Стандарт AES

Чому потрібен новий стандарт?

- 1997 рік – Distributed NET, атака «грубою силою» на DES - 96 днів;
- 1998 рік – 41 день;
- 1999 рік – 2 дні і 8 годин;
- 1999 рік – 22 години.
- ==========
- Застаріла архітектура;
- Модифікації (3DES) – повільні.

Конкурс симетричних алгоритмів

- 2 січня 1997 року – Національний інститут стандартів і технологій США (NIST) оголошує конкурс на новий стандарт симетричного шифрування.
- Претендентами було 15 алгоритмів, з яких після першого етапу залишилося 5:
  - MARS (IBM) – модифікований ланцюг Фейстеля;
  - RC6 (Ron Rivest, RSA - модифікований ланцюг Фейстеля);
  - Rijndael (V.Rijmen, J.Daemen - SP-мережа);
  - Serpent (Ross Anderson, Eli Biham, Lars Knudsen

Конкурс симетричних алгоритмів

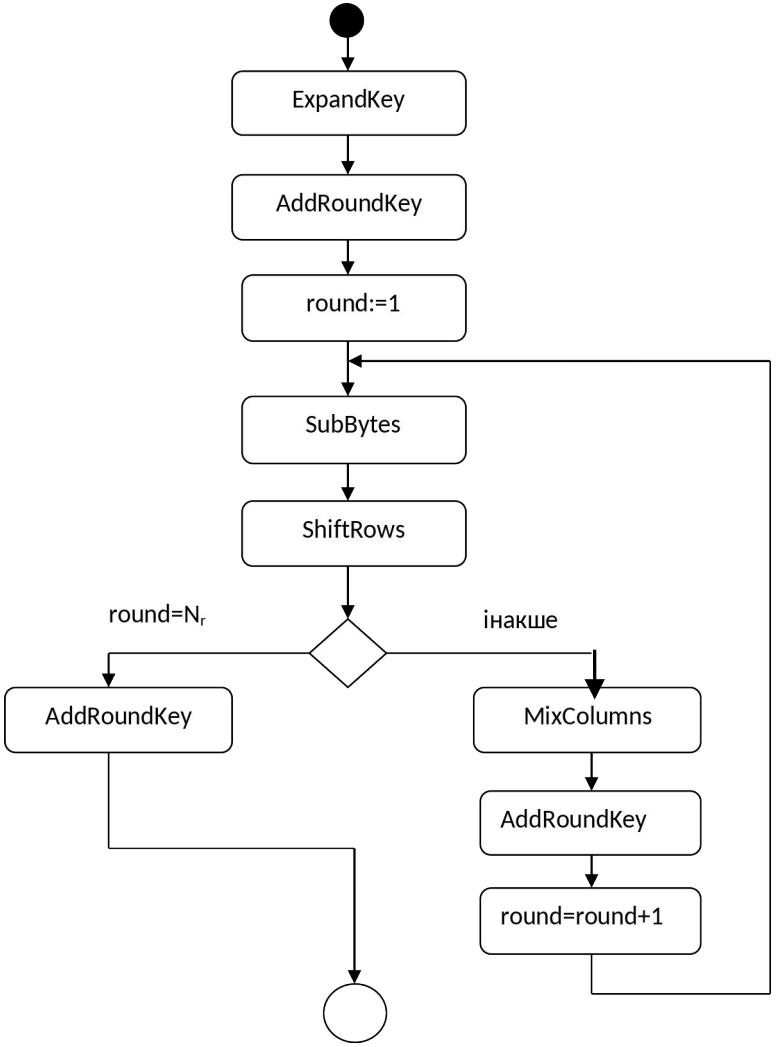
- Переможець – алгоритм Rijndael, який отримав назву AES.
- 2001 – оголошено переможця;
- 2002 рік – випущено стандарт;
- FIPS 197 – специфікація шифру та опис стандарту.

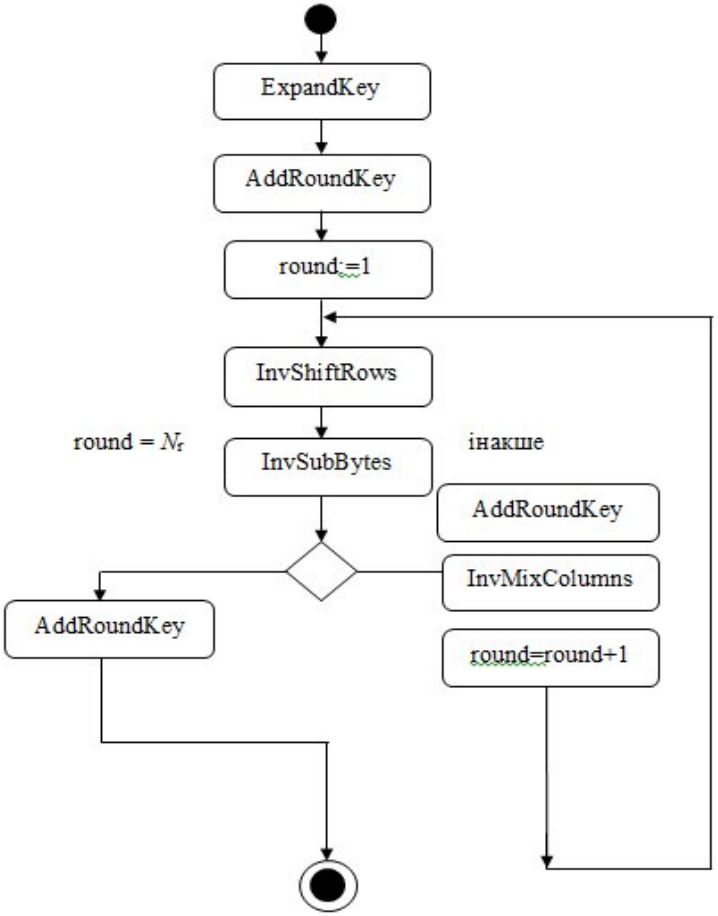
Основні параметри алгоритму

- Архітектура: Substitution-Permutation network (SP-мережа);
- Вхідний блок: 128/192/256 бітів (128 бітів – AES); за цикл обробляється цілий блок;
- Довжина ключа: 128/192/256 бітів.
- К-сть раундів: 10/12/14 (залежить від довжин вхідного блоку та ключа):

N_r	$N_b = 4$ (128 бітів)	$N_b = 6$ (192 біти)	$N_b = 8$ (256 бітів)
$N_k = 4$ (128 бітів)	10	12	14
$N_k = 6$ (192 біти)	12	12	14
$N_k = 8$ (256 бітів)	14	14	14

Блок-схема алгоритму



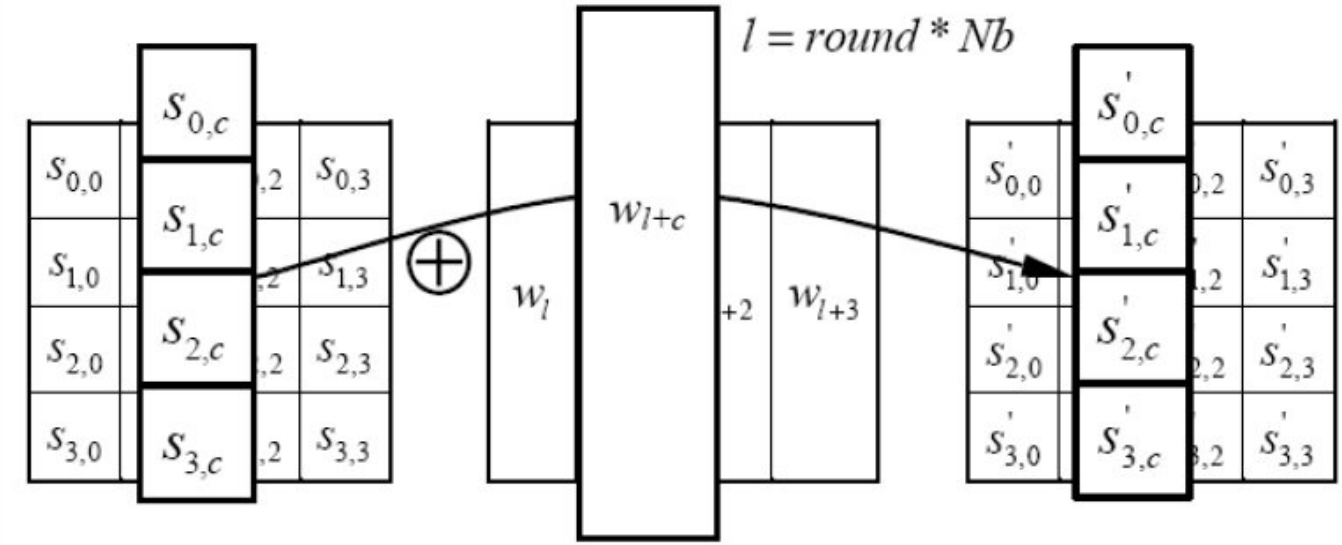


Процедура *ExpandKey*

- Процедура *ExpandKey* – розгортання ключа шифрування:
  - Необхідна кількість раундових ключів – $N_b \times (N_r+1)$, де N_b – довжина блоку; N_r – кількість раундів;
  - Використовують заміну кожного байта ключа за таблицями заміни (функція *SubWord*);
  - Циклічний зсув процедурою *RotWord*:

Процедура *AddRoundKey*

- *AddRoundKey* – додавання раундового ключа:



Перетворення *SubBytes*

- Перетворення SubBytes виконує заміну кожного байта блоку за допомогою таблиці замін:

		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

Наприклад, байт {fe} буде замінено на {bb}.

Перетворення inv*SubBytes*

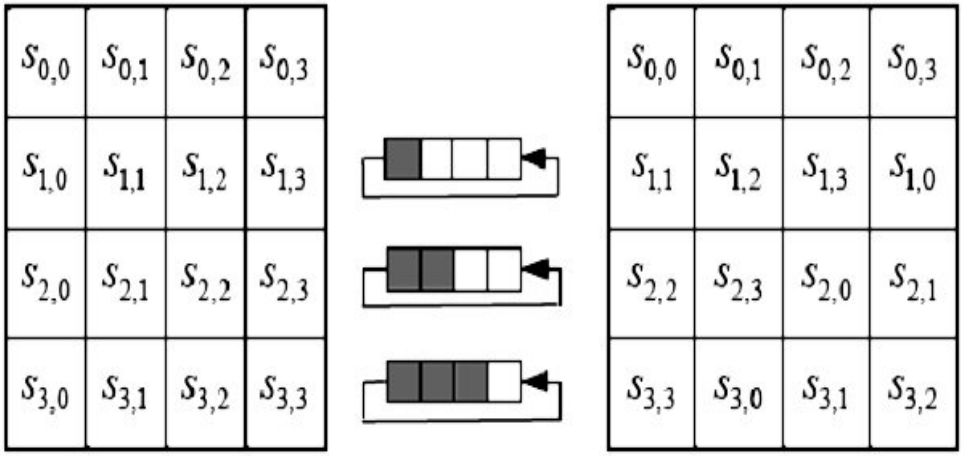
- Під час розшифрування використовують обернену таблицю:

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

Наприклад, байт {bb} буде замінено на {fe}.

Перетворення *ShiftRows*

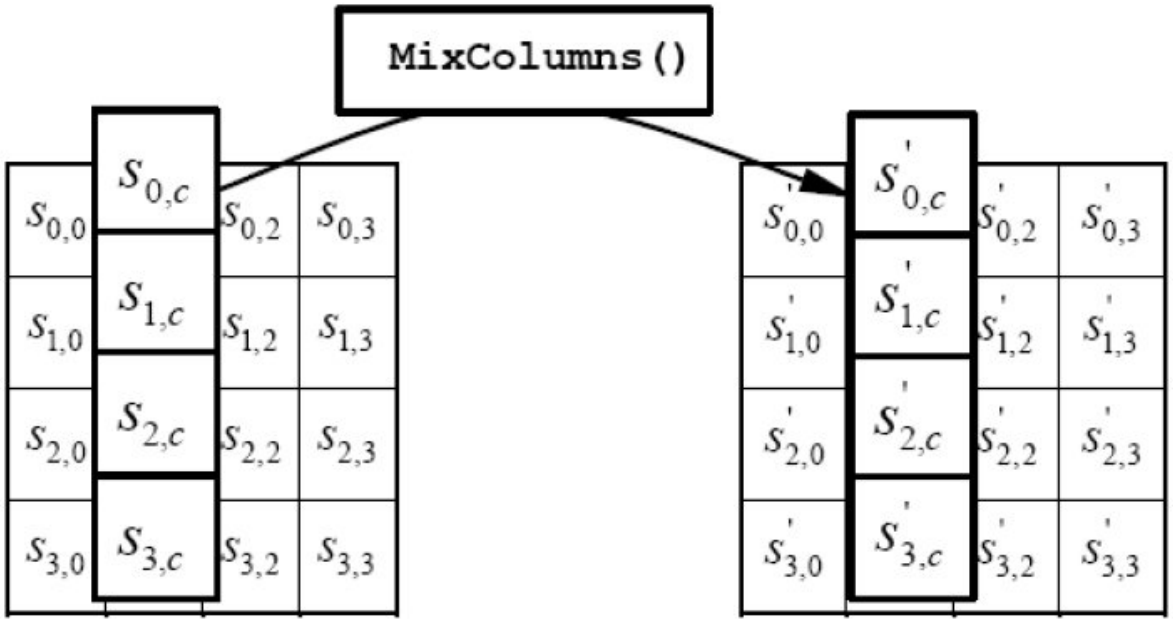
- Суть перетворення полягає в циклічному зсуві рядків стану. Перший рядок залишається незмінним, другий – зсувається вліво на один байт, а перший байт записується в кінець рядка. Третій зсувається на два байти, а четвертий – на три. Обернене перетворення – зсув вправо.



Перетворення *MixColumns*

Це перетворення виконується як множення квадратної матриці четвертого порядку на кожен стовпчик стану:

$$\begin{bmatrix} b_0 \\ \\ \\ \end{bmatrix} \quad \begin{bmatrix} 02 & 03 & 01 & 01 \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} a_0 \\ \\ \\ \end{bmatrix}$$



По суті, ц … гься за модулем …

Цей полі… ерсія MixColum…

Добуток …

У матрич… зується при розш…

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Тут множення також відбувається за правилами $GF(2^8)$.

Переваги AES

- Найбільш досліджений з сучасних криптоалгоритмів;
- Забезпечує високу криптографічну стійкість, стандартизований ISO;
- Оптимізований для виконання на 32-бітових платформах;
- Наявність апаратних акселераторів (включені до інструкцій Core iX Haswell)

Недоліки AES

- Наявність теоретичних атак зі складністю, меншою за атаку «грубою силою»;
- Знайдено недоліки алгоритму розгортання ключа;
- Не оптимізований для 64-бітових платформ;
- Деяка моральна застарілість;
- Наявність сучасніших алгоритмів криптографічного перетворення, в т.ч. тих, які стали відомими на конкурсі SHA-3.
- ==========
- Недовіра до апаратної підтримки алгоритму у

Порівняння алгоритмів

No	Категорії	DES	AES
1.	Архітектура	Мережа Фейстеля	SP-мережа
2.	Вхідний блок	64 біта	128 бітів (128/192/256)
3.	Довжина ключа	56+8 бітів	128/192/256
4.	Кількість раундів обробки	16	10/12/14
5.	Еквівалентність прямого/оберненого перетворень	Ключі подаються в оберненому порядку	Обернений порядок процедур, ключів, блоків заміни