ЗАШТИТА ПОДАТАКА

Симетрични алгоритми заштите

AES

Преглед

- биће објашњено:
  - како је настао AES
  - AES алгоритам
  - детаљи по корацима
  - проширивање кључа
  - имплементациони аспекти

Advanced Encryption Standard

- Кључне карактеристике:
  - AES је блок алгоритам намењен да замени DES у комерцијалним апликацијама. Користи 128-бита за величину блока и 128, 192 или 256 бита за величину кључа.
  - AES не користи Feistel структуру. Уместо тога, свака итерација се састоји од четири засебне функције:
    - byte substitution,
    - permutation,
    - arithmetic operations over a finite field,
    - XOR with a key.

Порекло AES

- постоји потреба за заменом DES
  - теоретски постоје аналитички напади који га могу разбити
  - напади испробавањем свих могућности кључа (brute force) постају све опаснији
- може се користити Triple-DES
  - са већим кључем, а истом основном структуром алгоритма, показано је да не постоји криптоанализа која га је могла оборити осим brute force-а
  - али је спор (три пута DES) и блокови остају мали (64-бита)

Порекло AES(2)

- US NIST (National Institute of Standards and Technology) издаје позив за предлоге за нови AES 1997. године
- AES би требало да има сигурност, једнаку или бољу од Triple-DES и знатно бољу ефикасност
- поред тога, NIST специфицира да би AES морао да буде симетрични блок алгоритам са дужином блока од 128 бита и подршком за дужине кључа од 128, 192 и 256 бита

Порекло AES(3)

- 15 кандидата је прихваћено у првом кругу избора 1998. године
- 5 кандидата је ушло у ужи избор након другог круга избора 1999. године
- изабран је Rijndael алгоритам (Dr. Joan Daemen и Dr. Vincent Rijmen) као финални AES 2000. године
- NIST је издао нови стандард (FIPS PUB 197) у новембру 2001. године

Захтеви AES

- симетрични блок алгоритам заштите (private key symmetric block cipher)
- 128-бита подаци, 128/192/256-бита кључеви
- сигурнији & бржи од Triple-DES
- животни век од 20-30 година

AES критеријуми избора

- првобитни критеријуми(први и други круг):
  - сигурност (security) – напор да би се неком криптоанализом разбила шифра (brute force напади су искључени као могућност, јер је најмања величина кључа 128-бита)
  - цена (cost) – ефикасност при израчунавању (јер је намењен за употребу у брзим апликацијама и за пренос по брзим мрежама)
  - карактеристике алгоритма и имплементације (algorithm & implementation characteristics):
    - флексибилност,
    - могућност различитих софтверских и хардверских имплементација,
    - једноставност дизајна

AES критеријуми избора(2)

- коначни критеријуми:
  1. општа сигурност (general security) - алгоритми тестирани од стране криптографске заједнице у року од три године (диференцијална и линеарна криптоанализа)
  2. софтверска имплементација (software implementation) - брзина, брзина у односу на платформу и брзина у односу на величину кључа
  3. окружења са ограниченим простором (restricted-space environments) - меморијске картице и сл.
  4. хардверска имплементација (hardware implementations) - могућност оптимизације перформанси

AES критеријуми избора(3)

- коначни критеријуми:
  5. напади на имплементацију (attacks on implementations) – дужина трајања операција,...
  6. шифровање наспрам дешифровања (encryption versus decryption) - колика је сличност
  7. променљивост кључа (key agility) - могућност замене кључа брзо и уз минималне ресурсе
  8. флексибилност (flexibility) - могућност унапређивања алгоритма (нове величине блокова и/или кључева, ...)
  9. потенцијал за ILP (potential for instruction-level parallelism) - могућност максималног искоришћења процесора

AES оцена Rijndael алгоритма

1. општа сигурност – нема познатих напада
2. софтверска имплементација – добре перформансе на различитим платформама
3. окружења са ограниченим простором – погодан, мала потрошња меморије
4. хардверска имплементација – најбољи од свих финалиста
5. напади на имплементацију – најлакше се брани од оваквих напада
6. шифровање наспрам дешифровања – разликују се
7. променљивост кључа – захтева извршавање проширивања кључа
8. флексибилност – може се прилагодити величина блока/кључа на било коју која је умножак 32
9. потенцијал за ILP – одличан потенцијал за паралелизам

AES алгоритам - параметри

- дизајниран од стране тима Rijmen-Daemen из Белгије
- има 128/192/256 битне кључеве, 128 битне блокове

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)	44/176	52/208	60/240

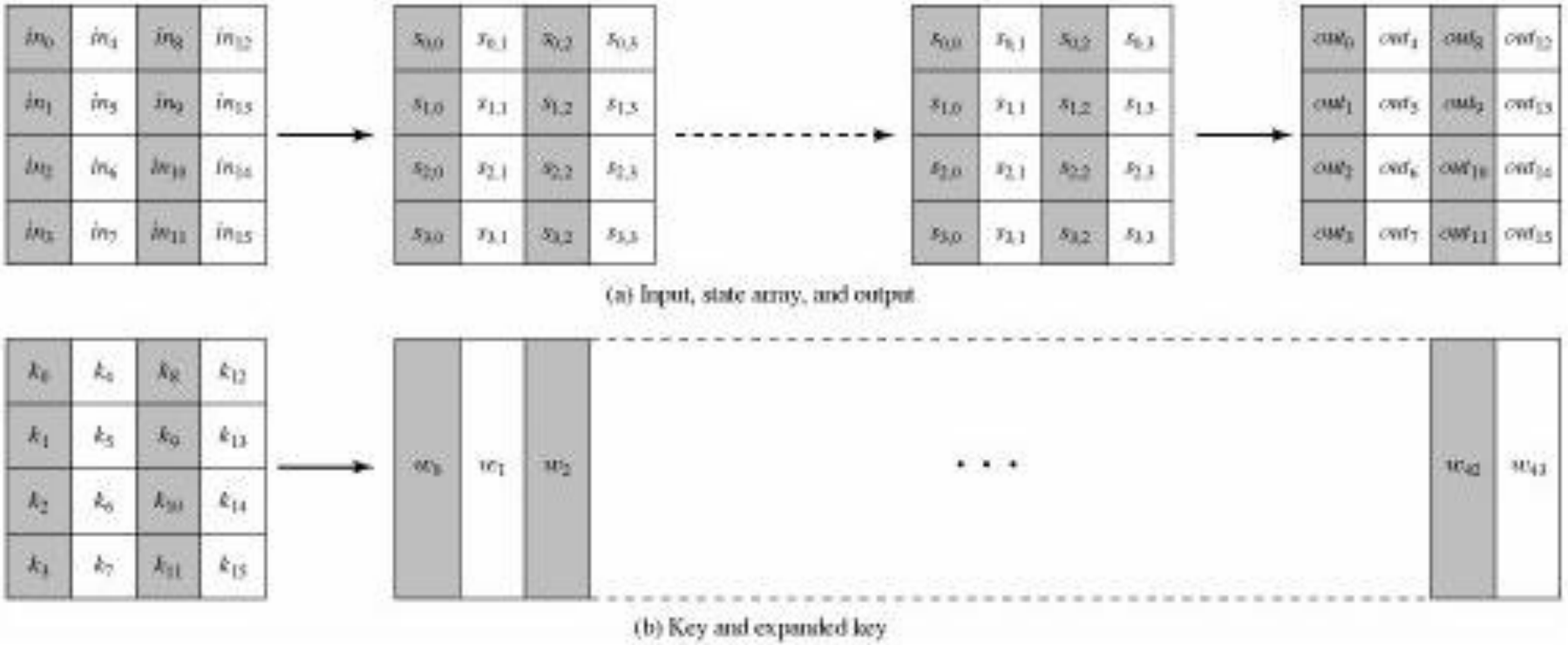
AES алгоритам - дизајн

- **итеративан** алгоритам не користи **feistel** структуру
  - дели податке у 4 групе од 4 бајта
  - у свакој итерацији извршава операције над целим блоком података
- дизајниран је тако да буде:
  - отпоран на све познате нападе
  - брз и компактан на широком спектру платформи
  - једноставног дизајна

AES алгоритам - структуре података

- обрађује податке као 4 групе од по 4 бајта (стање (**state**))
- 128-битни кључ се представља као матрица 4 колоне по 4 бајта. Кључ се затим проширује у низ речи од којих свака има по 4 бајта и укупно има 44 речи за кључ од 128 бита.
- Кључ је проширен на 44 32-битне речи, w[i]. Четири различите речи (128 бита) користе се као итеративни кључ у свакој итерацији

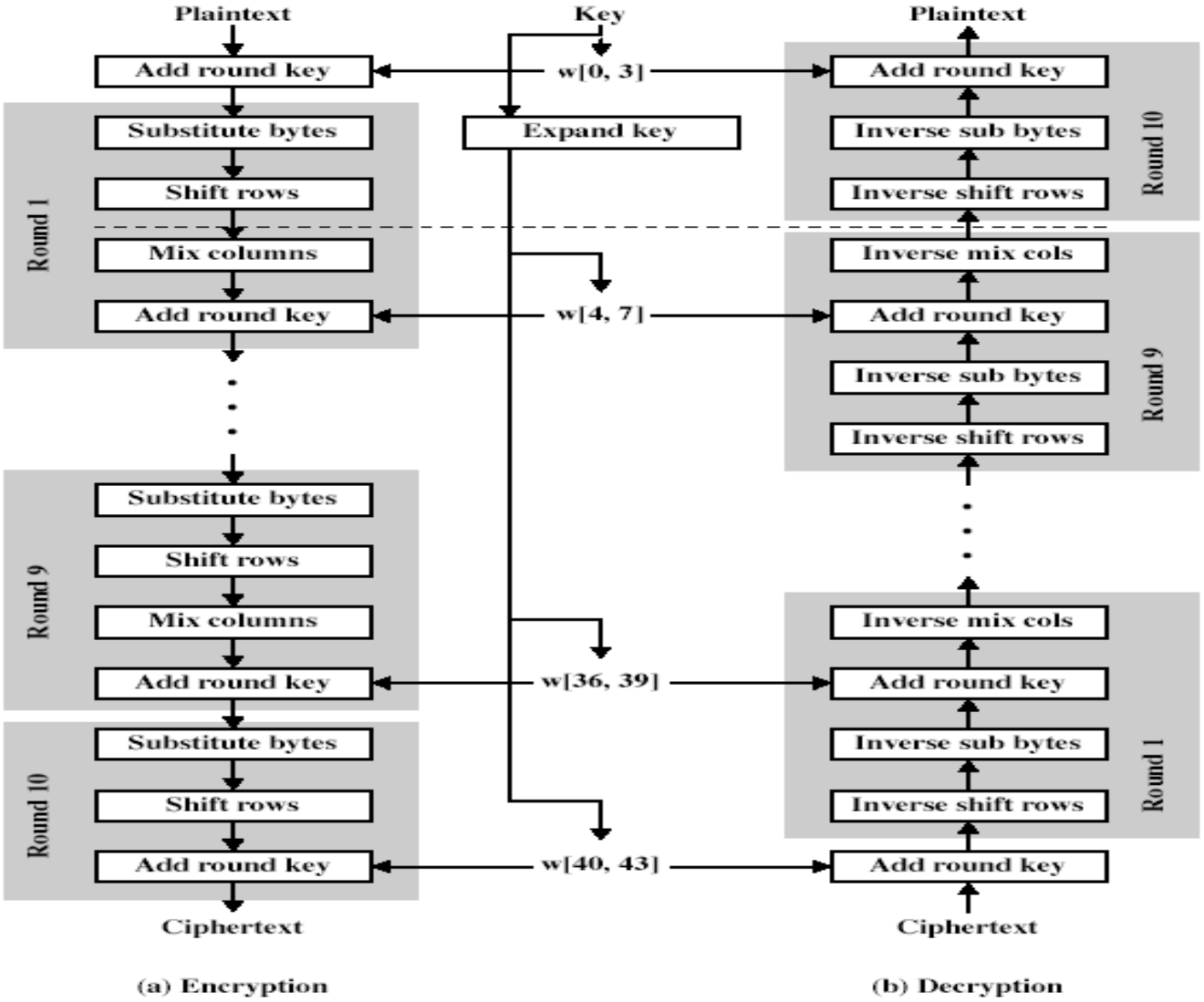
AES алгоритам - структуре података(2)



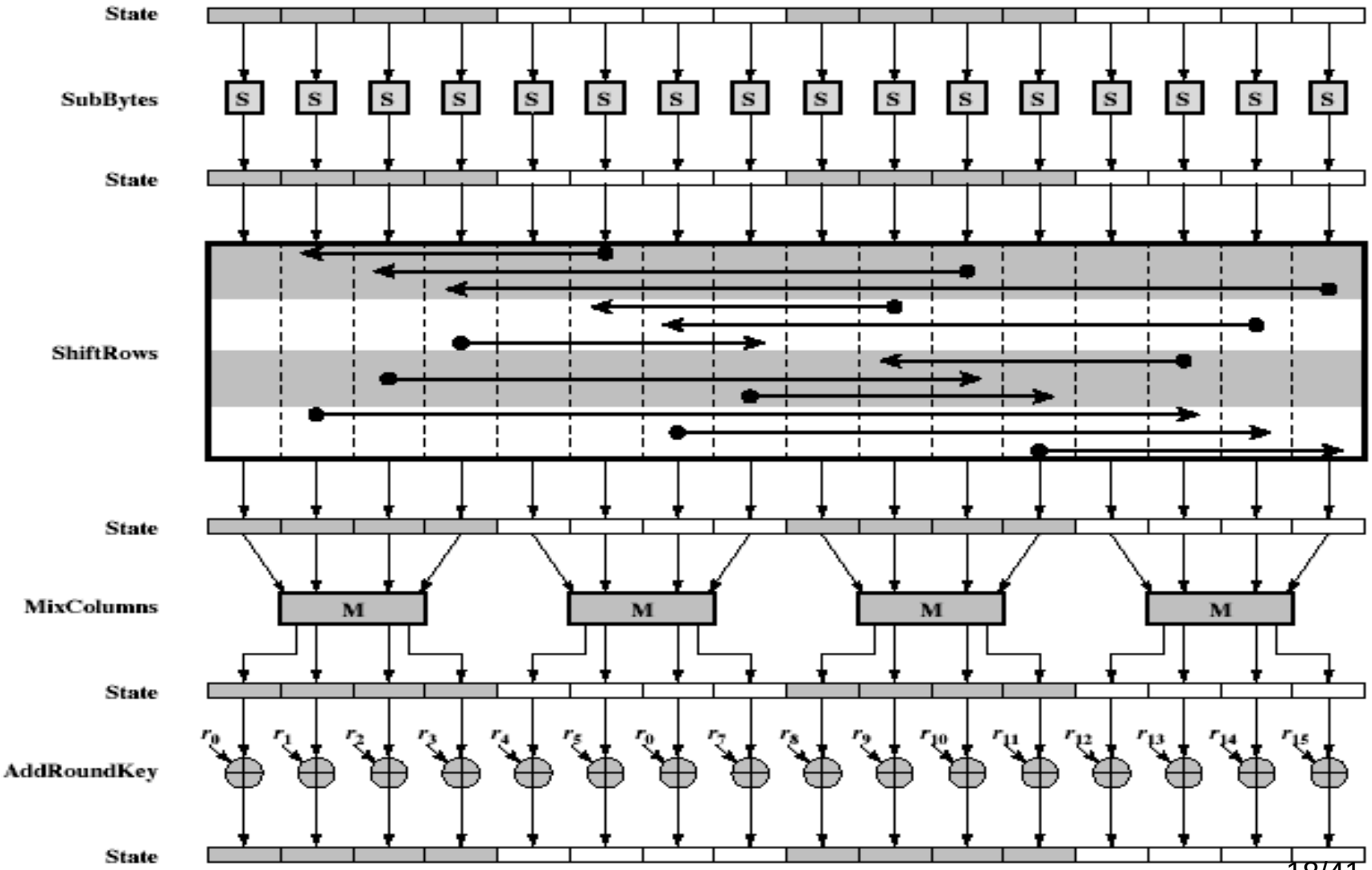
AES алгоритам - кораци

- има 9/11/13 итерација у којима стање(**state**) пролази кроз:
  - byte substitution (1 S-box се користи за сваки бајт)
  - shift rows (пермутују се бајтови између група/колона)
  - mix columns (замена која користи аритметику над GF(2^8))
  - add round key (XOR **state** with portion of the expanded key)
- почетно XOR са делом проширеног кључа и некомплетна последња итерација

AES алгоритам - кораци(2)



AES итерација



AES алгоритам - карактеристике

- карактеристике:
  - не користи Feistel структуру
  - почиње и завршава се са add round key, јер једино ова фаза користи кључ
  - остале три фазе обезбеђују конфузију, дифузију и нелинеарност, али самостално не доприносе безбедности, јер не користе кључ
  - свака фаза је инвертибилна. За add round key фазу, инвертибилност се постиже XOR-ом истог кључа итерације и блока користећи да је A XOR A XOR B = B. За остале фазе користе се инверзне функције при дешифровању.

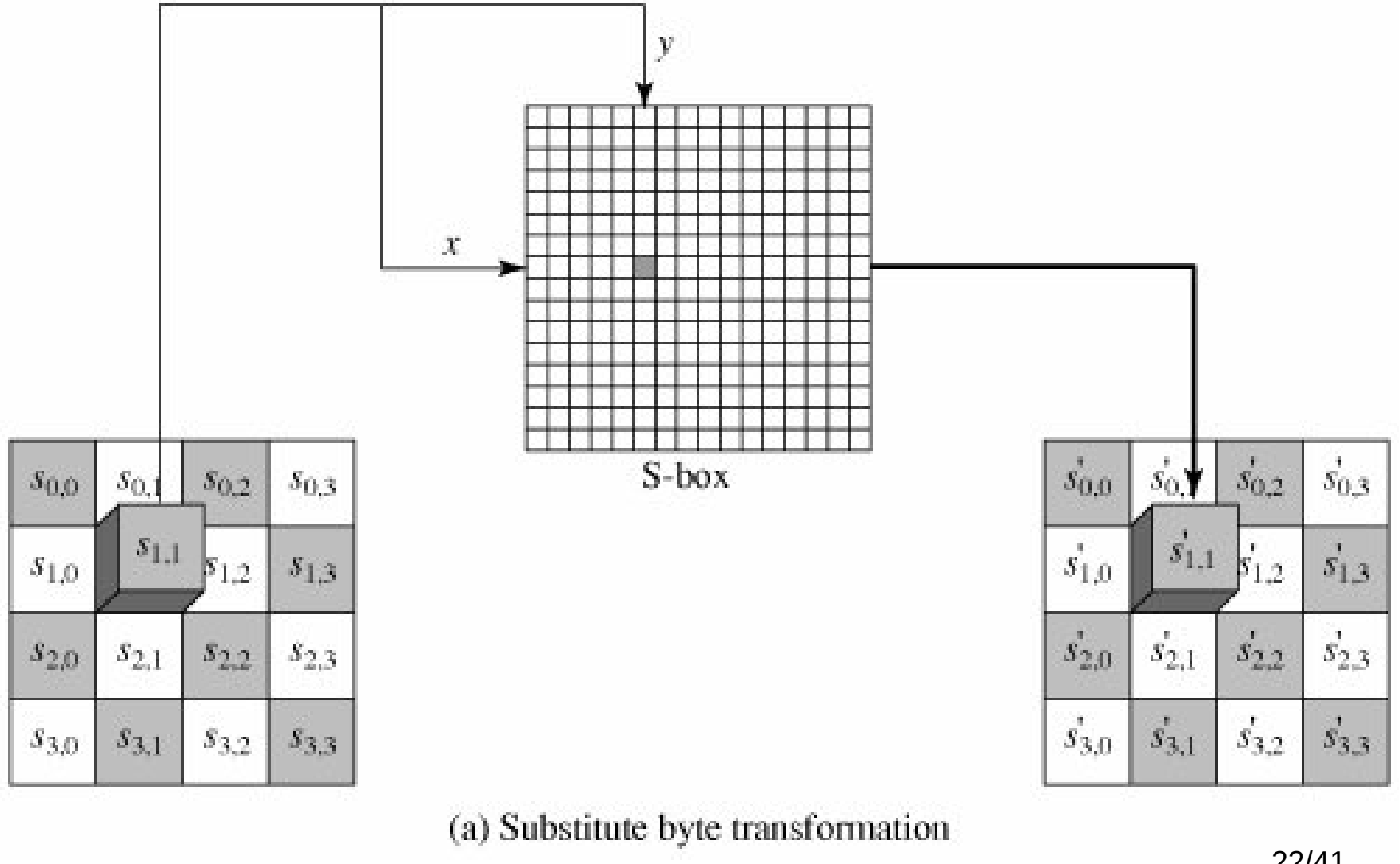
AES алгоритам - карактеристике(2)

- карактеристике:
  - алгоритам није идентичан код шифровања и дешифровања (мада се користи проширени кључ у инверзном редоследу)
  - финална итерација и код шифровања и код дешифровања се састоји од само три фазе

Byte Substitution

- једноставна замена сваког бајта
- користи се једна табела величине 16x16 бајтова (S-box) која садржи пермутацију свих могућих 256 8-битних вредности
- сваки бајт стања (**state**) се мења са бајтом из табеле из реда, који је одређен са лева 4 бита, и колоне, која је одређена са десна 4 бита
- S-box је конструисана коришћењем дефинисане трансформације вредности у GF(2^8)
- S-box је дизајнирана да буде отпорна на све познате нападе

Byte Substitution(2)



(a) Substitute byte transformation

Byte Substitution(3)

(a) S-box

x \ y	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

Byte Substitution(4)

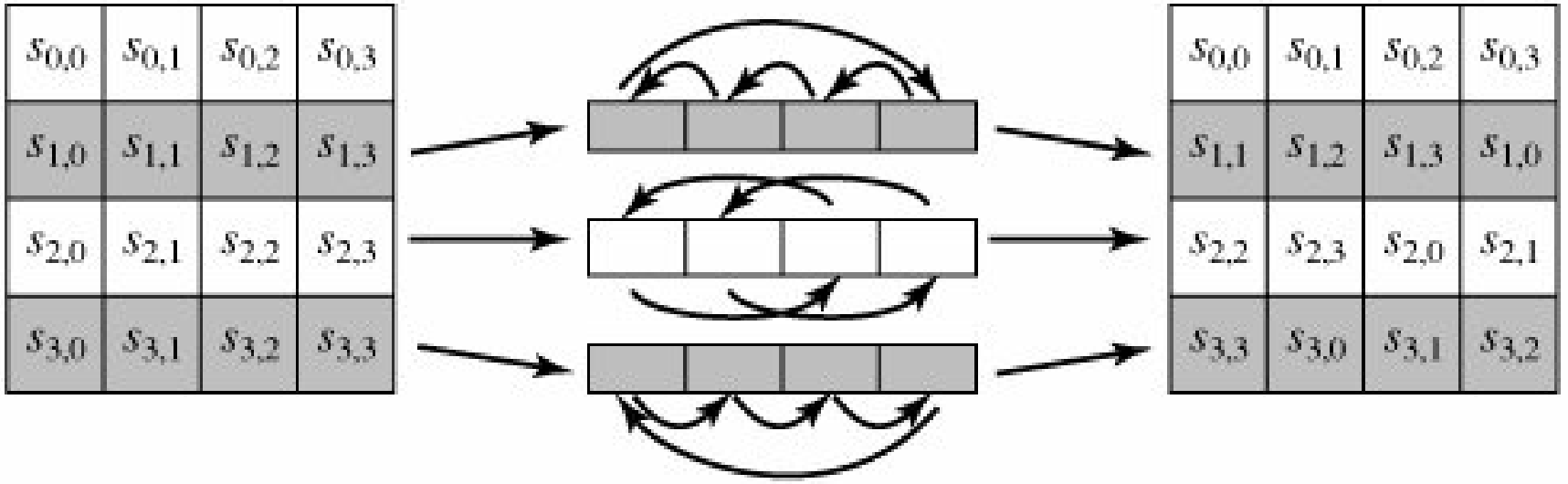
(b) Inverse S-box

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

Shift Rows

- за сваки ред се ради кружно померање у лево
  - 1. ред је непромењен
  - 2. ред ради се кружно померање у лево за 1 бајт
  - 3. ред ради се кружно померање у лево за 2 бајта
  - 4. ред ради се кружно померање у лево за 3 бајта
- дешифровање ради померање у десно
- како се стање (state) обрађује по колонама (групама), овај корак пермутује бајтове из једне колоне у 4 различите колоне

Shift Rows(2)



(a) Shift row transformation

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

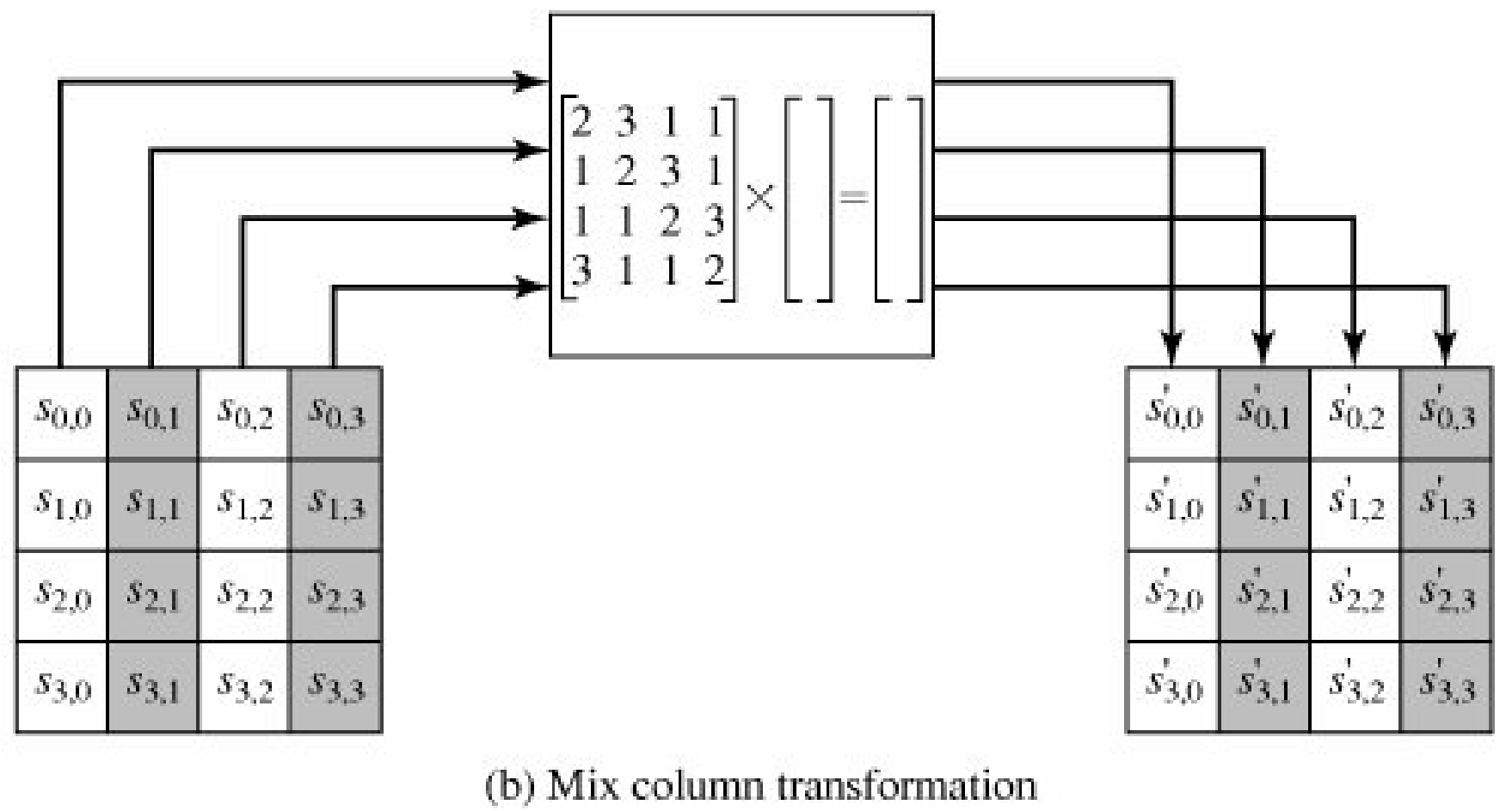
→

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

Mix Columns

- свака колона се обрађује засебно
- сваки бајт се замењује вредношћу која зависи од сва 4 бајта колоне
- заправо, множење матрица у $GF(2^8)$ коришћењем несводљивог полинома $m(x) = x^8+x^4+x^3+x+1$

Mix Columns(2)



(b) Mix column transformation

Mix Columns(3)

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

MixColumns

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

InversMixColumns

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

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Mix Columns(4)

- 87*02+6E*03+46*01+A6*01 = 47
- у $GF(2^8)$ сабирање је исто што и XOR
- множење са 2 је исто што и 1-bit shift left уз условно XOR са 0001 1011 ($x^8+x^4+x^3+x+1$) уколико је најзначајнији бит оригиналне вредности 1
- 87*02 = 1000 0111 – 0000 1110 XOR 0001 1011 = 0001 0101
- 6E*03 = 6E*02+6E
  - 6E*02 = 0110 1110 – 1101 1100
  - +6E = 1101 1100 XOR 0110 1110 = 1011 0010
- 46*01 = 0100 0110
- A6*01 = 1010 0110
- 0001 0101 XOR 1011 0010 XOR 0100 0110 XOR 1010 0110 = 0100 0111 = 47

Mix Columns(5)

- Коефицијенти матрице су базирани на линеарном коду са максималним растојањем између кодних речи, што обезбеђује добро мешање бајтова сваке колоне
- У комбинацији са Shift Rows, обезбеђује да након неколико итерација сви битови излаза зависе од свих битова улаза
- Избор коефицијената (01, 02, 03) заснован је на разматрањима која се односе на имплементацију. Множење са овим коефицијентима укључује највише операцију померања (shift) и XOR

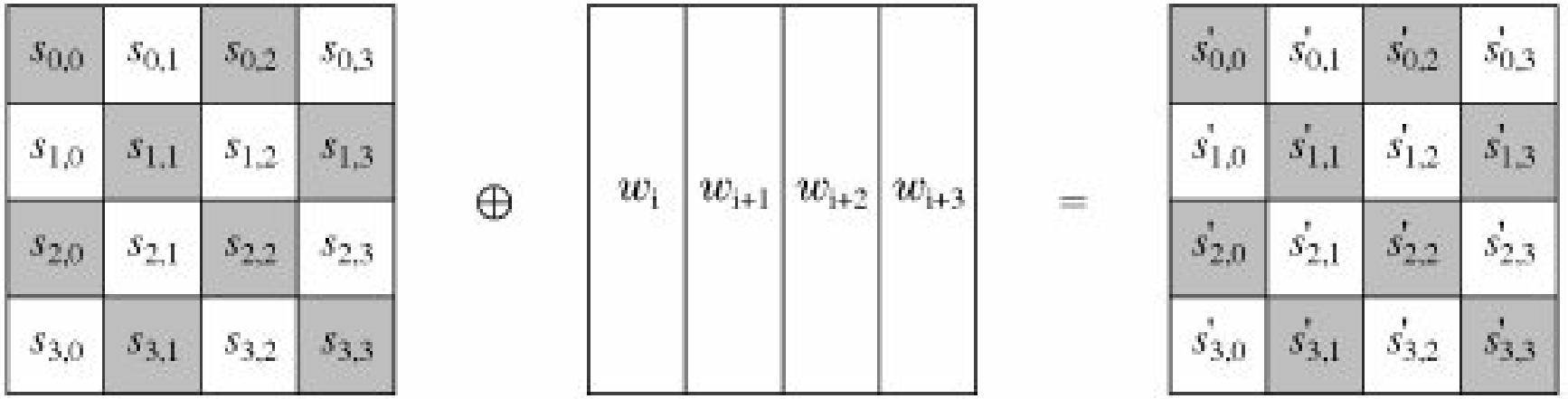
Mix Columns(6)

- Коефицијенти код инверзне операције Mix Columns су много робуснији и тежи за имплементацију, али је овакав дизајн усвојен из два разлога, која показују да је шифровање важније од дешифровања:
  - За CFB и OFB модове користи се само шифровање
  - када се користи за прављење кода за аутентикацију поруке (message authentication code)

Add Round Key

- XOR-уje се стање (state) са 128-бита кључа итерације
- обрада се врши по колонама (иако, технички представља низ бајтовских операција)
- инверзна операција код дешифровања је идентична, јер је XOR сам себи инверзан, само је потребан прави кључ итерације
- дизајнирано је да буде што је једноставније могуће
- комплексност key expansion и осталих фаза AES, обезбеђују сигурност

Add Round Key(2)



(b) Add Round Key Transformation

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

⊕

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

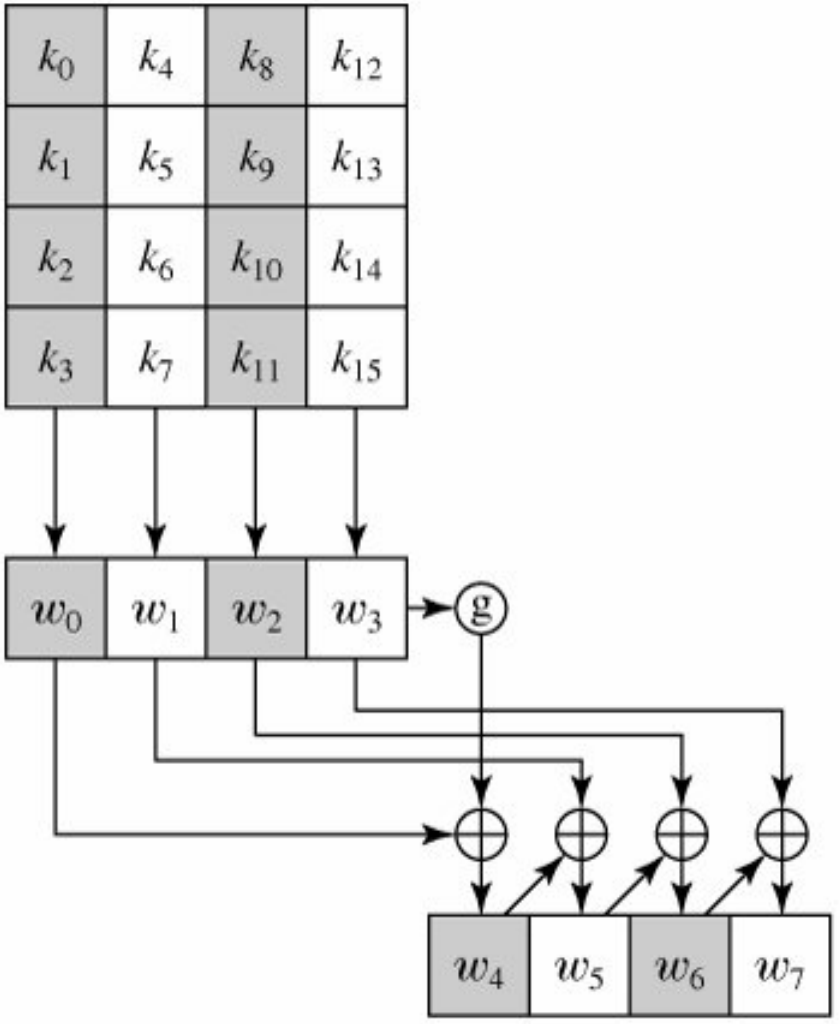
EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D2

The first matrix is **State**, and the second matrix is the round key.

AES Key Expansion

- узима 128/192/256-битни (16/24/32-бајтни) кључ и развија га у низ од 44/52/60 32-битних речи
- почиње се са копирањем кључа у прве четири речи
- затим се у петљи праве речи које зависе од претходне речи w[i-1], и речи која је за 4 места уназад w[i-4]
  - у 3 од 4 случаја, примењује се само XOR
  - сваки 4. има сложенију функцију (g)
- дизајниран да буде отпоран на све познате нападе

AES Key Expansion(2)



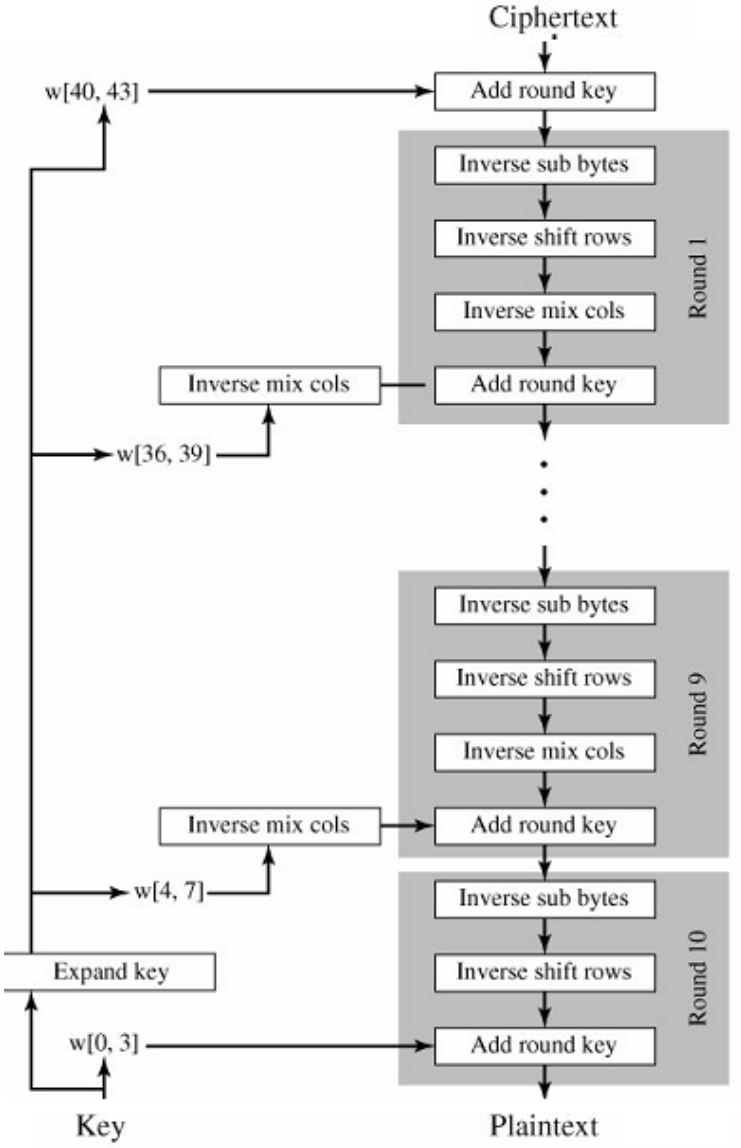
AES Key Expansion(3)

- Функција g састоји се од следећих подфункција:
  - кружно померање речи у лево за 1 бајт. Значи ако је улазна реч [b0, b1, b2, b3] трансформише се у [b1, b2, b3, b0].
  - S-box врши замену за сваки бајт улазне речи
  - резултат након подфункција 1 и 2 се XOR-ују са константом итерације, Rcon[j].

AES Дешифровање

- AES дешифровање није идентично са шифровањем, јер се кораци раде у инверзном редоследу
- међутим може се дефинисати еквивалентна инверзна шифра са корацима као код шифровања
  - али уз коришћење инверзних функција сваког корака
  - са различитим распоредом кључа
- ради, јер се резултат не мења када
  - заменимо byte substitution & shift rows
  - заменимо mix columns & add round key

AES Дешифровање - Equivalent Inverse Cipher



Имплементациони аспекти

- може се ефикасно имплементирати на 8-битном процесору
  - byte substitution ради са бајтовима користећи табелу са 256 улаза
  - shift rows је једноставно померање бајтова
  - add round key ради бајтовско XOR
  - mix columns захтева множење матрица у $GF(2^8)$ које ради на нивоу бајта (померање и XOR), а може се упростити да ради претрагу по табели

Имплементациони аспекти

- може се ефикасно имплементирати на 32-битном процесору
  - предефинишу се фазе тако да користе речи дужине 32-бита
  - може да прерачуна четири табеле од 256-речи
  - тада се свака колона у свакој итерацији може израчунати коришћењем 4 погледа у табелу + 4 XOR-а
  - по цени од 16Kb да се сачувају табеле
- дизајнери верују да је оваква јако ефикасна имплементација била одлучујући фактор приликом избора за AES алгоритам