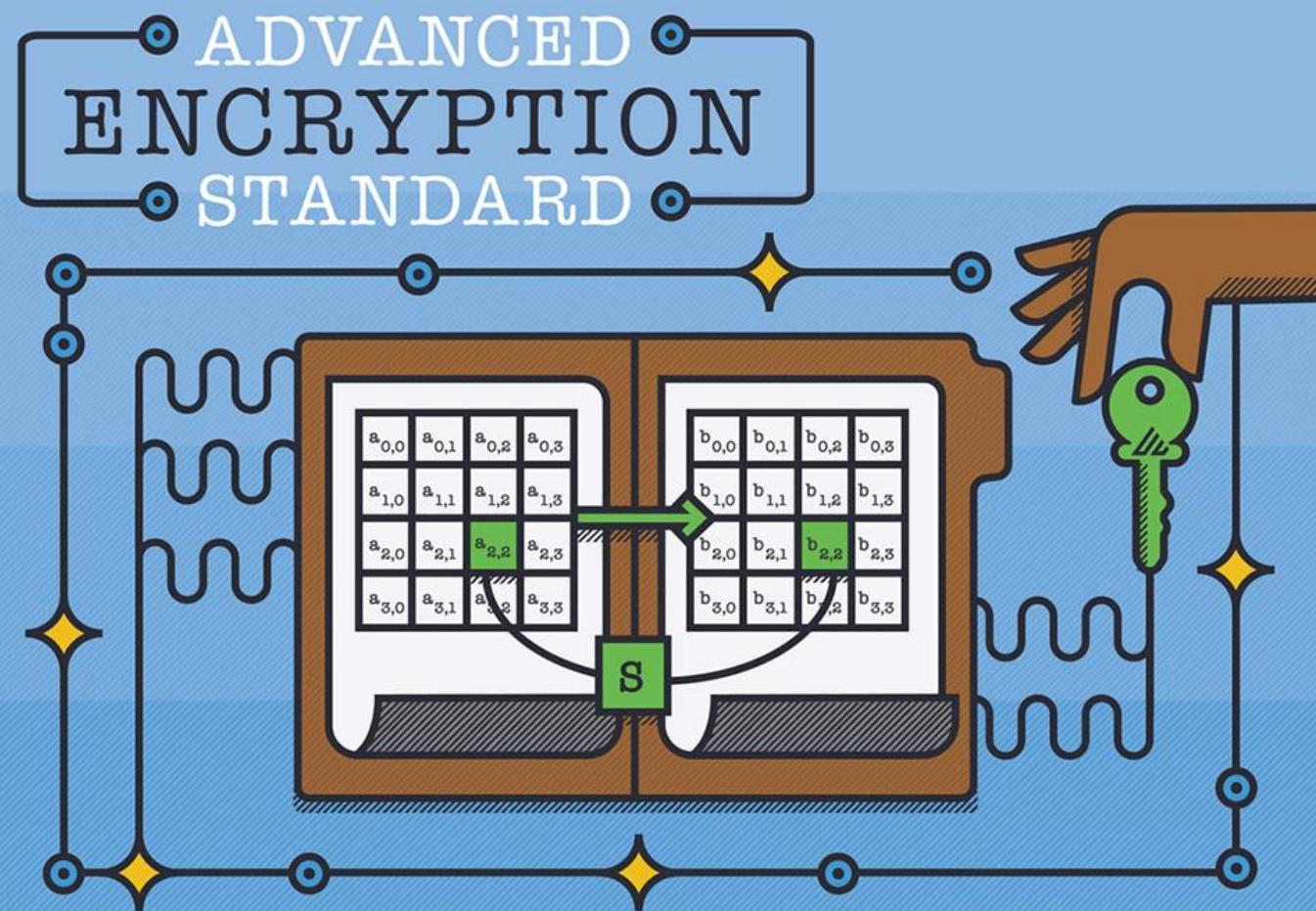


# ЛЕКЦІЯ 5

## Удосконалений стандарт шифрування AES



# План

1. Історія появи AES

2. Математична база

3. Алгоритм AES

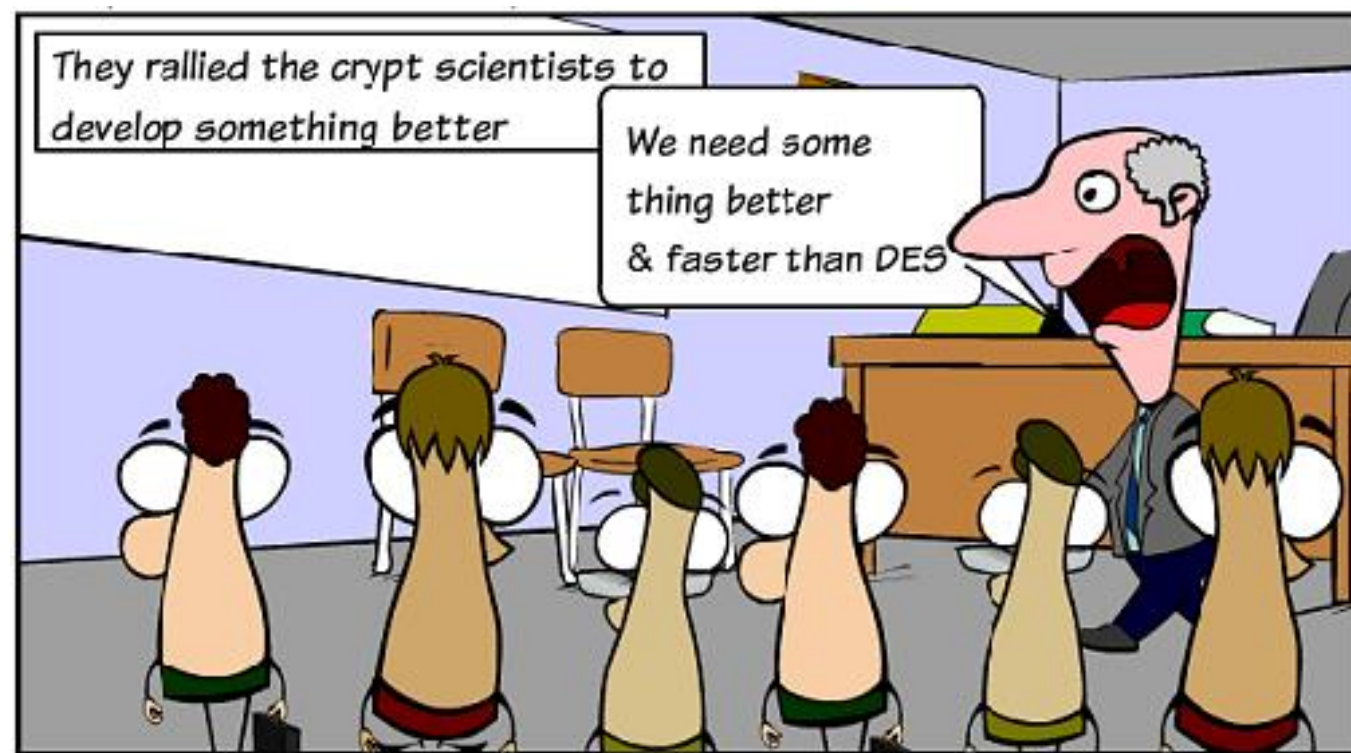
4. Режимы виконання блокових шифрів

# 1. Історія появи AES

У 1997 р. NIST оголосив конкурс на **новий блоковий симетричний стандарт шифрування**

Алгоритм повинен:

- 1) бути **симетричним**;
- 2) бути **блоковим**;
- 3) мати довжину блока 128 біт і підтримувати **три довжини** ключа: 128, 192 і 256 біт.



# 1. Історія появи AES

Алгоритми-фіналісти		
Алгоритм	Хто створив	Країна
MARS	IBM	US
RC6	R.Rivest & Co	US
Rijndael	V.Rijmen & J.Daemen	BE
Serpent	Universities	IS, UK, NO
TwoFish	B.Schneier & Co	US

# 1. Історія появи AES

У жовтні 2000 р. конкурс завершився. Переможцем став бельгійський алгоритм **Rijndael**, після чого був затверджений як стандарт та отримав назву **AES** (2001 рік)

Автори шифру **Rijndael** – бельгійські криптографи **Вінсент Реймен** та **Йоан Дамен**



Йоан  
Дамен

Вінсент  
Реймен



# 2. Математична база

## Поле Галуа

Скінченне поле  $GF(2^8)$  складається з многочленів вигляду

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0,$$

де  $a_i \in \{0,1\}$ .

У вигляді многочлена  $a(x)$  скінченого поля  $GF(2^8)$  можна подати будь-який байт, що складається з бітів  $a_7a_6a_5a_4a_3a_2a_1a_0$ .

### Приклад 2.1:

Байт: 01011010.

$$\begin{aligned} \text{Многочлен: } & 0 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 0 = \\ & = x^6 + x^4 + x^3 + x. \end{aligned}$$

# 2. Математична база

## Додавання байтів

$$\forall a(x), b(x) \in GF(2^8)$$

$$a(x) + b(x) = c(x) = c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0,$$

$$\text{де } c_i = a_i \oplus b_i$$

### Приклад 2.2:

У двійковій формі:

$$\begin{array}{r} \oplus \quad 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \\ \quad 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \\ \hline \quad 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \end{array}$$

У вигляді многочленів додавання коефіцієнтів при однакових степенях відбувається за модулем 2 ( $1 \cdot x^7 + 1 \cdot x^7 = (1 \oplus 1) \cdot x^7 = 0 \cdot x^7$ ):

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

## 2. Математична база

### Множення байтів

Для множення у полі  $GF(2^8)$  в AES використовується нерозкладний многочлен  $m(x) = x^8 + x^4 + x^3 + x + 1$

Два елементи поля  $GF(2^8)$  множать за модулем  $m(x)$  так:

- 1) Множать як звичайні многочлени;
- 2) Проміжний результат ділять на  $m(x)$  і за остаточний результат приймають остачу від ділення.



# 2. Математична база

## Приклад 2.3:

$$(x^6 + x^5 + x^4 + x^2) \cdot (x^7 + x^5 + x^4 + x) = x^{13} + x^{11} + x^{10} + x^7 + x^{12} + x^{10} + x^9 + x^6 + x^{11} + x^9 + x^8 + x^5 + x^9 + x^7 + x^6 + x^3 = x^{13} + x^{12} + x^9 + x^8 + x^5 + x^3$$

$$\begin{array}{r|l} x^{13} + x^{12} + x^9 + x^8 + x^5 + x^3 & x^8 + x^4 + x^3 + x + 1 \\ x^{13} + x^9 + x^8 + x^6 + x^5 & \hline x^{12} + x^6 + x^3 & \\ x^{12} + x^8 + x^7 + x^5 + x^4 & \\ \hline x^8 + x^7 + x^6 + x^5 + x^4 + x^3 & \\ x^8 + x^4 + x^3 + x + 1 & \\ \hline x^7 + x^6 + x^5 + x + 1 & \end{array}$$

$$(x^{13} + x^{12} + x^9 + x^8 + x^5 + x^3) \bmod (x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + x^5 + x + 1$$

# 3. Алгоритм AES

Основним елементом, яким оперує AES, є **байт** – послідовність **8 біт**, що обробляються як єдине ціле (в **шістнадцятковій** системі числення)

Розмір блоку **128 біт**

Довжина ключа може бути **128, 192** або **256 бітів**

AES базується на архітектурі **SQUARE** (КВАДРАТ), для якої характерно:

- 1) представлення блоку у вигляді масиву байтів;
- 2) шифрування за один раунд всього блоку даних;
- 3) виконання криптографічних перетворень, як над окремими байтами, так і над рядками і стовпцями.

# 3. Алгоритм AES

Блок проміжного результату називають  
**станом**

Матриця стану має 4 рядки та 4 стовпці ( $Nb$ )

$$\begin{pmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{pmatrix}$$

## Приклад 3.4:

Відкритий текст: A SECRET MESSAGE

У шістнадцятковому вигляді:

41 20 53 45 43 52 45 54 20 4D 45 53 53 41 47 45

41	43	20	53
20	52	4D	41
53	45	45	47
45	54	53	45

# 3. Алгоритм AES

Ключ: матриця байтів, яка має 4 рядки і кількість стовпців ( $Nk$ ), що дорівнює довжині ключа, поділеній на 32

Матриця ключа при  $Nk=4$ :

$$\begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix}$$

# 3. Алгоритм AES

Кількість раундів шифрування  $N_r$  залежить від значень  $N_k$

	$N_k$ (Довжина ключа)	$N_b$ (Довжина блоку)	$N_r$ (Кількість раундів)
<b>AES-128</b>	4 (128)	4 (128)	10
<b>AES-192</b>	6 (192)		12
<b>AES-256</b>	8 (256)		14

# 3. Алгоритм AES

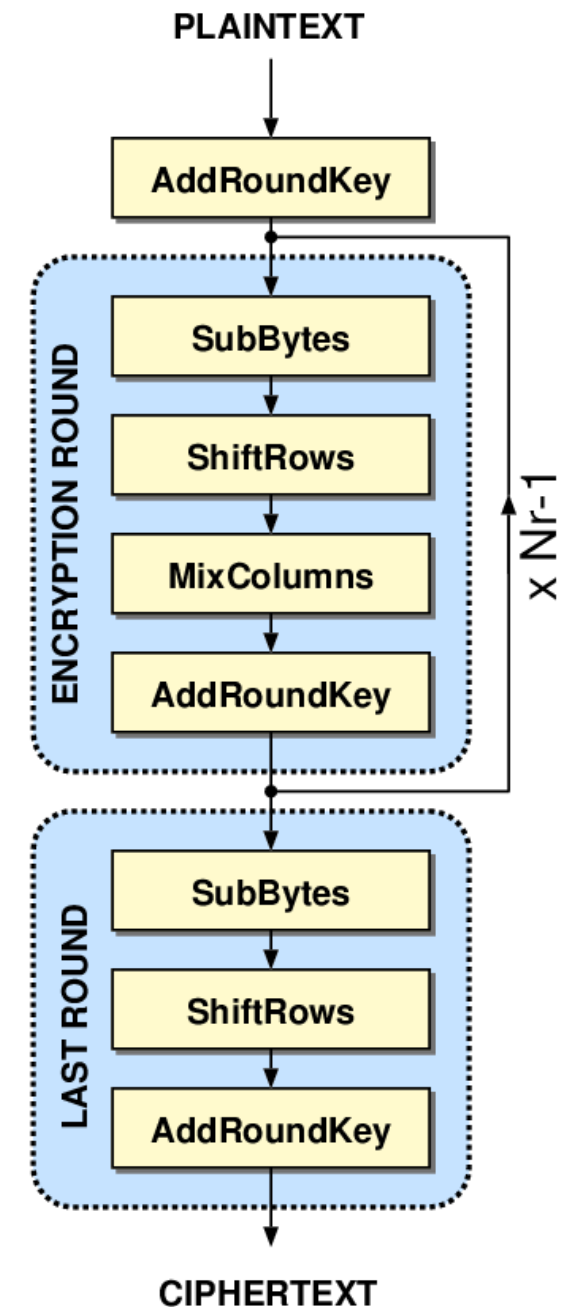
## Зашифрування за алгоритмом AES

I. Початкове додавання ключа

II.  $Nr-1$  раундів, кожен з яких складається з чотирьох етапів:

1. Підстановка байтів;
2. Зсув рядків;
3. Перемішування стовпців;
4. Додавання раундового ключа

III. Завершальний раунд  $Nr$ , в якому пропускається перемішування стовпців





# 3. Алгоритм AES

## Підстановка байтів

1. Якщо байт ненульовий, до нього шукають обернений відносно множення в полі  $GF(2^8)$ . Якщо ж байт нульовий, оберненого не існує. Тому нульовому байту 00000000 відповідає він сам.

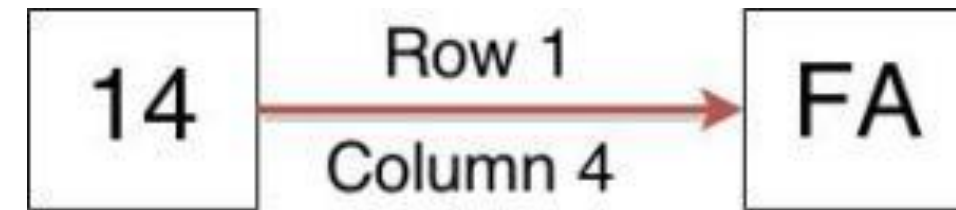
2. Над утвореним байтом виконують перетворення:

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

# 3. Алгоритм AES

На основі двох попередніх перетворень створено спеціальну таблицю замін байтів, що називається **S-боксом**

	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



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


# 3. Алгоритм AES

## Зсув рядків

Рядки стану циклічно зсувають на різні кількості байтів

<i>Nb</i>	Кількість зсувів			
	0-го рядка (-)	1-го рядка (C1)	2-го рядка (C2)	3-го рядка (C3)
4	0	1	2	3

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

# 3. Алгоритм AES

## Перемішування стовпців

Стовпці стану розглядають як многочлен над полем  $GF(2^8)$  та множать за модулем  $x^4 + 1$  на фіксований многочлен  $c(x)$ :

$$c(x) = 03_{16} \cdot x^3 + 01_{16} \cdot x^2 + 01_{16} \cdot x + 02_{16}$$

Якщо  $a(x)$  – стовпець до застосування до нього перемішування, а  $b(x)$  – після, то перетворення можна записати так:

$$b(x) = c(x) \otimes a(x),$$

або у матричному вигляді:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

# 3. Алгоритм AES

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

•

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

$$02_{16} = 0000\ 0010_2 \rightarrow x$$

$$87_{16} = 1000\ 0111_2 \rightarrow x^7 + x^2 + x + 1$$

$$03_{16} = 0000\ 0011_2 \rightarrow x + 1$$

$$6E_{16} = 0110\ 1110_2 \rightarrow x^6 + x^5 + x^3 + x^2 + x$$

$$46_{16} = 0100\ 0110_2$$

$$A6_{16} = 1010\ 0110_2$$

$$(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus (\{01\} \cdot \{46\}) \oplus (\{01\} \cdot \{A6\}) = (0000\ 0010 \cdot 1000\ 0111) \oplus (0000\ 0011 \cdot 0110\ 1110) \oplus 0100\ 0110 \oplus 1010\ 0110 = 00010101 \oplus 10110010 \oplus 0100\ 0110 \oplus 1010\ 0110 = 01000111 = 47_{16}$$

$$\{02\} \cdot \{87\} = x \cdot (x^7 + x^2 + x + 1) = (x^8 + x^3 + x^2 + x) \bmod (x^8 + x^4 + x^3 + x + 1) = x^4 + x^2 + 1$$

$$\begin{array}{r|l} x^8 + & x^3 + x^2 + x & \\ x^8 + x^4 + x^3 + & x + 1 & \\ \hline & x^4 + x^2 + 1 & \end{array} \quad \begin{array}{l} x^8 + x^4 + x^3 + x + 1 \\ \hline 1 \end{array}$$

$$\{03\} \cdot \{6E\} = (x + 1) \cdot (x^6 + x^5 + x^3 + x^2 + x) = x^7 + x^6 + x^4 + x^3 + x^2 + x^6 + x^5 + x^3 + x^2 + x = x^7 + x^5 + x^4 + x$$

# 3. Алгоритм AES

## Додавання раундового ключа

Виконується **побітове додавання** за модулем 2 раундового ключа до відповідних бітів, отриманих у попередньому раунді

Раундовий ключ  
отримують з  
**розширеного** ключа  
шифру

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

$\oplus$

DC	9B	97	38
90	49	FE	81
37	DF	72	15
B0	E9	3F	A7

=

9B	DB	34	74
A7	9B	8E	1E
A3	3B	48	57
5D	4C	99	1B





# 3. Алгоритм AES

## Розширення ключа

1. Перші  $Nk$  4-байтових слів  $W[i]$  послідовно вибираються з ключа шифру: 0-е слово – перші чотири байти, 1-е слово – другі чотири байти і т.д

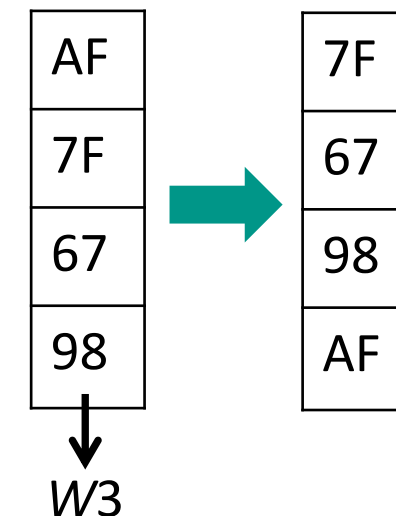
2. Якщо  $i$  кратне  $Nk$ :

2. 1. У слові  $W[i - 1]$  виконують циклічний зсув байтів за схемою:

$(a, b, c, d) \rightarrow (b, c, d, a)$  де  $a, b, c, d$  – байти

0F	47	0C	AF
15	D9	B7	7F
71	E8	AD	67
C9	59	D6	98

↓ ↓ ↓ ↓  
 $W_0$   $W_1$   $W_2$   $W_3$



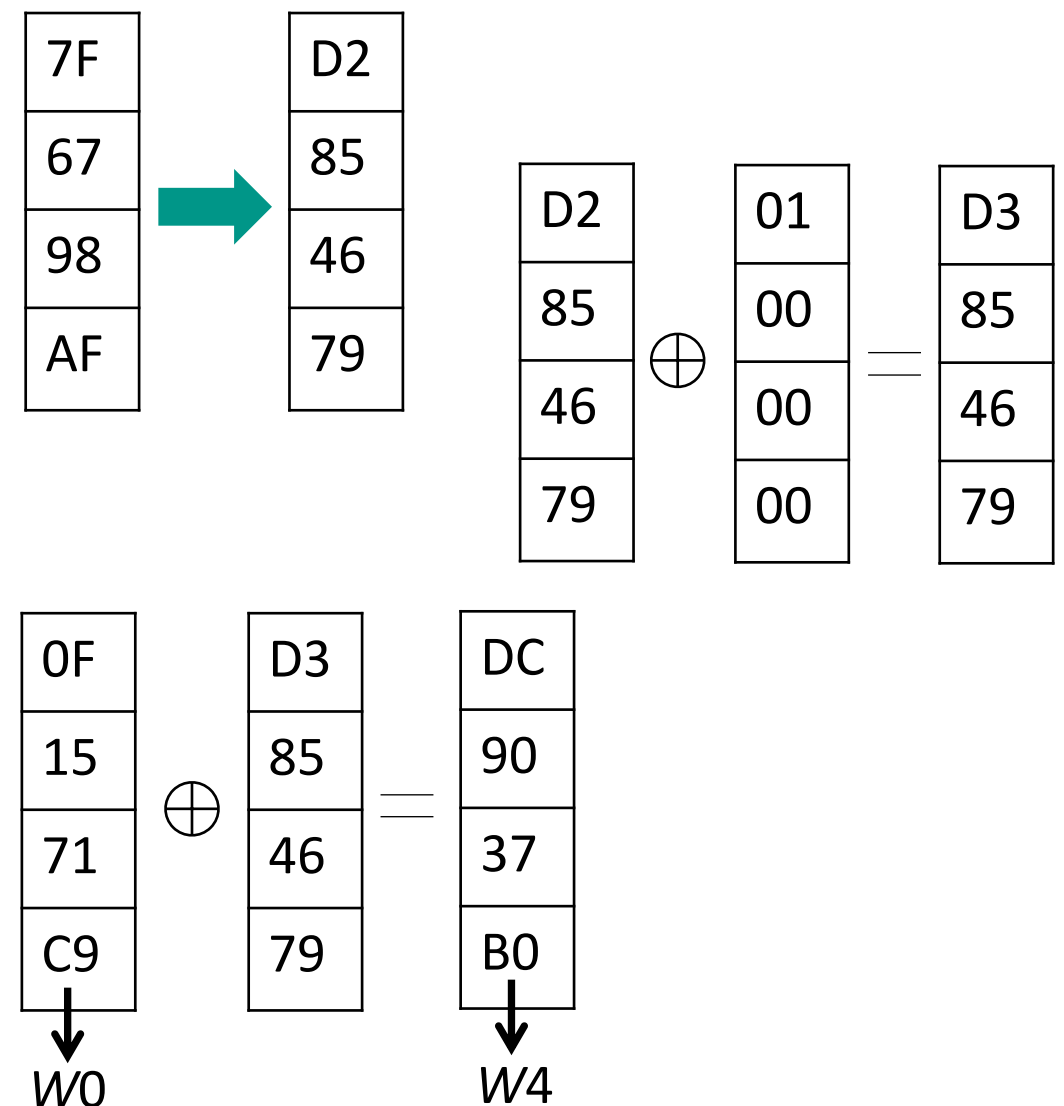
# 3. Алгоритм AES

## Розширення ключа

2.2. До кожного з 4-х байтів одержаного слова застосовують S-бокс

2.3. До результату додають раундову сталу  $Rcon$  за модулем 2

3. Решту слів  $W[i]$  визначають за формулою:  
$$W[i] = W[i - Nk] \oplus W[i - 1]$$



# 3. Алгоритм AES

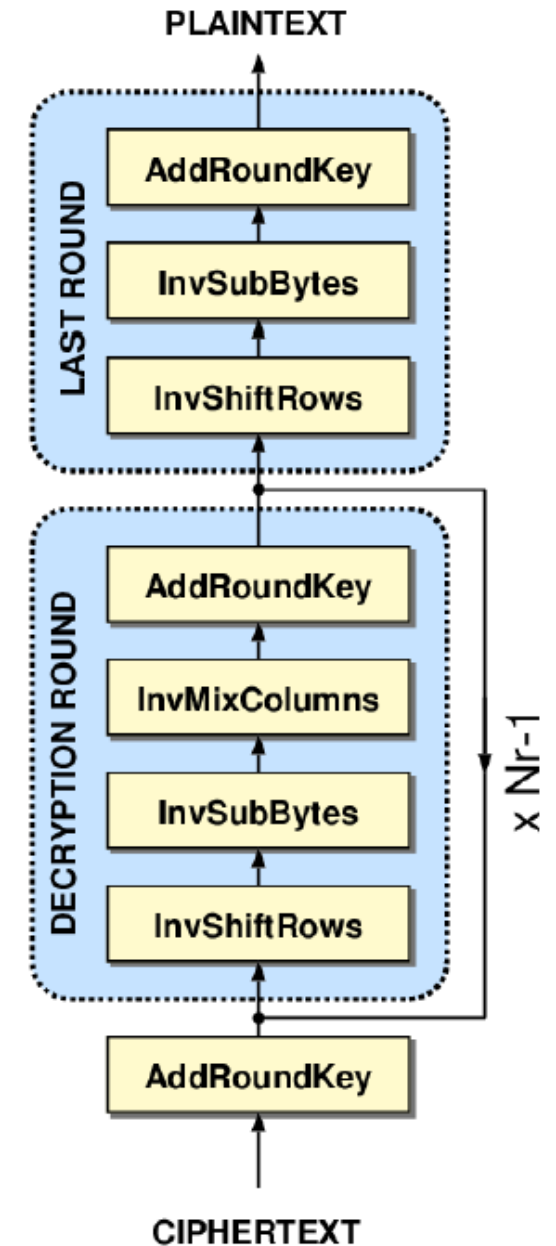
## Дешифрування за алгоритмом AES

I. Перед першим раундом дешифрування виконується операція додавання з ключем

II.  $Nr-1$  раундів, кожен з яких кожен з яких здійснює такі операції:

1. Зсув рядків в зворотному порядку;
2. Обернена операція до операції підстановки байтів;
3. Процедура, зворотна процедурі перемішування стовпців;
4. Додавання раундового ключа

III. Завершальний раунд  $Nr$ , в якому пропускається перемішування стовпців



# 3. Алгоритм AES

## Зсув рядків в зворотному порядку

Байти в останніх трьох рядках матриці зсуваються циклічно **вліво** на різне число байт

$Nb$	Кількість зсувів			
	0-го рядка (-)	1-го рядка (C1)	2-го рядка (C2)	3-го рядка (C3)
4	0	1	2	3

# 3. Алгоритм AES

Обернена операція до операції підстановки байтів

Байти матриці замінюються новими значеннями за таблицею зворотної заміни, що є інвертованим S-боксом

	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



# 3. Алгоритм AES

## Процедура, зворотна процедурі перемішування стовпців

Стовпці стану розглядають як многочлен над полем  $GF(2^8)$  та множать за модулем  $x^4+1$  на фіксований многочлен  $c^{-1}(x)$ :

Якщо  $b(x)$  – стовпець до застосування до нього процедури, а  $a(x)$  – після, то перетворення можна записати так:  
 $a(x) = c(x) \otimes b(x)$ ,  
або у матричному вигляді:

$$c^{-1}(x) = 0b_{16} \cdot x^3 + 0d_{16} \cdot x^2 + 09_{16} \cdot x + 0e_{16}$$

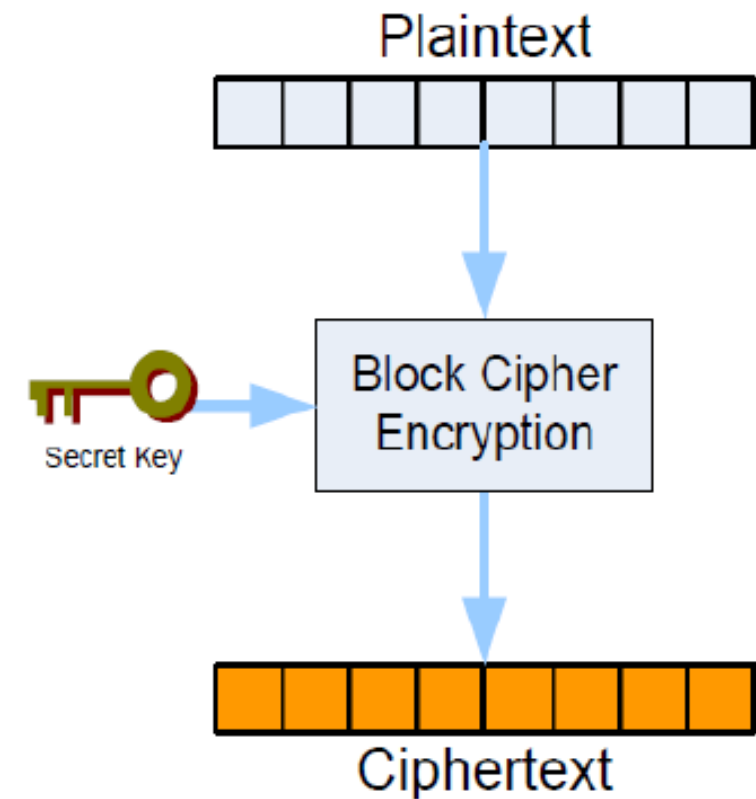
$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

# 3. Алгоритм AES

AES Visualization

# 4. Режими виконання блокових шифрів

**Режим шифрування** – метод застосування блочного шифру, що дозволяє перетворити послідовність блоків відкритих даних у послідовність блоків зашифрованих даних



# 4. Режими виконання блокових шифрів

## Режим простої заміни (ECB, Electronic Coding Book)

Кожен блок  $P_i$  шифрується **окремо** та **незалежно** від інших блоків алгоритмом  $E_k$  та ключем  $k$

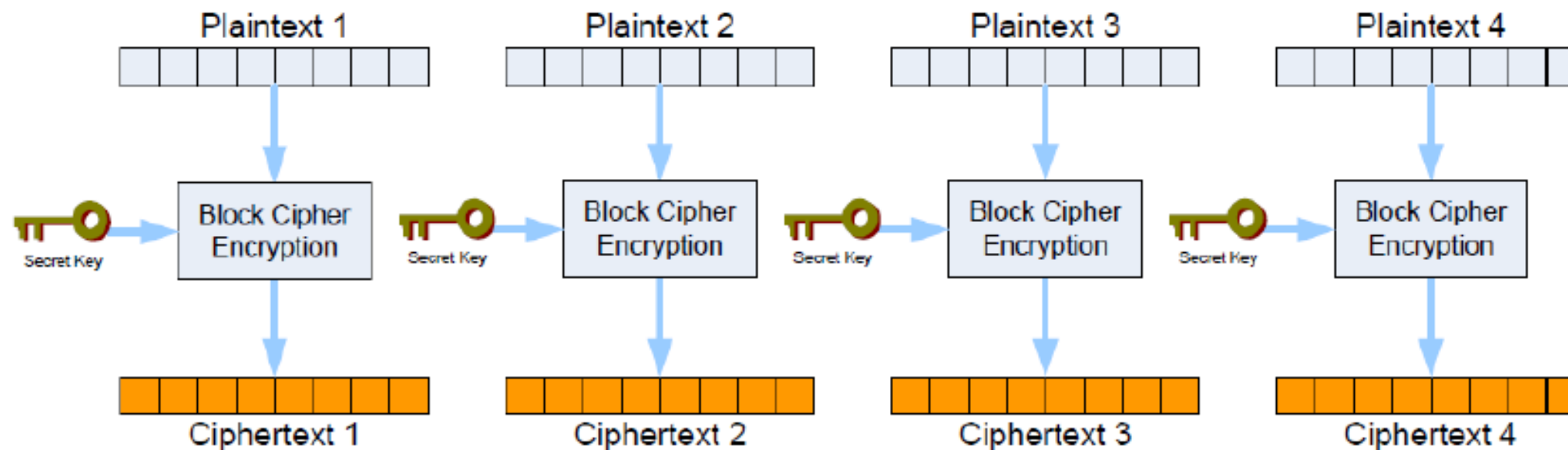
$$C_i = E_k(P_i)$$

$$P_i = D_k(C_i)$$

# 4. Режими виконання блокових шифрів

## Режим простої заміни (ECB, Electronic Coding Book)

- + Незалежне (паралельне) шифрування блоків
- При використанні одного ключа ідентичні блоки відкритого тексту шифруються в ідентичні блоки зашифрованого тексту
- Перестановка блоків зашифрованого тексту спричинює перестановку відповідних блоків відкритих текстів



# 4. Режим виконання блокових шифрів

## Режим зв'язування блоків (CBC, Cipher Block Chaining)

Кожен блок  $P_i$  додається за модулем 2 з **попередньо зашифрованим** блоком  $C_{i-1}$ , а потім результат передається на вхід функції  $E_k$ .

Для шифрування  $P_1$  використовують вектор ініціалізації  $IV$  (**Initialization Vector**) – послідовність випадкових символів розміром  $n$

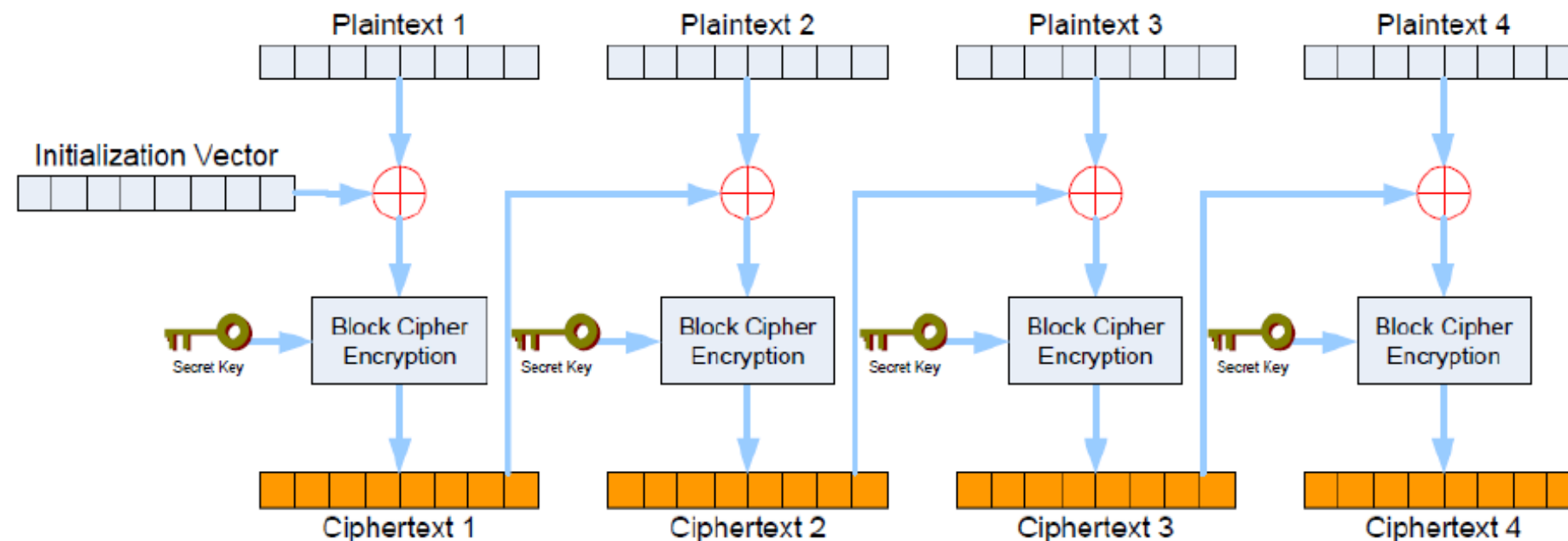
$$C_1 = E_k(P_1 \oplus IV)$$
$$C_i = E_k(P_i \oplus C_{i-1})$$

$$P_1 = D_k(C_1) \oplus IV$$
$$P_i = D_k(C_i) \oplus C_{i-1}$$

# 4. Режими виконання блокових шифрів

## Режим зв'язування блоків (CBC, Cipher Block Chaining)

- + Блоки з ідентичними початковими даними перетворюються у блоки із різними зашифрованими даними
- Помилка в одному блоці може поширюватися на інші блоки



# 4. Режими виконання блокових шифрів

## Режим зв'язування блоків (CBC, Cipher Block Chaining)

**Останній блок** шифротексту в CBC залежить від  $IV$ , ключів і всіх бітів відкритого тексту, тому він може використовуватися для автентифікації повідомлення та відправника і називається **кодом автентифікації повідомлення** або **MAC** (Message Authentication Code)





# 4. Режим виконання блокових шифрів

## Режим зі зворотнім зв'язком по шифротексту (CFB, Cipher Feedback)

CFB перетворює блоковий шифр у **потоковий**, що самосинхронізується.

Попередньо зашифрований блок  $C_{i-1}$  **шифрується** ще раз і додається за модулем 2 з  $P_i$

$$C_1 = P_1 \oplus E_k(IV)$$
$$C_i = P_i \oplus E_k(C_{i-1})$$

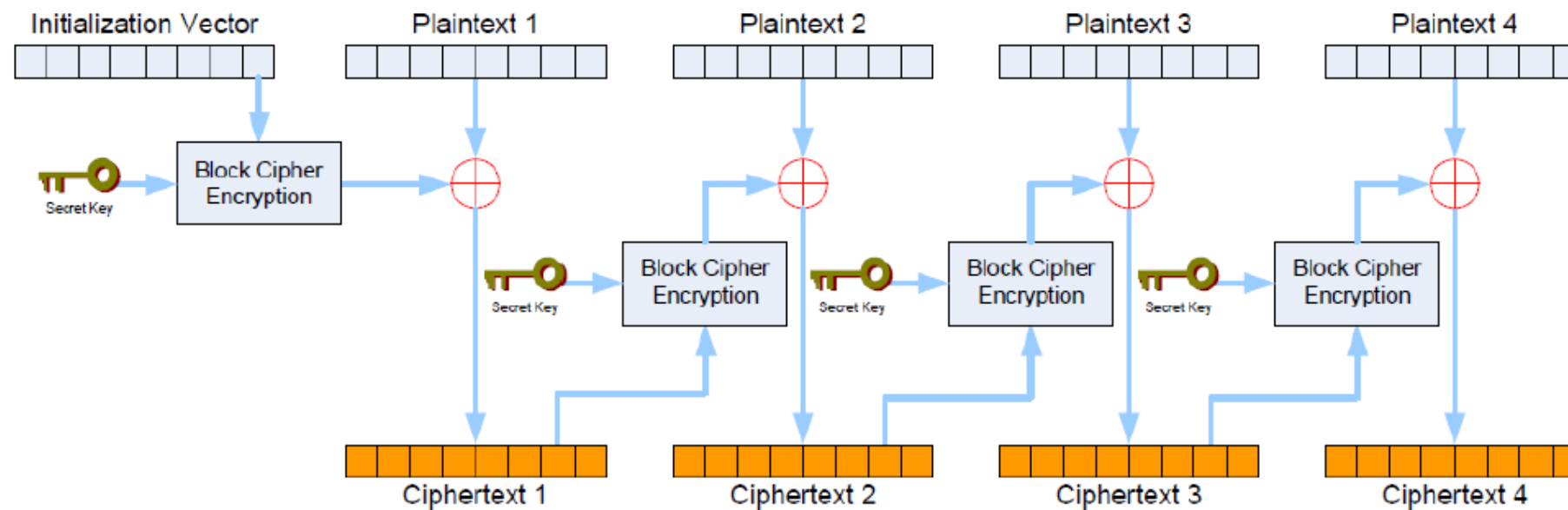
$$P_1 = C_1 \oplus D_k(IV)$$
$$P_i = C_i \oplus D_k(C_{i-1})$$

# 4. Режими виконання блокових шифрів

## Режим зі зворотнім зв'язком по шифротексту (CFB, Cipher Feedback)

+ Можливість шифрувати блоки довжиною **менше n** біт (не потрібне доповнення блоків)

+ – Помилка у відкритих даних впливає на всі подальші зашифровані дані, але **самоусувається** в ході розшифрування



# 4. Режими виконання блокових шифрів

## Режим зі зворотнім зв'язком по виходу (OFB, Output Feedback)

OFB перетворює блоковий шифр у синхронний **потоковий шифр**. Операції виконуються із підмножиною бітів  $O_{i-1}$ , що являє собою частину **попередньо зашифрованого** блоку  $C_{i-1}$ . Результат одразу передається на наступний крок та додається за модулем 2 з  $P_i$

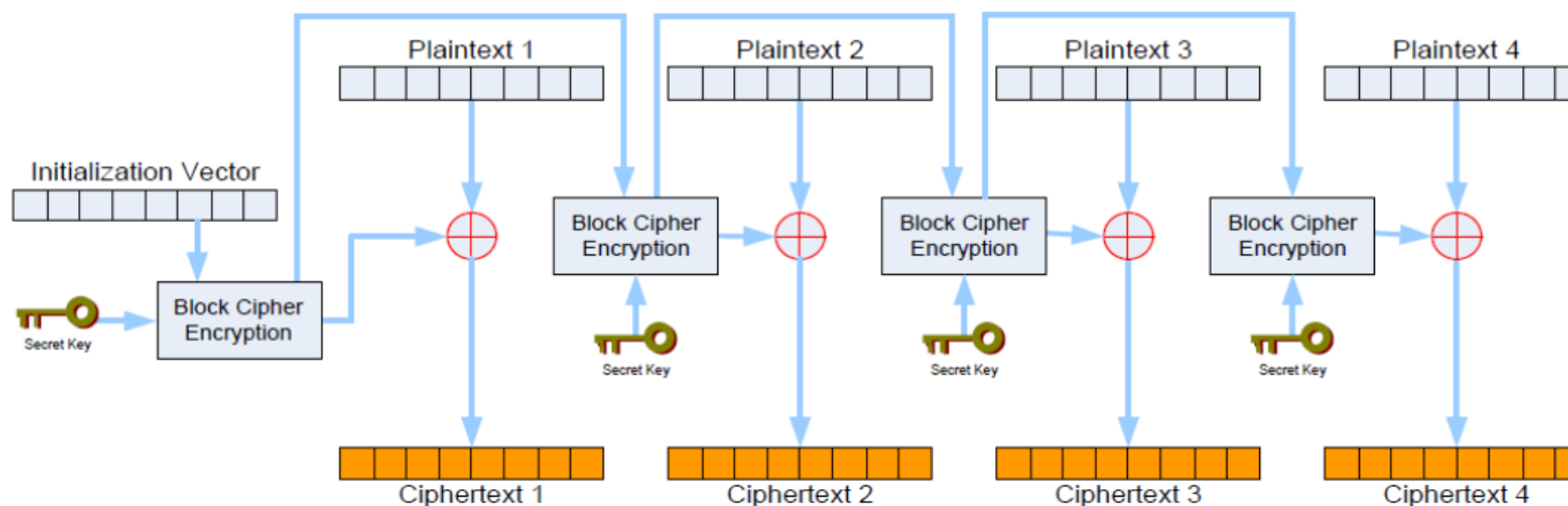
$$C_1 = P_1 \oplus E_k(IV)$$
$$C_i = P_i \oplus E_k(O_{i-1})$$

$$P_1 = C_1 \oplus D_k(IV)$$
$$P_i = C_i \oplus D_k(O_{i-1})$$

# 4. Режими виконання блокових шифрів

## Режим зі зворотнім зв'язком по виходу (OFB, Output Feedback)

- + Можливість шифрувати блоки довжиною **менше n** біт (не потрібне доповнення блоків)
- + Помилка у відкритих даних не впливає на всі подальші зашифровані дані
- Не дозволяє одночасно шифрувати декілька блоків



# 4. Режими виконання блокових шифрів

## Режим лічильника (CTR, Counter Mode)

Лічильник  $Ctr_{i-1}$  може бути будь-якими значеннями (послідовністю), що не повторюються. Найпоширенішими є прості лічильники, що на кожному кроці збільшуються на 1.

Лічильник зашифровується та додається за модулем 2 з  $P_i$

$$C_i = P_i \oplus E_k(Ctr_{i-1})$$

$$Ctr_{i-1} = IV | \text{Nonce}$$

$$P_i = C_i \oplus D_k(Ctr_{i-1})$$

# 4. Режими виконання блокових шифрів

## Режим лічильника (CTR, Counter Mode)

- + Блоки незалежні один від одного — вони залежать тільки від значень лічильника (не поширюється помилка)
- Потрібен синхронний лічильник для відправника та отримувача

