ADVANCED ENCRYPTION STANDARD

(AES)

Rijndael-Inspector-v1.1

Istoria algoritmilor simetrici

1. Mai 1973, Biroul National Federal din SUA lansează un apel ce priveşte construirea unui sistem de criptare oficial care să se numească Data Encryption Standard;
2. Martie 1975, firma IBM construieşte DES, modificând un sistem de criptare mai vechi numit Lucifer;
3. Ianuarie 1977 DES a fost adoptat oficial ca standard de criptare, el fiind evaluat o data la 5 ani;
4. Iulie 1998 DES pe 56 biţi este spart de către organizaţia Electronic Frontier Foundation; Aceştia au construit un calculator dedicat DESCHALL care poate decripta un mesaj care încearca toate cheile posibile in mai putin de 3 zile.

Atac DES:

Cheia: 56 biti – de incercat 72,057,594,037,927,936 chei posibile.
Cost calculator: < $250,000
Cauta 88 miliarde chei/sec

In anul 1990 apare un cifru simetric foarte asemănător cu DES dezvoltat de Xuejia Lai şi James Massey de la Institutul Federal al Tehnologiei din Elveţia numit PES (Proposed Encryption Standard). In acelaşi an Biham şi Shamir publică o metodă de atac prin criptanaliză diferenţială asupra DES-ului, de aceea un an mai târziu apare IPES (Improved Proposed Encryption Standard) o versiune modificată a PES astfel încât să reziste criptanalizei diferenţiale. In 1992 IPES este redenumit devenind IDEA (International Data Encryption Standard). IDEA este inclus în PGP (Pretty Good Privacy), practic este o componentă a sistemului de securitate prin poşta electronică care a contribuit la răspândirea acestuia.

La sfârşitul anilor 90 se decide inlocuirea sistemului de criptare DES, de aceea NIST (National Institute of Standards and Tehnology) anunţă un set de 15 algoritmi propuşi să inlocuiască DES.

Criteriile stabilite de NIST pentru noul sistem au fost:

1. Să fie un sistem de criptare simetric pe blocuri de 128 biţi.
2. Să accepte chei de lungime 128, 192 si 256 biti;
3. Să nu aiba chei slabe;
4. Să fie eficient atât pe platforme Intel Pentium Pro cât si pe alte platforme software sau hardware;

Martie 1999, în finala concursului propus de NIST ajung 5 algoritmi:
MARS – propus de IBM

1. RC6 – versiune a lui RC5 elaborată de laboratoarele RSA
2. SERPENT – propus de Ross Ross Anderson (Universitatea Cambridge),
3. Eli Biham (Institutul Tehnion, Haifa) ,si Lars Knudsen (Universitea Bergen).

4. TWOFISH – propus de un colectiv condus de Bruce Schenier.
5. RIJNDAEL propus de către Joan Daemen şi Vincent Rijmen (Belgia).

Mai 2000 NIST anunţă drept sistem câştigător sistemul de criptare Rijndael,care devine oficial în noiembrie 2001 standard FIPS-197.

Descrierea algorimului AES (Advanced Data Encryption)

Proiectarea algoritmului AES

Este un cifru bloc simetric proiectat pe principiul substitution-permutation network (SPN); (cifrurile bloc pot fi proiectate cu ajutorul a trei mari principii:

1. Substitution-permutation network SPN , adică fiecare rundă lucrează cu întregul bloc de date;
2. Feistel network , fiecare rundă operează pe o submulţime a blocului de date; funcţia de rundă se aplică doar unei submulţimi într-o rundă;
3. Un alt tip de cifru bloc este Lay-Massey scheme – ex: IDEA)

Rapid atât software cât şi hardware;

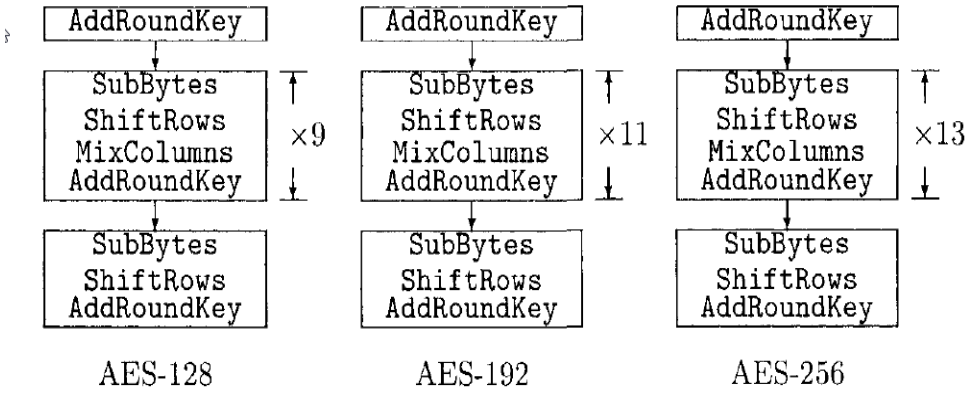
Dimensiunea blocului de date este de 128 biţi, iar a cheii poate fi 128, 192, 256 biţi;

Diferenţa dintre Rijndael şi AES o constituie intervalul în care ia valori dimensiunea blocului de date şi a cheii. În cazul Rijndael lungimea mesajului clar cât şi a cheii pot fi multiplu de 32, nu mai mică de 128 biţi şi maxim de 256 biţi, iar pentru AES lungimea blocului de text clar este fixă, adică 128 biţi iar lungimea cheii poate fi 128, 192 sau 256 biţi.

AES operează pe o matrice de octeţi de dimensiune 4x4; (dacă blocul de date este mai mare numărul de coloane va creşte);

Cele mai multe calcule sunt făcute într-un câmp finit special, numit câmp Galois şi notat GF(2^8);

Cifrarea blocului se face după un anumit număr de runde; acest număr depinde de dimensiunea cheii; fiecare rundă conţine 4 transformări mai puţin ultima;



Elemente teoretice folosite in implementarea algoritmului AES

Operaţii într-un câmp Galois

Câmpul Galois pentru AES este o construcţie matematică specială unde adunarea, scăderea, multiplicarea şi împarţirea sunt redefinite şi numărul de întregi din corp este finit.

Mai detaliat, câmpul Galois lucrează pe numere de opt biţi (numere de la 0 la 255). Toate operaţiile matematice definite pe acest corp au ca rezultat un număr pe opt biţi (un octet). AES lucrează la nivel de octet, o secvenţă de opt biţi tratata ca o singură entitate, de exemplu:

$$\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}.$$

Aceşti octeţi sunt reprezentaţi ca elemente ale unui câmp finit cu ajutorul unei reprezentări polinomiale:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + {}^{b}{}_1x^1 + b_0x^0 = \sum_{i=0}^{7} b_i x^i.$$

De exemplu: octetul {01100011} poate fi elementul $x^6 + x^5 + x + 1$.

De asemenea putem scrie un octet ca două caractere hexazecimale (un caracter hexazecimal fiind scris ca un grup de patru biţi).

De exemplu: {01100011} poate fi scris ca {63} .

Adunarea într-un câmp finit:

polinomial: $(x^6 + x^4 + x + 1) + (x^7 + x^6 + x^3 + x) = x^7 + x^4 + x^3 + 1$
binar: {01010011} ⊕ {11001010} = {10011001}
Hexa: {53} + {CA} = {99}

Inmultirea:

In reprezentarea polinomială, înmulţirea în $GF(2^8)$,(notată cu •) este acelaşi lucru cu înmulţirea polinoamelor modulo un polinom ireductibil de grad 8. Un polinom este ireductibil dacă şi numai dacă divizorii săi sunt 1 şi el însuşi. Pentru algoritmul AES acest polinom ireductibil este:

$m(x) = x^8 + x^4 + x^3 + x + 1$ sau in notatie hexazecimala {01}{1b}.

De exemplu:
{57}•{83}=
$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$
$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$ modulo $(x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + 1$

Operaţia modulo m(x) ne asigura ca rezultatul va fi un polinom binar cu gradul mai mic ca 8, şi acesta poate fi reprezentat pe un octet. Spre deosebire de adunare , înmulţirea nu este o operaţie simplă la nivel de octet.

Înmulţirea cum este definită mai sus este asociativă si elementul {01} este element neutru (identitate). Pentru orice polinom binar b(x) diferit de 0 si cu gradul mai mic ca 8, inversul multiplicativ, notat $b^{-1}(x)$, poate fi găsit astfel: cu algoritmul lui Euclid extins se calculează polinoamele a(x) si c(x) astfel:

b(x)a(x) + m(x)c(x) = c.m.m.d.c.(b(x),m(x)) = 1 (când m(x) este ireductibil).

Atunci a(x) • b(x) mod m(x) = 1 ceea ce înseamnă $b^{-1}(x) = a(x) \bmod m(x)$.

Mai mult, pentru orice a(x), b(x) si c(x) din câmpul finit, rezultă că:

a(x) • (b(x) + c(x)) = a(x) • b(x) + a(x) • c(x).

Rezultă ca mulţimea celor 256 de octeţi posibili, cu adunarea (operatia XOR) şi înmulţirea definite mai sus au structură de câmp finit $GF(2^8)$.

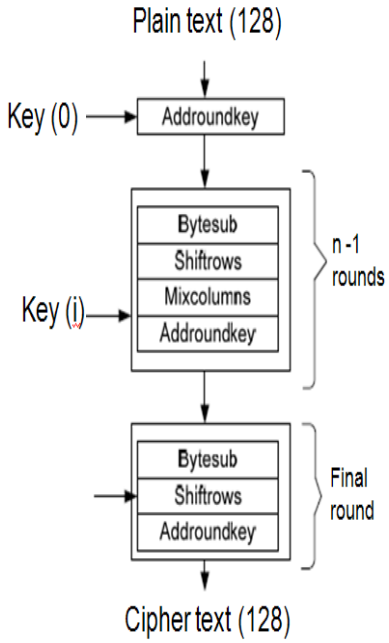
Înmultirea cu x:

Înmulţind polinomul binar $b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x^0$ cu x rezultă: $b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x$.

Rezultatul $x \bullet b(x)$ se obţine reducând rezultatul de mai sus modulo m(x). Dacă $b_7 = 0$, rezultatul este în formă deja redusă. Dacă $b_7 = 1$, se scade (XOR-ing) polinomul m(x). Apoi înmulţirea cu x (adică cu {00000010} sau {02}) poate fi implementată la nivel de octet ca deplasare la stanga (left shift) şi apoi XOR cu {1b}

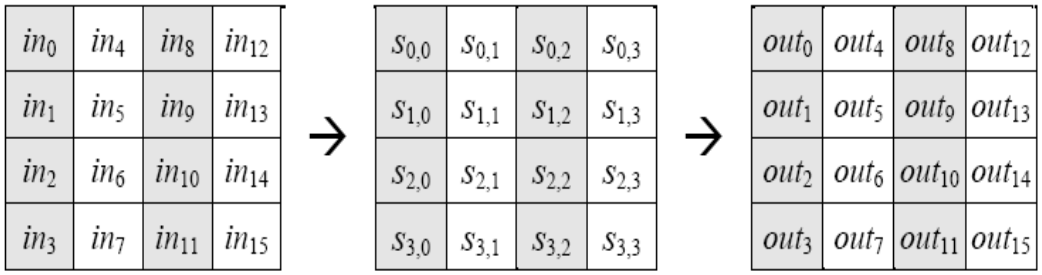
Etapele procesului de criptare:

1. Expandarea cheii ce are ca rezultat obţinerea cheilor de rundă; (se face cu ajutorul unui algoritm din cheia principală)
2. Runda iniţiala în care se execută funcţia:
   AddRoundKey
3. Runde intermediare ce conţin 4 transformări fiecare:
   SubBytes: transformare neliniară;
   ShiftRows: o transpoziţie a liniilor:
   MixColumns: o mixare de operaţii pe coloana;
   AddRoundKey;
4. Ultima rundă: conţine transformările:
   SubBytes;
   ShiftRows;
   AddRoundKey.



Starea intermediară: această stare intermediară este reprezentată de un tablou (matrice) a cărui elemente sunt reprezentate de octeţi. Această matrice are 4 linii şi Nb coloane; Nb este egal

cu lungimea blocului de date împărţit la 32 de biţi. In matricea stării intermediare notate cu s, fiecare octet are doi indici, r numărul liniei în intervalul $0 \leq r < 4$ si c numărul coloanei în intervalul $0 \leq c < Nb$. Deci elementele matricei de stare vor fi notate cu $s_{r,c}$ sau s[r,c].



octeţi de intrare starea intermediară octeţi de iesire

Atât la criptare cât şi la decriptare matricea de intrare trece în matricea de stare după formula: s[r,c] = in[r + 4c] pentru $0 \leq c < Nb$; şi la sfârşitul criptării sau decriptării matricea de stare trece în matricea de ieşire astfel: out[r + 4c] = s[r,c] pentru $0 \leq r < 4$ si $0 \leq c < Nb$.

Toţi octeţi din algoritmul AES sunt interpretaţi ca elemente ale unui câmp finit. Aceste elemente pot fi adunate, multiplicate dar aceste operaţii sunt diferite de cele obişnuite cu numere întregi.

Descrierea unei runde:

Transformarea SubBytes(stare) este o substituţie neliniară de octeţi care operează independent pe fiecare octet cu ajutorul unui S-box. Acest S-box este construit prin compunerea a două transformări:

1. Calcularea inversului multiplicativ pentru fiecare octet nenul în corpul finit $GF(2^8)$, elementul {0,0} rămânând neschimbat;
2. Rezultatul este modificat printr-o transformare afină peste *Z*2:Această transformare scrisă în forma matriceală arată astfel:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

de exemplu dacă $S_{1,1}$={53}-> S-box->$s'_{1,1}$={ed} elementul aflat la intersecţia liniei 5 cu coloana 3

S- boxul este o funcţie neliniară, bijectivă (singura parte neliniară a cifrului).

S- boxul din AES foloseşte următoarea transformare afină:

$y = Ax \oplus C \bmod m(x)$ unde: $m(x) = x^8 + x^4 + x^3 + x + 1$

$A = [f8, 7c, 3e, 1f, 8f, c7, e1, f1]^T$, matrice 8x8 în GF(2)

$C = [63]^T$, matrice coloană în GF(2).

Pentru a fi generatoare pentru S-box matricea A trebuie să fie nesingulară. Putem genera aproximativ 2^{63} astfel de matrici nesingulare cu fiecare dintre polinoamele ireductibile. Polinoamele rezultate în matricile nesingulare sunt [01, 02, 04, 08, 10, 20, 40, 80], marginea inferioară şi [fe, 7f, bf, df, ef, f7, fb, fd], margine superioară.

Pentru a satisface efectul de avalanşă înseamnă ca modificarea unui singur bit la intrare să implice modificarea a cel puţin 50% din biţii de ieşire. Pentru a satisface Strict Criterion Avalanche este echivalent a spune că dacă modificăm un bit la intrare se vor altera exact 50% din biţii de ieşire.

Criterii în proiectarea S-boxului:

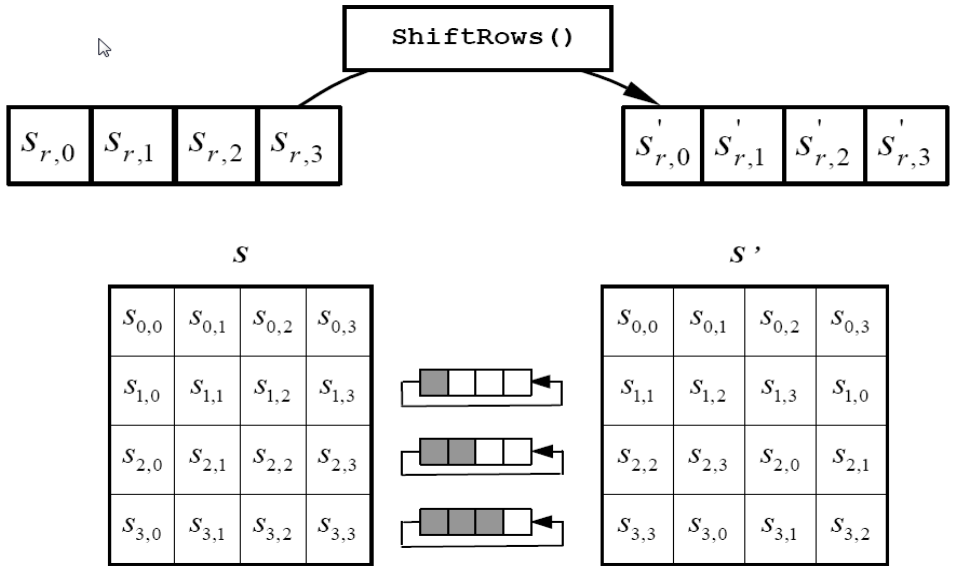
1. Neliniaritate - Corelaţia intrări-ieşiri să fie cât mai mică posibil
2. Complexitatea algebrică.

Pentru transformarea tuturor octeţilor se foloseşte un singur S-box. Cu siguranţă asta nu este o necessitate, transformarea SubBytes putînd fi cu uşurinţă definită cu S-boxuri diferite pentru fiecare octet

Inversa transformării se obţine aplicând fiecărui octet transformarea afină inversă, după care se ia inversul multiplicativ din $GF(2^8)$ (dacă octetul nu este nul).

Transformarea ShiftRows (stare) modifică octeţii ultimilor 3 linii permutându-i ciclic cu un număr diferit de octeţi. Prima linie rămâne nemodificată .

Metoda transpoziţiei asigură, în cadrul sistemelor criptografice, realizarea difuziei: împrăştierea proprietăţilor statistice ale textului clar în textul cifrat.

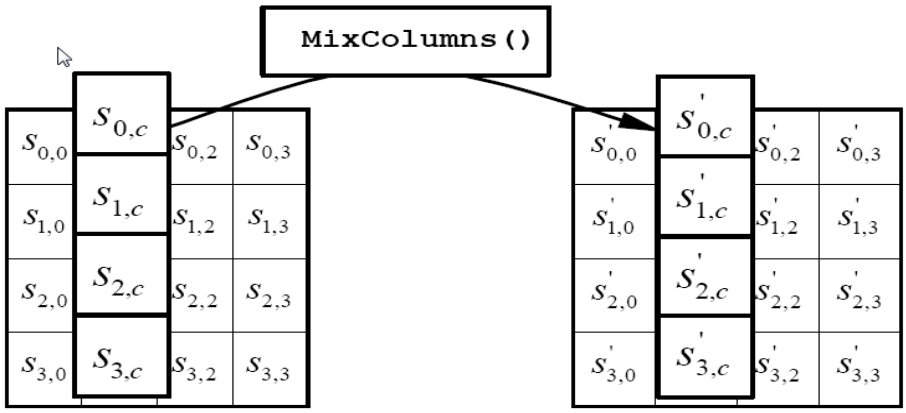


Observăm că se modifică doar poziţia octeţilor nu şi valoarea lor.

Inversa transformării *ShiftRow* constă în permutarea ciclică spre stânga cu Nb -Ci octeţi pentru linia i (1 < i < 3); în acest fel, fiecare octet aflat pe poziţia *j* în linia *i* se deplasează pe poziţia j + Nb - Ci (mod Nb).

Transformarea MixColumns(stare)

Această transformare modifică coloana matricei de stare, transformand fiecare coloana intr-un polinom cu patru termeni peste GF(2^8) .



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

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

Criterii de proiectare:

1. Dimensiunea. Transformarea operează pe coloane de patru octeţi.
2. Liniaritate. Este de preferat liniară peste GF(2).
3. Difuzia. Trebuie să aibă putere de difuzie.
4. Performanţe pe procesoare de 8 biţi.

Coloanele matricei de stare sunt considerate polinoame peste GF(2^8) şi sunt înmulţite modulo $x^4 + 1$ cu un polinom fixat c(x), unde:

$$c(x) = 03x^3 + 01x^2 + 01x + 02.$$

Criteriul de performanţă poate fi atins dacă coeficienţii au valori simple. Înmulţirile cu 00, 01 nu implică procesare.

Bazat pe înmulţirea polinoamelor în GF(2^8). Această înmulţire se face modulo polinomul generator al corpului GF(2^8) care este:

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

Operaţia inversă este similară. Fiecare coloană este transformată prin înmulţire cu polinomul invers lui $c(X)$ modulo $X4 + 1$; acesta este:

$$d(X) = 0BX^3 + 0DX^2 + 09X + 0E$$

AddRoundKey(Stare, Cheie): Această transformare constă în aplicarea unui XOR între starea curentă şi cheia de rundă. Cheia de rundă are lungimea Nb şi este dedusă din cheia de criptare pe baza unui procedeu pe care îl descriem mai jos.

Generarea cheilor de rundă

Criteriile ce au stat la baza algoritmului de extindere al cheii au fost:

1. Eficienţa
   a. Memorie de lucru. Posibilitatea de a executa expandarea cheii folosind o mică parte a memoriei de lucru.

b. Performanţă.

2. Eliminarea simetriilor.
3. Difuzie
4. Neliniaritate.

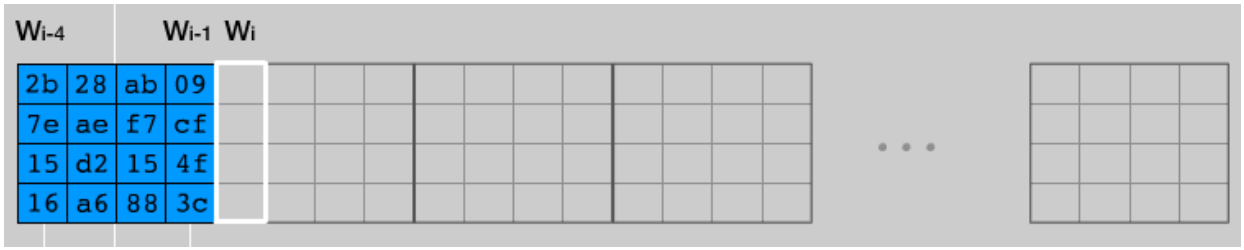
Cheile de rundă se obţin din cheia de criptare printr-o prelucrare separată, formată din două componente: extinderea cheii şi alegerea cheii de rundă.

Principiile de bază ale prelucrării sunt:

• Numărul total al biţilor din toate cheile de rundă este Nb(Nr + 1).

• Cheia de criptare este extinsă într-o Cheie Expandată.

• Cheia de rundă se obţine luând primii Nb octeţi din Cheia Expandată, care nu au fost folosiţi pentru alte chei.

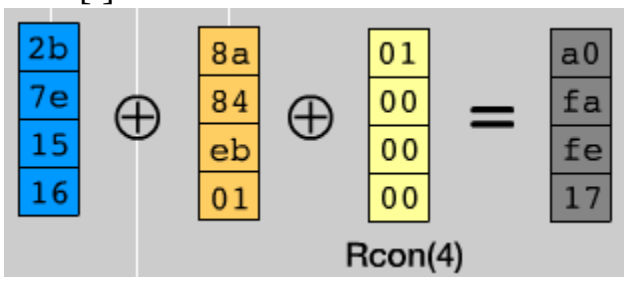


Fiecare coloana este vazuta ca un cuvant adica 32 biti.

w_0, w_1, w_2, w_3 reprezinta cheia initiala;

Cum se obtin cuvintele in pozitiile multiplu de 4 w_4, w_8, w_{12}, … ,w_{40}:

a) Se aplică RotWord() si SubWord() cuvantului anterior;
b) Se adună w_{i-4} cu rezultatul de la punctual a) si cu o constantă de rundă Rcon[i]



2b ⊕ 8a ⊕ 01 = a0

00101011 ⊕ 10001010 ⊕ 00000001 = 10100000

Funcţia SubWord() este o funcţie care are ca intrare patru octeţi. Fiecărui octet i se aplică un S-box obţinându-se astfel un octet nou.

Funcţia RotWord() are ca parametru de intrare un cuvânt [*a*0,*a*1,*a*2,*a*3] asupra căruia se execută o permutare ciclică şi resulta [*a*1,*a*2,*a*3,*a*0].

Rcon[i], conţine valorile date de [x^{i-1},{00},{00},{00}] cu x^{i-1} puteri ale lui x (x este notat ca {02}) în câmpul GF(2^8). (reţineţi că i începe cu valoarea 1 nu 0)

Este important de reţinut că rutina de expandare a cheii pentru cifrul cu cheie de 256 biţi (N_k=8) este puţin diferit de cel cu cheie de 128 şi 192 biţi. Dacă Nk=8 şi i-4 este multiplu de Nk, atunci SubWord() este aplicat lui w[i-1] mai întâi de XOR.

2. Exemplu:

AES 128 Criptarea pe o rundă:

Intrarea în runda i = 9 a algoritmului AES 128/128 pentru cifrarea textului ,,zero peste tot", cu ajutorul cheii ,,zero peste tot", este:

23	13	*aa*	2*e*	blocul de text clar
*e*7	21	*c*0	03	
8*c*	63	*c*6	*cb*	
3*c*	*db*	57	95	
*b*1	8*a*	1*d*	4*c*	cheia de rundă
*d*4	7*d*	7*b*	66	
*d*8	*b*9	*b*3	49	
*e*2	*da*	*de*	41	

Într-o rundă se execută patru transformări:

- SubBytes: transformare neliniară;
- ShiftRows: o transpoziţie a liniilor:
- MixColumns: o mixare de operaţii pe coloana;
- AddRoundKey;

Transformarea SubBytes(stare)

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

23	13	*aa*	2*e*		26	7*d*	*ac*	31
*e*7	21	*c*0	03	⇒	94	*fd*	*ba*	7*b*
8*c*	63	*c*6	*cb*		64	*fb*	*b*4	1*f*
3*c*	*db*	57	95		*eb*	*b*9	5*b*	2*a*

Găsirea octetului din S-box corespunzator octetului din stare se face astfel: pentru octetul 23 se caută în SBox elementul aflat la intersect»ia liniei 2 cu coloana 3 şi se substituie în stare elementul găsit in Sbox. 23 se va substitui cu 26
Procedeul se aplică similar pentru ceilalţi octeţi din stare astfel:

23 devine 26 (elem din S-box care se află la intersecţia liniei 2 cu coloana 3)

13 → 7d

aa → ac

2e → 31

e7 → 94

21 → fd

c0 → ba

03 → 7b

8c → 64

63 → fb

c6 → b4

cb → 1f

3c → eb

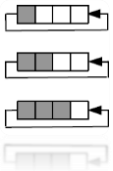
db → b9

57 → 5b

95 → 2a

Transformarea ShiftRows stare) Rutina ShiftRows acţionează asupra stării astfel: prima linie rămâne neschimbată, a doua linie se roteşte la stânga cu un octet, a treia linie se roteşte la stânga cu doi octeţi iar a patra linie se roteşte la stânga cu trei octeţi.

26	7*d*	*ac*	31
94	*fd*	*ba*	7*b*
64	*fb*	*b*4	1*f*
eb	*b*9	5*b*	2*a*



26	7*d*	*ac*	31
fd	*ba*	7*b*	94
*b*4	1*f*	64	*fb*
2*a*	*eb*	*b*9	5*b*

Transformarea MixColumns: (stare)

Rutina MixColumns presupune înmulţirea fiecărei coloane din stare cu următoarea matrice fixată:

$$\begin{matrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{matrix} * \begin{matrix} 26 & 7d & ac & 31 \\ fd & ba & 7b & 94 \\ b4 & 1f & 64 & fb \\ 2a & eb & b9 & 5b \end{matrix} =$$

$$\begin{matrix} 02*26 \oplus 03*fd \oplus b4 \oplus 2a & 02*7d \oplus 03*ba \oplus 1f \oplus eb & 02*ac \oplus 03*7b \oplus 64 \oplus b9 & 02*31 \oplus 03*94 \oplus fb \oplus 5b \\ 26 \oplus 02*fd \oplus 03*b4 \oplus 2a & 7d \oplus 02*ba \oplus 03*1f \oplus eb & ac \oplus 02*7b \oplus 03*64 \oplus b9 & 31 \oplus 02*94 \oplus 03*fb \oplus 5b \\ 26 \oplus fd \oplus 02*b4 \oplus 03*2a & 7d \oplus ba \oplus 02*1f \oplus 03*eb & ac \oplus 7b \oplus 02*64 \oplus 03*b9 & 31 \oplus 94 \oplus 02*fb \oplus 03*5b \\ 03*26 \oplus fd \oplus b4 \oplus 02*2a & 03*7d \oplus ba \oplus 1f \oplus 02*eb & 03*ac \oplus 7b \oplus 64 \oplus 02*b9 & 03*31 \oplus 94 \oplus fb \oplus 02*5b \end{matrix}$$

Fiecare elment reprezintă un polinom astfel:

01 = 00000001 = 1

Observaţii:

1. Înmulţirile cu 01 nu au sens.
2. În reprezentarea polinomială înmulţirea în GF(2^8) este echivalentă cu înmulţirea polinoamelor modulo un polinom ireductibil de grad 8. Un polinom este ireductibil dacă nu are nici un divizor mai puţin 1 şi el însuşi. Pentru Rijndael acest polinom este:

   $$m(x) = x^8 + x^4 + x^3 + x + 1 = 11b.$$

Un polinom reprezintă un octet, de aceea gradul acestuia trebuie să fie mai mic ca 8.

Calculăm elementele primei coloane:

02*26⊕03*fd⊕b4⊕2a=

$$26 = \begin{matrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ x^7 & x^6 & x^5 & x^4 & x^3 & x^2 & x^1 & x^0 \end{matrix} = x^5 + x^2 + x$$

02 = 00000010 = x

$02*26 = x(x^5 + x^2 + x) = x^6 + x^3 + x^2 = 01001100 = 4c$

$03*fd = (x + 1)(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1)$

$= x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1$

$= x^8 + x^2 + x + 1$

$(x^8 + x^2 + x + 1)$ modulo $(x^8 + x^4 + x^3 + x + 1) = x^4 + x^3 + x^2 = 00011100 = 1c$

$03 = 00000011 = x+1$

$fd = 11111101 = (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1)$

$02*26 \oplus 03*fd \oplus b4 \oplus 2a = 4c \oplus 1c \oplus b4 \oplus 2a$

$= 01001100 \oplus 00011100 \oplus 10110100 \oplus 00101010$

$= ce$

$26 \oplus 02*fd \oplus b4*03 \oplus 2a=$

$02*fd = x(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1) = x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x$

$(x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x)$ (modulo $x^8 + x^4 + x^3 + x + 1$) $= x^7 + x^6 + x^5 + 1 = 11100001 = e1$

$03*b4 = (x+1)(x^7 + x^5 + x^4 + x^2)$

$= x^8 + x^6 + x^5 + x^3 + x^7 + x^5 + x^4 + x^2 = x^8 + x^7 + x^6 + x^4 + x^3 + x^2$

$b4 = 10110100 = x^7 + x^5 + x^4 + x^2$

$(x^8 + x^7 + x^6 + x^4 + x^3 + x^2)$ (modulo $x^8 + x^4 + x^3 + x + 1$) $= x^7 + x^6 + x^2 + x + 1$

$= 11000111 = c7$

$26 \oplus 02*fd \oplus b4*03 \oplus 2a = 26 \oplus e1 \oplus c7 \oplus 2a$

$= 00100110 \oplus 11100001 \oplus 11000111 \oplus 00101010$

$= 2a$

$26 \oplus fd \oplus b4*02 \oplus 2a*03=$

$b4*02 = x(x^7 + x^5 + x^4 + x^2) = x^8 + x^6 + x^5 + x^3$

($x^8 + x^6 + x^5 + x^3$) (modulo $x^8 + x^4 + x^3 + x + 1$) = $x^6 + x^5 + x^4 + x + 1$ = 01110011 = 73

2a = 00101010 = $x^5 + x^3 + x$

2a*03 = (x+1)($x^5 + x^3 + x$) = $x^6 + x^4 + x^2 + x^5 + x^3 + x$

= 01111110 = 7e

26⊕fd⊕b4*02⊕2a*03= 00100110⊕11111101⊕01110011⊕01111110= d6

03*26⊕fd⊕b4⊕2a*02=

03*26 = (x+1)($x^5 + x^2 + x$) = $x^6 + x^3 + x^2 + x^5 + x^2 + x = x^6 + x^3 + x^5 + x$ = 01101010=6a

2a*02= x($x^5 + x^3 + x$)= $x^6 + x^4 + x^2$= 01010100 = 54

03*26⊕fd⊕b4⊕2a*02= 01101010⊕11111101⊕10110100⊕01010100= 77

Calculăm valorile de pe coloana a doua a matricei rezultat:

02*7d⊕03*ba⊕01*1f⊕01*eb= 02*7d ⊕03*ba ⊕1f ⊕ ef

02*7d = 00000010 * 01111101 = x ($x^6 + x^5 + x^4 + x^3 + x^2 + 1$) = $x^7 + x^6 + x^5 + x^4 + x^3 + x$

= 11111010 = fa

03 * ba = 00000011 * 10111010 = (x+1)($x^7 + x^5 + x^4 + x^3 + x$)

= $x^8 + x^6 + x^5 + x^4 + x^2 + x^7 + x^5 + x^4 + x^3 + x$

= $x^8 + x^6 + x^2 + x^7 + x^3 + x$

03 * ba = $x^8 + x^7 + x^6 + x^3 + x^2 + x$ (modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^7 + x^6 + x^4 + x^2 + 1$ = 11010101 = d5

02*7d⊕03*ba⊕01*1f⊕01*eb= 02*7d ⊕03*ba ⊕1f ⊕ ef

= 11111010 ⊕11010101 ⊕00011111 ⊕11101011 =db

7d⊕02*ba⊕03*1f⊕eb =

7d*01 = $(x^6 + x^5 + x^4 + x^3 + x^2 + 1)*1$ = 7d

7d = 01111101 = $x^6 + x^5 + x^4 + x^3 + x^2 + 1$

01 = 00000001 = 1

02*ba = x($x^7 + x^5 + x^4 + x^3 + x$) = $x^8 + x^6 + x^5 + x^4 + x^2$(modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^6 + x^5 + x^3 + x^2 + x + 1$ = 01101111 = 6f

03*1f = (x+1)($x^4 + x^3 + x^2 + x + 1$)= $x^5 + x^4 + x^3 + x^2 + x + x^4 + x^3 + x^2 + x + 1$

= $x^5 + 1$= 00100001 = 21

7d⊕02*ba⊕03*1f⊕eb = 01111101⊕01101111⊕00100001⊕11101011 = 11011000 = d8

7d⊕ba⊕02*1f⊕03*eb =

02*1f = x ($x^4 + x^3 + x^2 + x + 1$) = $x^5 + x^4 + x^3 + x^2 + x$ = 00111110 = 3e

03*eb = (x+1)($x^7 + x^6 + x^5 + x^3 + x + 1$) = $x^8 + x^7 + x^6 + x^4 + x^2 + x + x^7 + x^6 + x^5 + x^3 + x + 1$

= $x^8 + x^5 + x^4 + x^3 + x^2 + 1$(modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^5 + x^2 + x$ = 00100110 = 26

eb = 11101011 = $x^7 + x^6 + x^5 + x^3 + x + 1$

7d⊕ba⊕02*1f⊕03*eb = 01111101⊕10111010⊕00111110⊕00100110 = 11011111 = df

03*7d⊕ba⊕1f⊕02*eb =

03*7d = (x+1)($x^6 + x^5 + x^4 + x^3 + x^2 + 1$)= $x^7 + x^6 + x^5 + x^4 + x^3 + x + x^6 + x^5 + x^4 + x^3 + x^2 + 1$

$= x^7 + x^2 + x + 1 = 10000111 = 87$

02*eb = $x(x^7 + x^6 + x^5 + x^3 + x + 1) = x^8 + x^7 + x^6 + x^4 + x^2 + x$ (modulo $x^8 + x^4 + x^3 + x + 1$)

$= x^7 + x^6 + x^3 + x^2 + 1 = 11001101 = cd$

03*7d⊕ba⊕1f⊕02*eb = 10000111⊕10111010⊕00011111⊕11001101 = 11101111 = ef

Calculăm a treia coloană:

02*ac⊕03*7b⊕64⊕b9 =

02*ac = $x\ (x^7 + x^5 + x^3 + x^2) = x^8 + x^6 + x^4 + x^3$(modulo $x^8 + x^4 + x^3 + x + 1$)

$= x^6 + x + 1 = 01000011 = 43$

03*7b = $(x+1)(\ x^6 + x^5 + x^4 + x^3 + x + 1) = x^7 + x^6 + x^5 + x^4 + x^2 + x + x^6 + x^5 + x^4 + x^3 + x + 1$

$= x^7 + x^3 + x^2 + 1 = 10001101 = ad$

7b = 01111011= $x^6 + x^5 + x^4 + x^3 + x + 1$

02*ac⊕03*7b⊕64⊕b9 = 01000011⊕10001101⊕01100100⊕10111001 = 00010011 = 13

ac⊕02*7b⊕03*64⊕b9 =

02*7b = $x(\ x^6 + x^5 + x^4 + x^3 + x + 1) = x^7 + x^6 + x^5 + x^4 + x^2 + x = 11110110 = f6$

03*64 = $(x+1)(\ x^6 + x^5 + x^2) = x^7 + x^6 + x^3 + x^6 + x^5 + x^2 = x^7 + x^3 + x^5 + x^2 = 10101100 = ac$

ac⊕02*7b⊕03*64⊕b9 = 10101100⊕11110110⊕10101100⊕10111001 = 01001111 = 4f

ac⊕7b⊕02*64⊕03*b9 =

02*64 = $(x)(\ x^6 + x^5 + x^2) = x^7 + x^6 + x^3 = 11001000 = c8$

03*b9 = (x+1)($x^7 + x^5 + x^4 + x^3 + 1$) = $x^8 + x^6 + x^5 + x^4 + x + x^7 + x^5 + x^4 + x^3 + 1$

= $x^8 + x^6 + x + x^7 + x^3 + 1$ (modulo $x^8 + x^4 + x^3 + x + 1$) = + $x^7 + x^6 + x^4$ = 11010000 = d0

ac⊕7b⊕02*64⊕03*b9 = 10101100⊕01111011⊕11001000⊕11010000 = 11001111 = cf

03*ac⊕7b⊕64⊕02*b9 =

03*ac =(x+1) ($x^7 + x^5 + x^3 + x^2$)

= $x^8 + x^6 + x^4 + x^3 + x^7 + x^5 + x^3 + x^2$ (modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^7 + x^6 + x^5 + x^3 + x^2 + x + 1$= 11101111 = ef

02*b9 = x($x^7 + x^5 + x^4 + x^3 + 1$) = $x^8 + x^6 + x^5 + x^4 + x$(modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^6 + x^5 + x^3 + 1$ = 01101001 = 69

03*ac⊕7b⊕64⊕02*b9 = 11101111⊕01111011⊕01100100⊕01101001 = 10011001 = 99

Ultima coloană:

02*31⊕03*94⊕fb⊕5b =

02*31 = x($x^5 + x^4 + 1$) = $x^6 + x^5 + x$ = 01100010 = 62

03*94 = (x+1) ($x^7 + x^4 + x^2$) = $x^8 + x^5 + x^3 + x^7 + x^4 + x^2$(modulo $x^8 + x^4 + x^3 + x + 1$)

= + $x^7 + x^5 + x^2 + x + 1$= 10100111 = a7

02*31⊕03*94⊕fb⊕5b = 01100010⊕10100111⊕11111011⊕01011011 = 01100101 = 65

31⊕02*94⊕03*fb⊕5b =

02*94 = x($x^7 + x^4 + x^2$) = $x^8 + x^5 + x^3$(modulo $x^8 + x^4 + x^3 + x + 1$) = $x^5 + x^4 + x + 1$

= 00110011 = 33

03*fb = (x+1)($x^7 + x^6+x^5+x^4 + x^3 + x + 1$)

= $x^8 + x^7+x^6+x^5 + x^4 + x^2 + x + x^7 + x^6+x^5+x^4 + x^3 + x + 1$(modulo $x^8 + x^4 + x^3 + x + 1$)

= $x^4 + x^2 + x$ = 00010110 = 16

31⊕02*94⊕03*fb⊕5b = 00110001⊕00110011⊕00010110⊕01011011 = 01001111 = 4f

31⊕94⊕02*fb⊕03*5b =

02*fb = x($x^7 + x^6+x^5+x^4 + x^3 + x + 1$)

= $x^8 + x^7+ x^6+x^5 + x^4 + x^2 + x$(modulo $x^8 + x^4 + x^3 + x + 1$) = $x^7+ x^6+x^5 + x^3 + x^2 + 1$

= 11101101 = ed

03*5b = (x+1)($x^6 + x^4 + x^3 + x + 1$) = $x^7+ x^5 + x^4 + x^2 + x + x^6 + x^4 + x^3 + x + 1$

= $x^7+ x^5 + x^2 + x^6 + x^3 + 1$ = 11101101 = ed

5b = 01011011 = $x^6 + x^4 + x^3 + x + 1$

31⊕94⊕02*fb⊕03*5b = 00110001⊕10010100⊕11101101⊕11101101 = 10100101 = a5

03*31⊕94⊕fb⊕02*5b =

03*31 = (x+1)($x^5 +x^4 +1$) = $x^6 +x^5 + x + x^5 +x^4 +1$= 01010011 = 53

02*5b = x($x^6 + x^4 + x^3 + x + 1$) = $x^7+ x^5 + x^4 + x^2 + x$ =10110110 = b6

03*31⊕94⊕fb⊕02*5b = 01010011⊕10010100⊕11111011⊕10110110 = 10001010 = 8a

Matricea rezultat este:

ce	*db*	13	65
2*a*	*d*8	4*f*	4*f*
*d*6	*df*	*cf*	*a*5
77	*ef*	99	8*a*

AddRoundKey(Stare, Cheie): Această transformare constă în aplicarea unui XOR între starea curentă şi cheia de rundă..

ce	*db*	13	65		*b*1	8*a*	1*d*	4*c*		7*f*	51	0*e*	29
2*a*	*d*8	4*f*	4*f*	⊕	*d*4	7*d*	7*b*	66	=	*fe*	*a*5	34	29
*d*6	*df*	*cf*	*a*5		*d*8	*b*9	*b*3	49		0*e*	66	7*c*	*ec*
77	*ef*	99	8*a*		*e*2	*da*	*de*	41		95	35	47	*cb*

ce⊕b1 = 11001110 ⊕10110001 = 01111111 = 7f

db⊕8a = 11011011 ⊕ 10001010 = 01010001 = 51

13⊕1d = 00010011 ⊕00011101 = 00001110 = 0e

65⊕4c = 01100101 ⊕01001100 = 00101001 = 29

2a⊕d4 = 00101010 ⊕11010100 = 11111110 = fe

d8⊕7d = 11011000 ⊕ 01111101 = 10100101 = a5

4f⊕7b = 01001111 ⊕ 01111011 = 00110100 = 34

4f⊕66 = 01001111 ⊕ 01100110 = 00101001 = 29

d6⊕d8 = 11010110 ⊕ 11011000 = 00001110 = 0e

df⊕b9 = 11011111 ⊕ 10111001 = 01100110 = 66

cf⊕b3 = 11001111 ⊕ 10110011 = 01111100 = 7c

a5⊕49 = 10100101 ⊕ 01001001 = 11101100 = ec

77⊕e2 = 01110111 ⊕ 11100010 = 10010101 = 95

ef⊕da = 11101111 ⊕ 11011010 = 00110101 = 35

99⊕de = 10011001 ⊕ 11011110 = 01000111 = 47

8a⊕41 = 10001010 ⊕ 01000001 = 11001011 = cb

Observaţie:

1. Matricea rezultată constituie intrare pentru runda următoare.
2. Operaţiile deşi multe sunt simplu de implementat astfel înmulţirea cu 00 şi 01nu implică nici o operaţie, înmulţirea cu 02 poate fi implementată ca o rutină dedicată iar înmulţirea cu 03 poate fi implementată ca o înmulţire cu 02 şi operaţia XOR.
3. Decriptarea se face prin inversarea transformărilor de mai sus.

Bibliografie:

1. FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001.

   http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

2. Joan Daemen and Vincent Rijmen, The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag 2002 .

3. Joan Daemen, Vincent Rijmen "AES Proposal: Rijndael ".

4. Algebraic Aspects of the Advanced Encryption Standard by Carlos Cid, Sean Murphy and Matthew Robshaw, 2006 Springer Science^-Business Media, LLC