedua, dst. Pada key, 4 byte pertama dari expanded key menempati kolom pertama pada matriks w.
- **Perbedaan dengan Feistel cipher:**
  - Memproses data sebagai blok yang terdiri dari 4 kolom masing-masing terdiri dari 4 byte.
  - Beroperasi pada seluruh blok data pada setiap tahap (round)
- **Didesain untuk memiliki:**
  - Ketahanan terhadap serangan yang diketahui
  - Kecepatan dan keringkasan kode pada CPU
  - Kemudahan desain

# POLINOMIAL DI AES

- Menggunakan  $\text{GF}(2^8)$ .
- Irreducible polynomial:  $m(x) = x^8 + x^4 + x^3 + x + 1$ .
- Contoh:
  - $f(x) = x^6 + x^4 + x^2 + x + 1$
  - $g(x) = x^7 + x + 1$
  - $f(x) + g(x) = x^7 + x^6 + x^4 + x^2$
  - $f(x) \times g(x) = (x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
  - $x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1) / (x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + 1$
- Maka:  $f(x) \times g(x) \bmod m(x) = x^7 + x^6 + 1$

# PROSES ENKRIPSI

- Sumber: Stallings (2011)

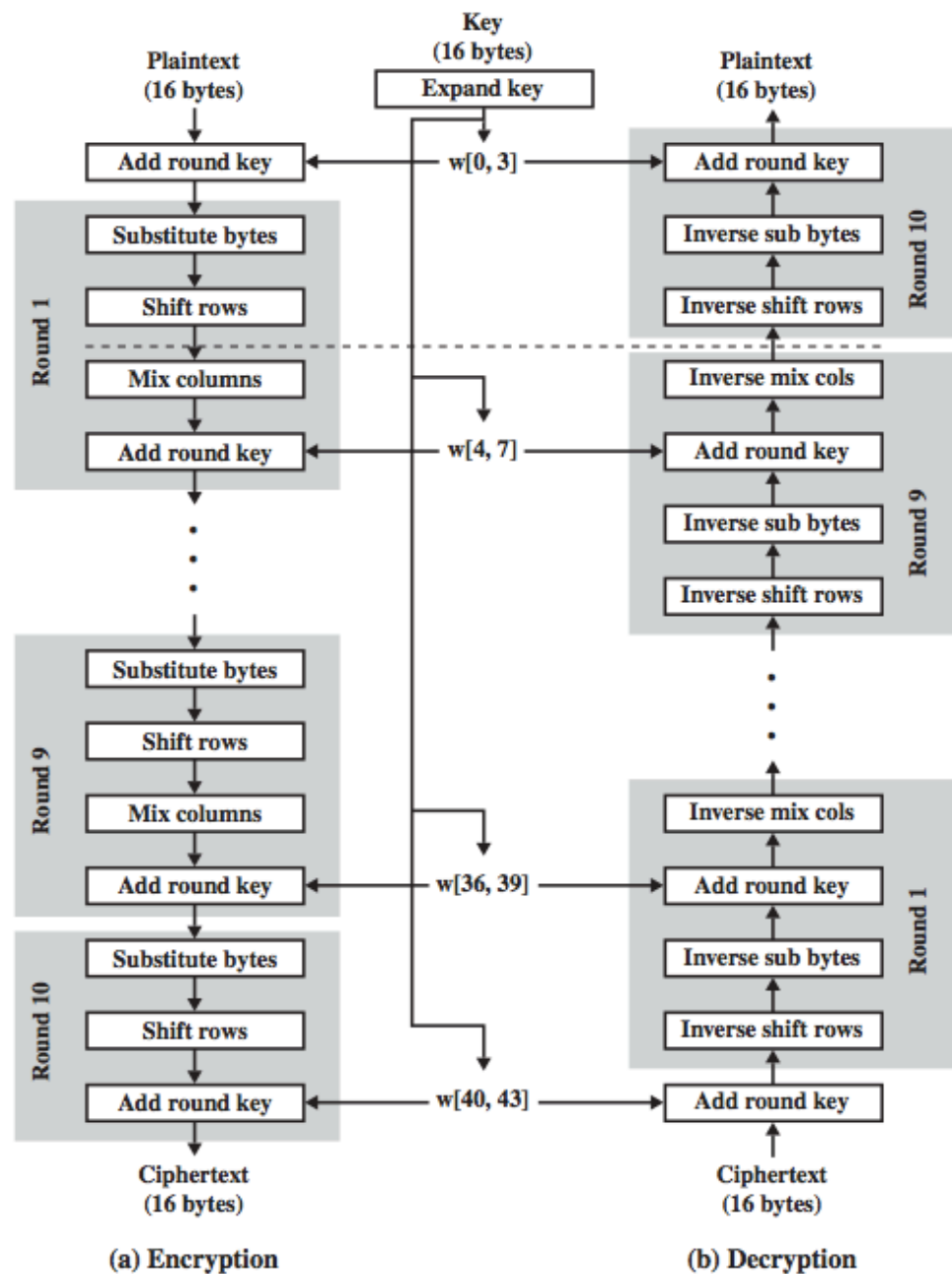


# STRUKTUR AES

- **Blok data 4 kolom masing-masing terdiri dari 4 byte merupakan state**
- **Kunci di-expand menjadi array of words**
- **Memiliki 9/11/13 tahap, di mana setiap tahapnya (kecuali tahap terakhir) state mengalami:**
  - byte substitution (1 S-box digunakan pada setiap byte)
  - shift rows (permutasi bytes antara group/kolom)
  - mix columns (substitusi menggunakan matriks)
  - add round key (XOR pada state dengan kunci)

# SKEMA AES

Sumber: Stallings (2011)





# BEBERAPA HAL UMUM

- **AES bukan merupakan struktur Feistel.**
  - Pada struktur Feistel klasik, separuh blok data digunakan untuk memodifikasi separuh blok yang lain, kemudian ditukar. AES memproses keseluruhan blok data secara paralel pada setiap tahap menggunakan substitusi dan permutasi.
- **Kunci yang diinputkan di-expand menjadi array yang berupa empat puluh empat 32-bit words,  $w[i]$ .**
  - 4 word yang berbeda (128 bits) dipakai sebagai kunci untuk setiap tahap.
- **Dilakukan 4 fase pemrosesan, yaitu 1 permutasi dan 3 substitusi:**
  - Byte substitution: menggunakan tabel yang disebut S-box untuk mensubstitusi blok byte demi byte.
  - Shift rows: permutasi yang dilakukan baris demi baris.
  - Mix columns: substitusi yang mengganti setiap byte pada kolom sebagai fungsi semua byte pada kolom tersebut.
  - Add round key: operasi XOR pada bit untuk blok dengan sebagian dari expanded key.

- **Struktur AES sederhana.**
  - Untuk enkripsi dan dekripsi:
    - Dilakukan fase AddRoundKey
    - Dilakukan 9 tahap pemrosesan yang masing-masing memuat 4 fase pemrosesan.
    - Tahap ke-10 meliputi 3 fase pemrosesan.
- **Hanya fase AddRoundKey yang menggunakan kunci.**
  - Enkripsi berawal dan berakhir pada fase AddRoundKey.
  - Fase-fase yang lain reversible tanpa perlu mengetahui kunci sehingga tidak menambah keamanan.
- **Enkripsi dapat dipandang sebagai operasi enkripsi XOR (Add Round Key) pada block, diikuti dengan operasi pengocokan block (3 fase lain), diikuti dengan enkripsi XOR, dst. Skema ini efisien dan aman.**

- **Setiap tahap reversible.**
  - Untuk fase-fase Substitute Byte, Shift Row, dan Mix Columns, fungsi invers digunakan pada algoritma dekripsi.
  - Untuk fase Add Round Key, invers didapatkan dari operasi XOR antara blok dan kunci yang sama untuk blok tersebut menggunakan teorema:  $A \oplus B \oplus B = A$ .
- **Algoritma dekripsi menggunakan expanded key dengan urutan terbalik.**
  - Algoritma dekripsi tidak identik dengan algoritma enkripsi.

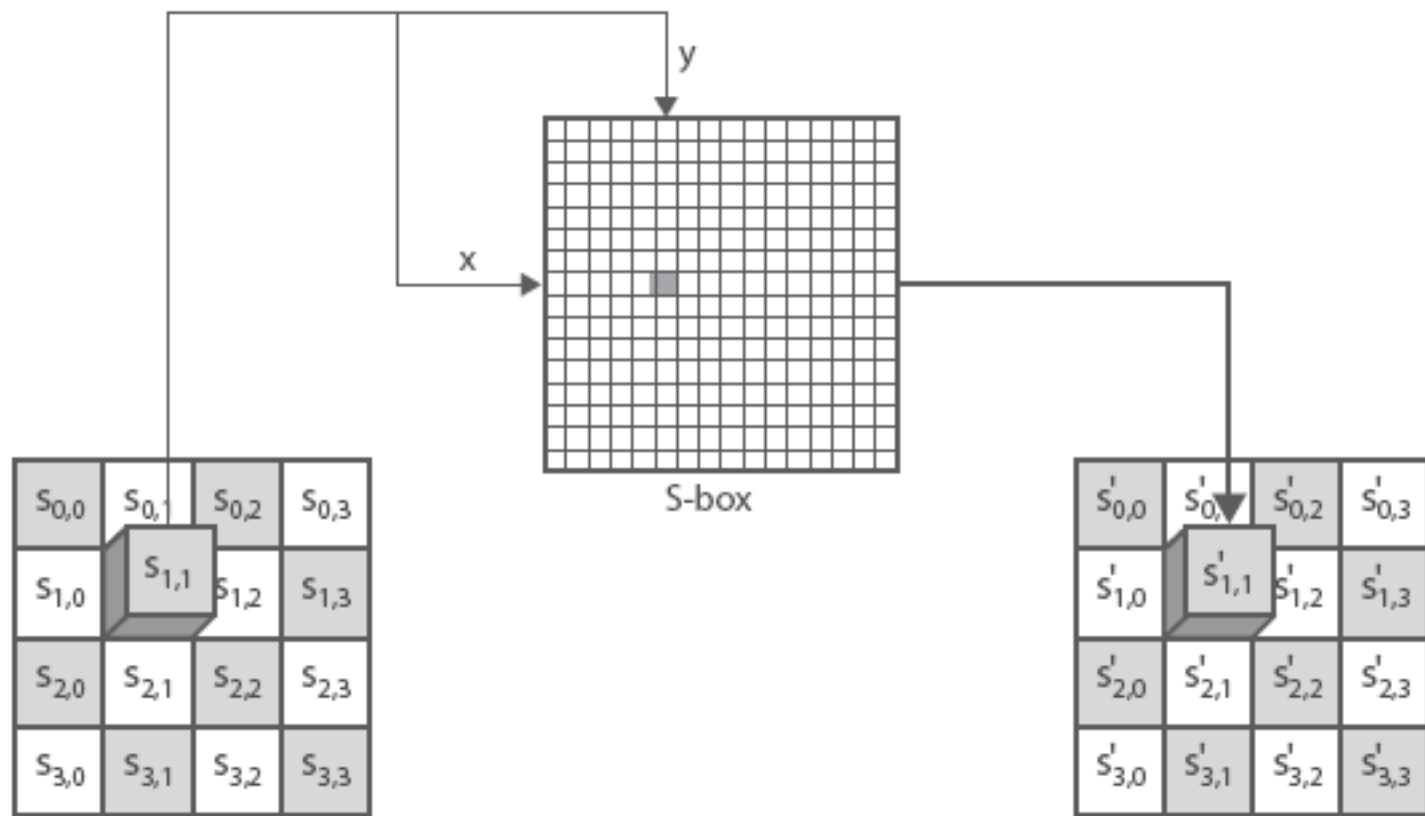
# TRANSFORMASI SUBSTITUTE BYTES

- **Forward substitute bytes hanya merupakan operasi table look-up.**
- **AES memiliki matriks 16x16 yang disebut S-box yang memuat permutasi semua 256 nilai yang mungkin untuk 8-bit.**
- **Setiap byte pada State array dipetakan ke byte yang baru dengan cara:**
  - 4 bit paling kiri digunakan untuk menunjukkan baris pada S-box, 4 bit paling kanan menunjukkan kolom pada S-box.
  - Contoh:  $95_H$  menunjuk pada nilai di baris ke 9 kolom ke 5 dari S-box, yang memuat nilai  $2A_H$ . Maka  $95_H$  dipetakan ke  $2A_H$ .

# S-BOX


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

- **S-box disusun dengan cara sbb:**
  - S-box diinisialisasi dengan nilai-nilai byte baris demi baris.
  - Petakan setiap byte ke multiplicative invers-nya pada  $GF(2^8)$ .
  - Lakukan operasi bit-per-bit pada setiap byte pada S-box dengan suatu rumus tertentu (meliputi operasi modulo, penjumlahan, dan XOR).
- **Invers dari operasi substitute bytes menggunakan invers dari table S-box.**



# CONTOH SUBSTITUTE BYTE

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5



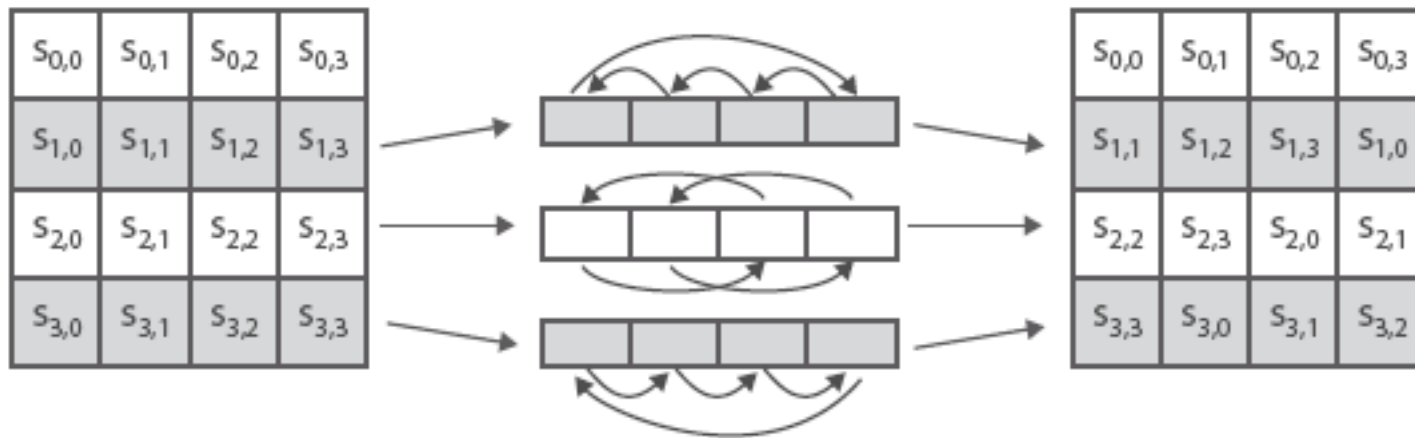
87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6



# TRANSFORMASI SHIFTROWS

- **Operasi forward:**
  - Baris pertama pada State tidak digeser.
  - Baris kedua dikenai operasi 1-byte circular left shift.
  - Baris ketiga dikenai operasi 2-byte circular left shift.
  - Baris keempat dikenai operasi 3-byte circular left shift.
- **Operasi invers:**
  - Menggunakan kebalikan operasi forward.
  - Contoh: baris kedua dikenai operasi 1-byte circular right shift.

# ILUSTRASI DAN CONTOH



87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

# TRANSFORMASI MIXCOLUMN

- **Forward mixcolumn:**

- Beroperasi pada setiap kolom secara individual.
- Setiap byte pada kolom dipetakan ke nilai baru yang merupakan fungsi dari semua nilai pada kolom tersebut.
- Merupakan perkalian State dengan suatu matriks persegi.

02 03 01 01

01 02 03 01

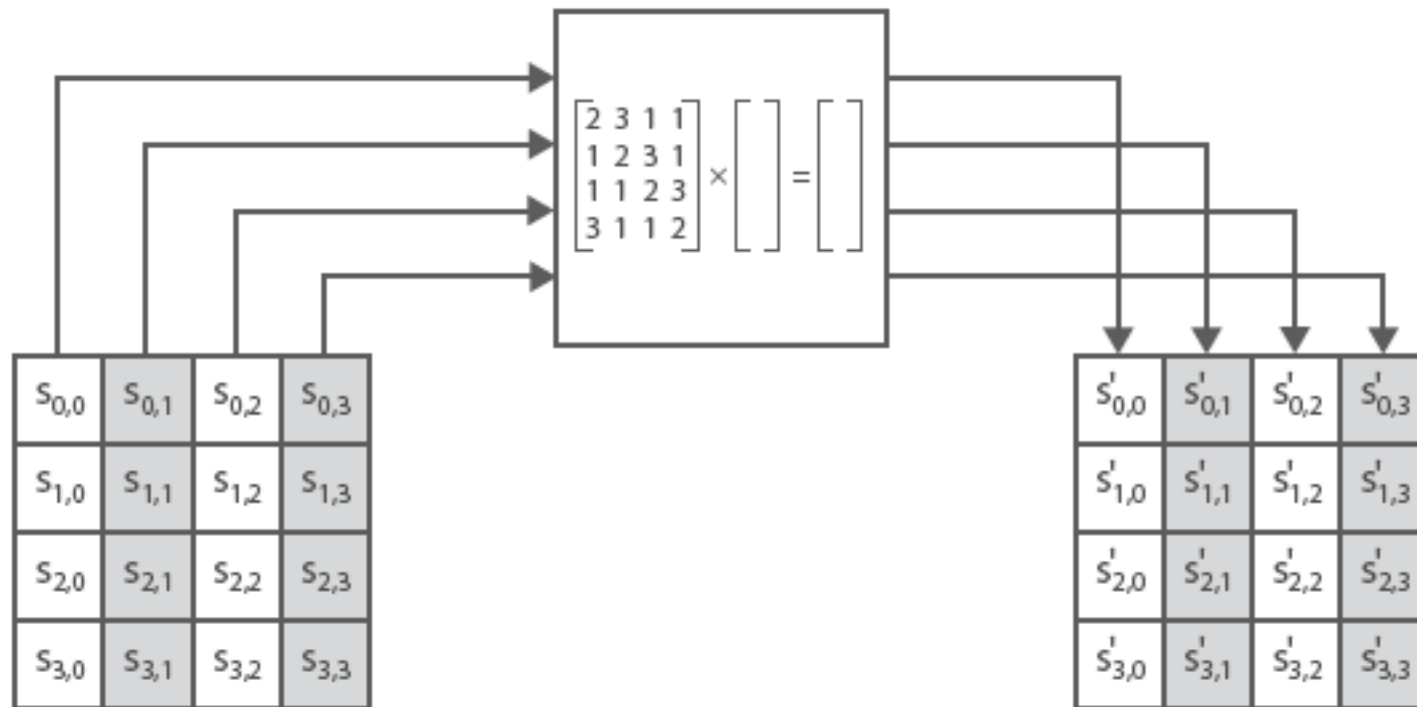
01 01 02 03

03 01 01 02

- Penjumlahan dan perkalian dilakukan pada GF(2<sup>8</sup>).

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

# ILUSTRASI



# CONTOH

87	F2	4D	97		47	40	A3	4C
6E	4C	90	EC		37	D4	70	9F
46	E7	4A	C3	→	94	E4	3A	42
A6	8C	D8	95		ED	A5	A6	BC

$$(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

$$\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\}$$

$$\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\}$$

$$(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\}$$

# ARITMETIKA AES

- Menggunakan aritmetika finite field  $GF(2^8)$
- Irreducible polynomial

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

Yang berupa (100011011)

- **Contoh :**

$$\begin{aligned}\{02\} \cdot \{87\} \bmod (100011011) &= (1\ 0000\ 1110) \bmod (100011011) \\ &= (1\ 0000\ 1110) \text{ xor } (1\ 0001\ 1011) \\ &= (0001\ 0101)\end{aligned}$$

- **Invers mixcolumn:**

- Menggunakan invers dari matriks mixcolumn.

# TRANSFORMASI ADDROUNDKEY

- **Forward AddRoundKey:**
  - 128 bit state di-XOR-kan dengan 128 bit kunci pada tahap tersebut.
  - Dapat dipandang sebagai operasi kolom antara 4 byte State dengan 1 word kunci.
- **Invers:**
  - Sama dengan forward AddRoundKey karena operasi XOR inversnya adalah operasi XOR itu sendiri.



$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

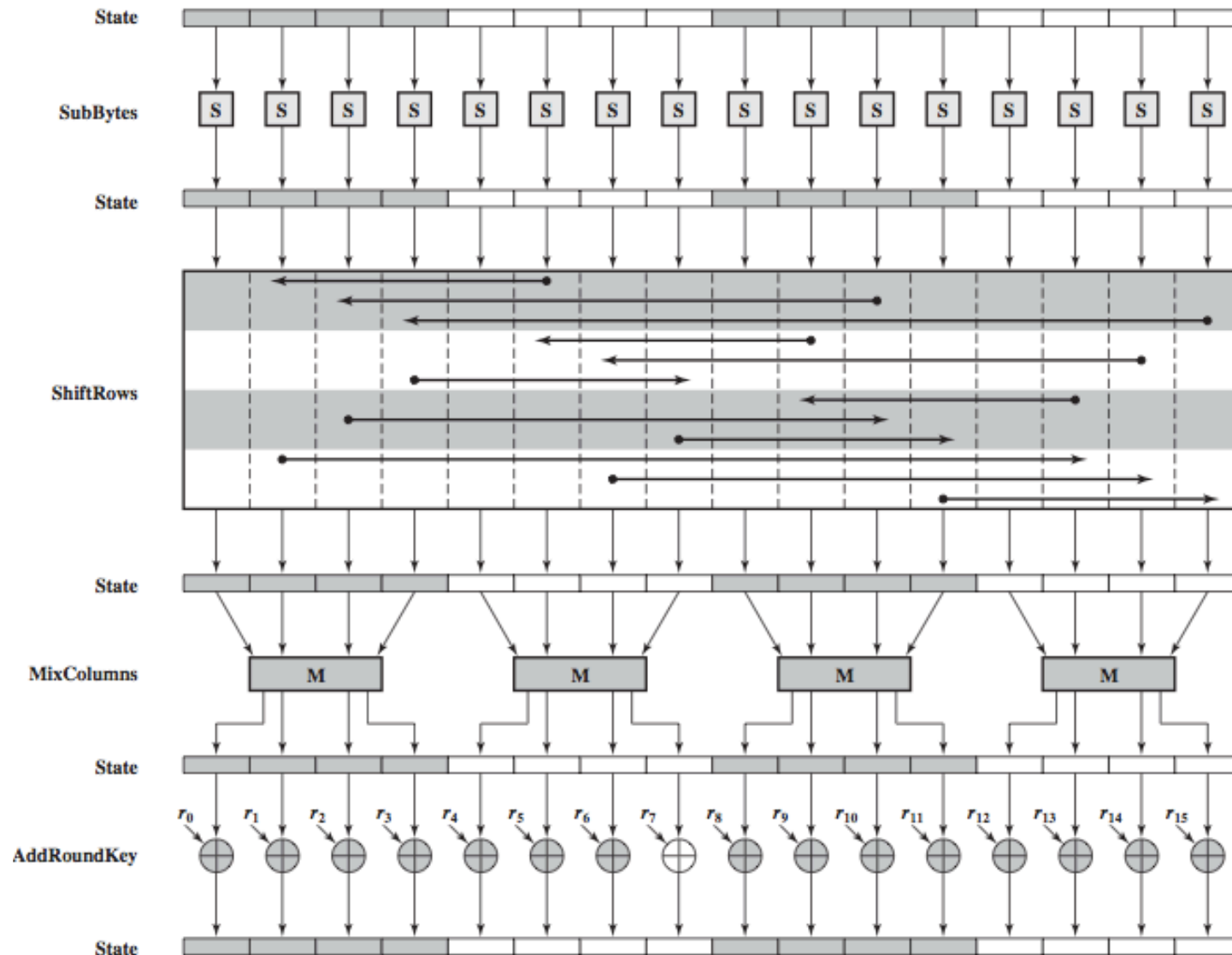
 $\oplus$ 

$w_i$	$w_{i+1}$	$w_{i+2}$	$w_{i+3}$
-------	-----------	-----------	-----------

 $=$ 

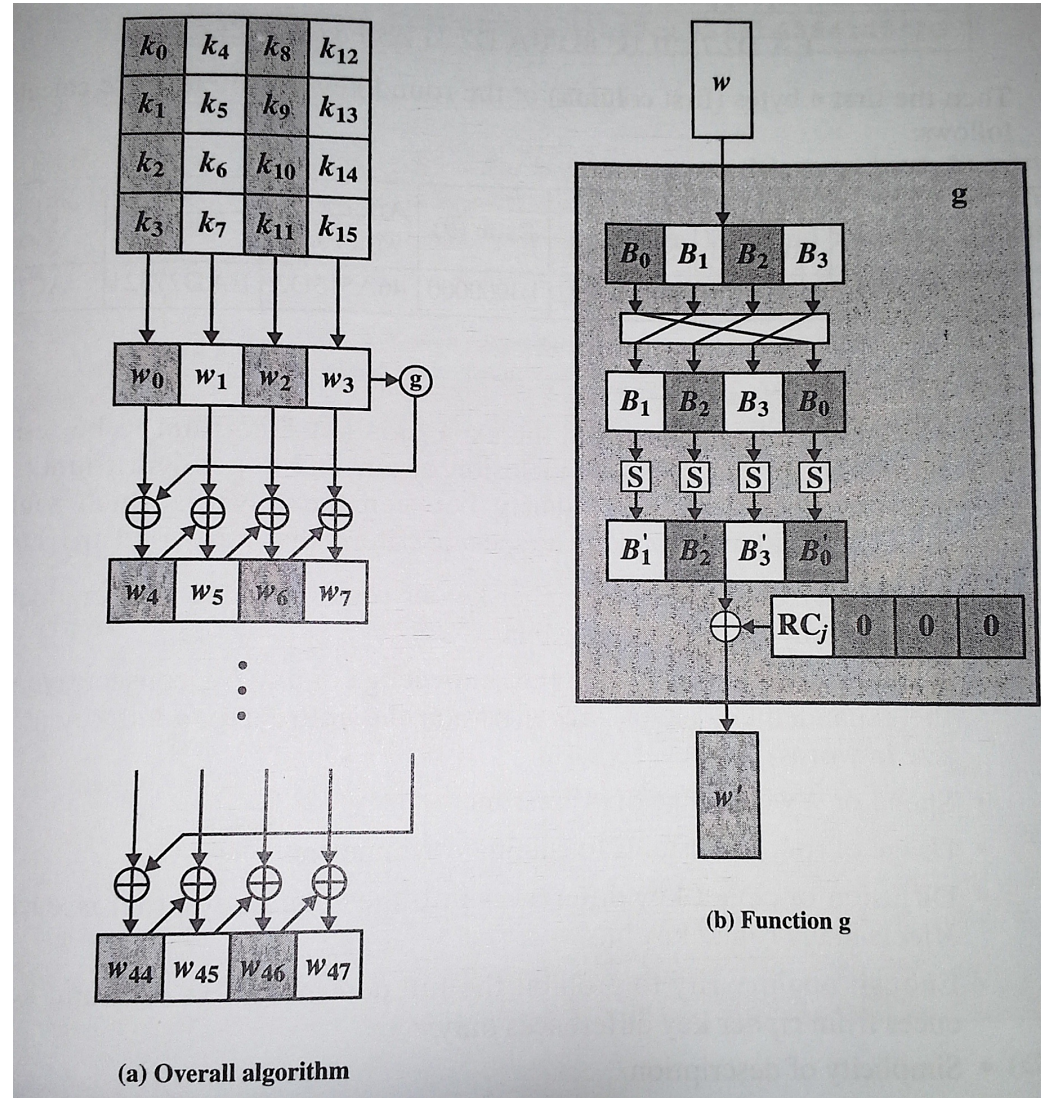
$s'_{0,0}$	$s'_{0,1}$	$s'_{0,2}$	$s'_{0,3}$
$s'_{1,0}$	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$
$s'_{2,0}$	$s'_{2,1}$	$s'_{2,2}$	$s'_{2,3}$
$s'_{3,0}$	$s'_{3,1}$	$s'_{3,2}$	$s'_{3,3}$

# TAHAP ENKRIPSI AES



# EKSPANSI KUNCI

- Dari 16 byte key diekspansi menjadi 44 word (176 byte).
- Masing-masing tahap memerlukan 4 word.
- Kunci dikopi ke 4 word pertama.
- Sisa words diisi 4 word sekaligus dalam sekali waktu.
- Setiap word (kecuali word pada kelipatan 4) bergantung pada word sebelumnya dan 4 word sebelumnya (dilakukan operasi XOR).



- **Untuk word dengan posisi kelipatan 4, digunakan fungsi g sbb:**
  - Rotword: 1 byte circular left shift pada word.
  - Subword: substitusi setiap byte pada input word dengan menggunakan S-box.
  - Hasil dari 2 operasi di atas dioperasikan XOR dengan suatu konstanta  $Rcon[j]$ .
    - Satu word dengan 3 byte paling kanan selalu 0 → melakukan XOR pada byte terkiri dari word tsb.
    - Konstanta setiap round:  
 $Rcon[j] = (RC[j], 0, 0, 0)$  dengan  $RC[1] = 1$ ,  $RC[j] = 2^{RC[j-1]}$ 
      - Perkalian didefinisikan pada field  $GF(2^8)$

# AES PARAMETERS

Key size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext block size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of rounds	10	12	14
Round key size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded key size (words/bytes/bits)	44/176	52/208	60/240

# ASPEK-ASPEK IMPLEMENTASI

- **Untuk 8 bit prosesor:**
  - Dapat diimplementasikan dengan efisien → smart cards dengan 8 bit
- **Untuk 32 bit prosesor:**
  - Implementasi akan lebih efisien jika operasi didefinisikan pada 32-bit words.

# LATIHAN

1. **Buatlah tabel penjumlahan dan perkalian untuk  $GF(2^4)$  dengan  $m(x) = x^4 + x + 1$ .**
2. **Bagaimana difusi dilakukan pada AES?**
3. **Misalkan terdapat bit string 00000000 dan 01011100 menjadi input dari AES-S-box. Apakah output-nya?**
4. **Misalkan plaintext pada AES adalah sbb:**
  - a. Bagaimana nilai state array setelah transformasi substitute byte?
  - b. Bagaimana nilai state array setelah transformasi shift rows?
  - c. Bagaimana nilai state array setelah transformasi mix column?

State			
00	12	0C	08
04	04	00	23
12	12	13	19
14	00	11	19