

Stefan Sokołowski

KRYPTOGRAFIA

wykład 3

Inst. Informatyki Stosowanej, ANS

Elbląg, 2026/2027

AES — *Advanced Encryption Standard*

AES — *Advanced Encryption Standard*

- AES to aktualny (od 2002) oficjalny standard szyfrowania w USA. Jest używany w korespondencji handlowej, ale nie tylko.

AES — *Advanced Encryption Standard*

- AES to aktualny (od 2002) oficjalny standard szyfrowania w USA. Jest używany w korespondencji handlowej, ale nie tylko.
- Blokowy szyfr symetryczny (tak jak DES).

AES — *Advanced Encryption Standard*

- AES to aktualny (od 2002) oficjalny standard szyfrowania w USA. Jest używany w korespondencji handlowej, ale nie tylko.
- Blokowy szyfr symetryczny (tak jak DES).
- Bloki są 128 bitowe (16 bajtów); a klucze mogą mieć różne długości:
 - 128 bitów czyli 16 bajtów,
 - 192 bity czyli 24 bajty,
 - 256 bitów czyli 32 bajty.

AES — *Advanced Encryption Standard*

- AES to aktualny (od 2002) oficjalny standard szyfrowania w USA. Jest używany w korespondencji handlowej, ale nie tylko.
- Blokowy szyfr symetryczny (tak jak DES).
- Bloki są 128 bitowe (16 bajtów); a klucze mogą mieć różne długości:
 - 128 bitów czyli 16 bajtów,
 - 192 bity czyli 24 bajty,
 - 256 bitów czyli 32 bajty.

W tym wykładzie to zawsze będzie 128 bitów (16 bajtów).

Ciąg dalszy na następnym slajdzie ...

AES — *Advanced Encryption Standard*

...ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**

AES — *Advanced Encryption Standard*

...ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**, każda z innym kluczem powstałym z głównego przez t.zw. **rozszerzanie klucza**.

AES — *Advanced Encryption Standard*

...ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**, każda z innym kluczem powstałym z głównego przez t.zw. **rozszerzanie klucza**.
- Na każdą rundę składa się:
 - etap *SubBytes* — zastępowanie bajtów innymi wg specjalnej tablicy,
 - etap *ShiftRows* — przestawienia bajtów w bloku,
 - etap *MixColumns* — inne przestawienia bajtów w bloku,
 - etap *AddRoundKey* — $\oplus$ -owanie bloku z kluczem rundy

AES — *Advanced Encryption Standard*

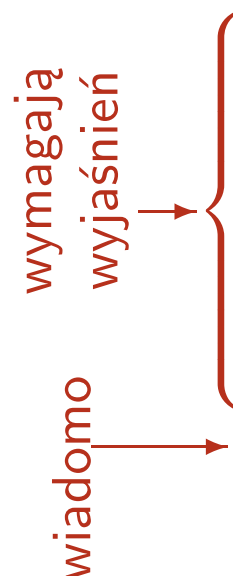
...ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**, każda z innym kluczem powstałym z głównego przez t.zw. **rozszerzanie klucza**.
- Na każdą rundę składa się:
 - etap *SubBytes* — zastępowanie bajtów innymi wg specjalnej tablicy,
 - etap *ShiftRows* — przestawienia bajtów w bloku,
 - etap *MixColumns* — inne przestawienia bajtów w bloku,
 - etap *AddRoundKey* — $\oplus$ -owanie bloku z kluczem rundy

AES — *Advanced Encryption Standard*

...ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**, każda z innym kluczem powstałym z głównego przez t.zw. **rozszerzanie klucza**.
- Na każdą rundę składa się:

- 
- etap *SubBytes* — zastępowanie bajtów innymi wg specjalnej tablicy,
 - etap *ShiftRows* — przestawienia bajtów w bloku,
 - etap *MixColumns* — inne przestawienia bajtów w bloku,
 - etap *AddRoundKey* — $\oplus$ -owanie bloku z kluczem rundy

AES — *Advanced Encryption Standard*

... ciąg dalszy z poprzedniego slajdu.

- Na szyfrowanie bloku składa się 10 **rund**, każda z innym kluczem powstałym z głównego przez t.zw. **rozszerzanie klucza**.
- Na każdą rundę składa się:
 - etap *SubBytes* — zastępowanie bajtów innymi wg specjalnej tablicy,
 - etap *ShiftRows* — przestawienia bajtów w bloku,
 - etap *MixColumns* — inne przestawienia bajtów w bloku,
 - etap *AddRoundKey* — $\oplus$ -owanie bloku z kluczem rundy
- prawdziwy AES ma jeszcze inne drobne komplikacje, których nie omawiam.

wymagają
wyjaśnień

wiadomo

Jak szyfrować?

Jak szyfrować?

Wstęp:

Ustawienie bloku 128 bajtów w tablicę o 4 kolumnach po 4 bajty

Jak szyfrować?

Dajcie mi punkt podparcia, a poruszę Ziemię
(Archimedes)

Wstęp:

Ustawienie bloku 128 bajtów w tablicę o 4 kolumnach po 4 bajty; np.

Jak szyfrować?

Dajcie mi punkt podparcia, a poruszę Ziemię
(Archimedes)

Wstęp:

Ustawienie bloku 128 bajtów w tablicę o 4 kolumnach po 4 bajty; np.

D	I	I	N
A	E	sp	K
J	sp	P	T
C	M	U	sp

Etap *ShiftRows*

Cykliczne przesuwanie i -tego wiersza w lewo o i

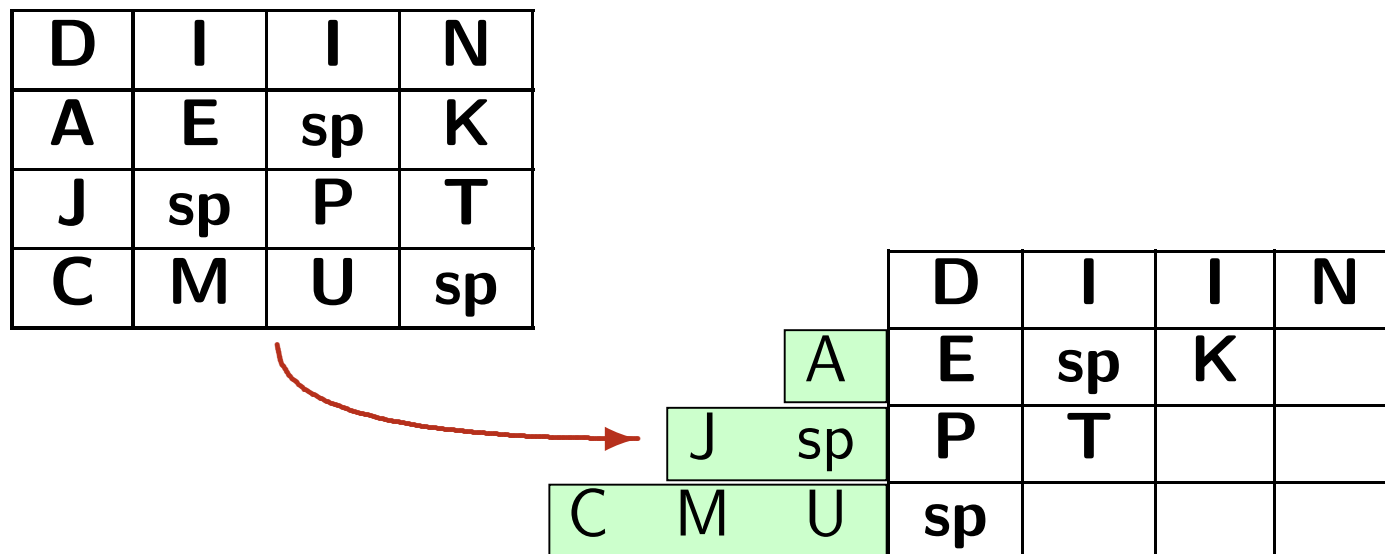
Etap *ShiftRows*

Cykliczne przesuwanie *i*-tego wiersza w lewo o *i*:

D	I	I	N
A	E	sp	K
J	sp	P	T
C	M	U	sp

Etap *ShiftRows*

Cykliczne przesuwanie i -tego wiersza w lewo o i :



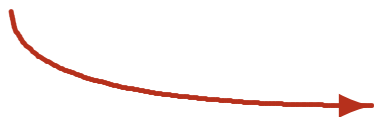
Etap *ShiftRows*

Cykliczne przesuwanie i -tego wiersza w lewo o i :

D	I	I	N
A	E	sp	K
J	sp	P	T
C	M	U	sp

			D	I	I	N
		A	E	sp	K	
	J	sp	P	T		
C	M	U	sp			

D	I	I	N
E	sp	K	A
P	T	J	sp
sp	C	M	U



Etap *ShiftRows*

Cykliczne przesuwanie i -tego wiersza w lewo o i :

D	I	I	N
A	E	sp	K
J	sp	P	T
C	M	U	sp

			D	I	I	N
		A	E	sp	K	
	J	sp	P	T		
C	M	U	sp			

D	I	I	N
E	sp	K	A
P	T	J	sp
sp	C	M	U

Litery przestają układać się w słowa.

Etap *ShiftRows*

Cykliczne przesuwanie i -tego wiersza w lewo o i :

D	I	I	N
A	E	sp	K
J	sp	P	T
C	M	U	sp

			D	I	I	N
		A	E	sp	K	
	J	sp	P	T		
C	M	U	sp			

D	I	I	N
E	sp	K	A
P	T	J	sp
sp	C	M	U

Litery przestają układać się w słowa.

Odszyfrowywanie:

Cykliczne przesunięcia w prawo.

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał.

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Np.

ABC

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Np.

ABC

$$= (10 \cdot 16 + 11) \cdot 16 + 12$$

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Np.

ABC

$$\begin{aligned}
 &= (10 \cdot 16 + 11) \cdot 16 + 12 \\
 &= 2748
 \end{aligned}$$

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**:

D	I	I	N
E	sp	K	A
P	T	J	sp
sp	C	M	U

=

44	49	49	4E
45	20	4B	41
50	54	4A	20
20	43	4D	55

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Np.

ABC

$$= (10 \cdot 16 + 11) \cdot 16 + 12$$

$$= 2748$$

Arytmetyka szesnastkowa

Poprzedni etap nie ingerował w kształt liter, tylko je przestawiał. Pozostałe etapy wykonują działania bitowe na liczbach reprezentujących litery; więc litery ulegają zniekształceniom.

Zamiana wg ASCII — o bitach wygodnie jest dyskutować w układzie **binarnym** lub **heksadecymalnym**:

D	I	I	N		44	49	49	4E
E	sp	K	A		45	20	4B	41
P	T	J	sp	=	50	54	4A	20
sp	C	M	U		20	43	4D	55

	01000100	01001001	01001001	01001110
=	01000101	00100000	01001011	01000001
	01010000	01010100	01001010	00100000
	00100000	01000011	01001101	01010101

DEC	HEX	BIN
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Np.

ABC

$$= (10 \cdot 16 + 11) \cdot 16 + 12$$

$$= 2748$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają)

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykle dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 0 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 001 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 0011 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 001111 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykle dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 0011110 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykle dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykle dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

szesnastkowe

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

szesnastkowe

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

szesnastkowe

- potrzebne jest jeszcze **mnożenie:** ●

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

szesnastkowe

- potrzebne jest jeszcze **mnożenie:** ●

przemienne: $a \bullet b = b \bullet a$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

szesnastkowe

- potrzebne jest jeszcze **mnożenie:** •

przemienne: $a \bullet b = b \bullet a$

łączne: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

szesnastkowe

- potrzebne jest jeszcze **mnożenie:** •

przemienne: $a \bullet b = b \bullet a$

łączne: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

z jedyneką: $a \bullet 01 = a$

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

- potrzebne jest jeszcze **mnożenie:** •

przemienne: $a \bullet b = b \bullet a$

łączne: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

z jedyneką: $a \bullet 01 = a$

z elementem odwrotnym: $a \neq 00 \Rightarrow \exists_b a \bullet b = 01$

szesnastkowe

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

- potrzebne jest jeszcze **mnożenie:** •

przemienne: $a \bullet b = b \bullet a$

łączne: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

z jedyką: $a \bullet 01 = a$

z elementem odwrotnym: $a \neq 00 \Rightarrow \exists_b a \bullet b = 01$

rozdzielne wzgl. dodawania: $a \bullet (b \oplus c) = (a \bullet b) \oplus (a \bullet c)$

szesnastkowe

Arytmetyka szesnastkowa

Potrzebujemy 2 działań na **bajtach**, czyli 8-bitowych ciągach zer i jedynek, nie wyprowadzających poza bajty (zwykłe dodawanie i mnożenie wyprowadzają); z „porządnymi” własnościami.

- **dodawanie:** $\oplus$ stosowane bit za bitem; np.

$$\begin{array}{r} 10110010 \\ \oplus 10001111 \\ \hline 00111101 \end{array}$$

przemienne: $a \oplus b = b \oplus a$

łączne: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

z zerem: $a \oplus 00 = a$

z elementem przeciwnym: $a \oplus a = 00$

- potrzebne jest jeszcze **mnożenie:** •

przemienne: $a \bullet b = b \bullet a$

łączne: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$

z jedyneką: $a \bullet 01 = a$

z elementem odwrotnym: $a \neq 00 \Rightarrow \exists_b a \bullet b = 01$

rozdzielne wzgl. dodawania: $a \bullet (b \oplus c) = (a \bullet b) \oplus (a \bullet c)$

— odwracalność działań przyda się przy odszyfrowywaniu.

szesnastkowe

Arytmetyka szesnastkowa — definicja mnożenia ●

Arytmetyka szesnastkowa — definicja mnożenia ●

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=}$$

Arytmetyka szesnastkowa — definicja mnożenia ●

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \overbrace{\bullet 2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2) = 41 \oplus ((7E \cdot 2) \bullet 2)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2) = 41 \oplus ((7E \cdot 2) \bullet 2)$
 $= 41 \oplus (FC \bullet 2)$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \overbrace{\bullet 2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2) = 41 \oplus ((7E \cdot 2) \bullet 2)$
 $= 41 \oplus (FC \bullet 2) = 41 \oplus (FC \cdot 2) \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2) = 41 \oplus ((7E \cdot 2) \bullet 2)$
 $= 41 \oplus (FC \bullet 2) = 41 \oplus (FC \cdot 2) \oplus 11B$
 $= 41 \oplus 1F8 \oplus 11B$

Arytmetyka szesnastkowa — definicja mnożenia •

- Mnożenie przez 2:

$$a \bullet 2 \stackrel{\text{def}}{=} \begin{cases} a \cdot 2 & \text{jeśli } a < 80 \\ (a \cdot 2) \oplus 11B & \text{jeśli } a \geq 80 \end{cases}$$

liczba o sprawdzonych
pożądanych własnościach

Przykład: $5A \bullet 2 = 5A \cdot 2 = B4$

Przykład: $B4 \bullet 2 = (B4 \cdot 2) \oplus 11B = 168 \oplus 11B = 73$

- Mnożenie przez potęgi 2: $a \bullet 2^n \stackrel{\text{def}}{=} a \bullet \overbrace{2 \bullet 2 \dots \bullet 2}^n$

Przykład: $7A \bullet 4 = 7A \bullet 2 \bullet 2 = F4 \bullet 2 = (F4 \cdot 2) \oplus 11B$
 $= 1E8 \oplus 11B = F3$

- Mnożenie przez dowolną liczbę — z rozdzielności:

Przykład: $3F \bullet B = 3F \bullet (1 \oplus 2 \oplus 8) = (3F \bullet 1) \oplus (3F \bullet 2) \oplus (3F \bullet 8)$
 $= 3F \oplus (3F \cdot 2) \oplus ((3F \cdot 2) \bullet 4)$
 $= 3F \oplus 7E \oplus ((7E \cdot 2) \bullet 2) = 41 \oplus ((7E \cdot 2) \bullet 2)$
 $= 41 \oplus (FC \bullet 2) = 41 \oplus (FC \cdot 2) \oplus 11B$
 $= 41 \oplus 1F8 \oplus 11B = A2$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus \\ \\ \\ \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{2} \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & \mathbf{3} & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ \mathbf{45} \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus \\ \\ \\ \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{2} \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ \mathbf{50} \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus \\ \\ \\ \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{3} & \mathbf{1} & \mathbf{1} & \mathbf{2} \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & \mathbf{1} \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ \mathbf{20} \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ \\ \\ \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (2 \bullet 45) \oplus (3 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (1 \bullet 45) \oplus (2 \bullet 50) \oplus (3 \bullet 20) \\ (3 \bullet 44) \oplus (1 \bullet 45) \oplus (1 \bullet 50) \oplus (2 \bullet 20) \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (2 \bullet 45) \oplus (3 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (1 \bullet 45) \oplus (2 \bullet 50) \oplus (3 \bullet 20) \\ (3 \bullet 44) \oplus (1 \bullet 45) \oplus (1 \bullet 50) \oplus (2 \bullet 20) \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (2 \bullet 45) \oplus (3 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (1 \bullet 45) \oplus (2 \bullet 50) \oplus (3 \bullet 20) \\ (3 \bullet 44) \oplus (1 \bullet 45) \oplus (1 \bullet 50) \oplus (2 \bullet 20) \end{bmatrix}$$

Odszyfrowywanie:

Mnożenie przez macierz odwrotną

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (2 \bullet 45) \oplus (3 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (1 \bullet 45) \oplus (2 \bullet 50) \oplus (3 \bullet 20) \\ (3 \bullet 44) \oplus (1 \bullet 45) \oplus (1 \bullet 50) \oplus (2 \bullet 20) \end{bmatrix}$$

Odszyfrowywanie:

Mnożenie przez macierz odwrotną:

$$\begin{bmatrix} E & B & D & 9 \\ 9 & E & B & D \\ D & 9 & E & B \\ B & D & 9 & E \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Etap *MixColumns*

Każdą kolumnę zastąpić jej iloczynem przez macierz

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Przykład:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 44 \\ 45 \\ 50 \\ 20 \end{bmatrix} = \begin{bmatrix} (2 \bullet 44) \oplus (3 \bullet 45) \oplus (1 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (2 \bullet 45) \oplus (3 \bullet 50) \oplus (1 \bullet 20) \\ (1 \bullet 44) \oplus (1 \bullet 45) \oplus (2 \bullet 50) \oplus (3 \bullet 20) \\ (3 \bullet 44) \oplus (1 \bullet 45) \oplus (1 \bullet 50) \oplus (2 \bullet 20) \end{bmatrix}$$

Odszyfrowywanie:

na odpowiedzialność Wikipedii:

https://en.wikipedia.org/wiki/Rijndael_MixColumns

Mnożenie przez macierz odwrotną:

$$\begin{bmatrix} E & B & D & 9 \\ 9 & E & B & D \\ D & 9 & E & B \\ B & D & 9 & E \end{bmatrix} \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Etap *SubBytes*

Zastąpienie każdego bajtu innym w sposób odwracalny według tabeli.

Etap *SubBytes* — szyfrowanie (*S-Box substitution*)

	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

Etap *SubBytes* — szyfrowanie (*S-Box substitution*)

	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

Przykład: C3 $\longrightarrow$ 2E

Etap *SubBytes* — odszyfrowywanie (*inverse S-Box substitution*)

	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

Etap *SubBytes* — odszyfrowywanie (*inverse S-Box substitution*)

	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

Przykład: 2E → C3

AES — „rozmnożenie” klucza (*key expansion*)

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
 - Jak z jednego klucza zrobić kilkanaście?
-

Oryginalny klucz 128-bitowy (16 bajtów):

k^0

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

Oryginalny klucz 128-bitowy (16 bajtów):

Liczba to istota rzeczy
(Pitagoras)

k^0

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

Oryginalny klucz 128-bitowy (16 bajtów):

$k^0 =$

L	B	O	T
I	A		O
C		I	T
Z	T	S	A

Liczba to istota rzeczy
(Pitagoras)

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

Oryginalny klucz 128-bitowy (16 bajtów):

Liczba to istota rzeczy
(Pitagoras)

$k^0 =$

L	B	O	T
I	A		O
C		I	T
Z	T	S	A

ASCII
=
HEX

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

Oryginalny klucz 128-bitowy (16 bajtów):

Liczba to istota rzeczy
(Pitagoras)

k^0	=	L	B	O	T	ASCII = HEX	4C	42	4F	54
		I	A		O		49	41	20	47
		C		I	T		43	20	49	54
		Z	T	S	A		5A	54	53	41

Tego klucza nie stosujemy bezpośrednio, tylko tworzymy z niego klucze do kolejnych 10 rund:

$$k^0 \rightarrow k^1 \rightarrow k^2 \rightarrow \dots \rightarrow k^{10}$$

AES — „rozmnożenie” klucza (*key expansion*)

- Potrzebujemy kilkunastu rund szyfrowania; każdą z innym kluczem.
- Jak z jednego klucza zrobić kilkanaście?

Oryginalny klucz 128-bitowy (16 bajtów):

Liczba to istota rzeczy
(Pitagoras)

$k^0 =$

L	B	O	T
I	A		O
C		I	T
Z	T	S	A

ASCII
=
HEX

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

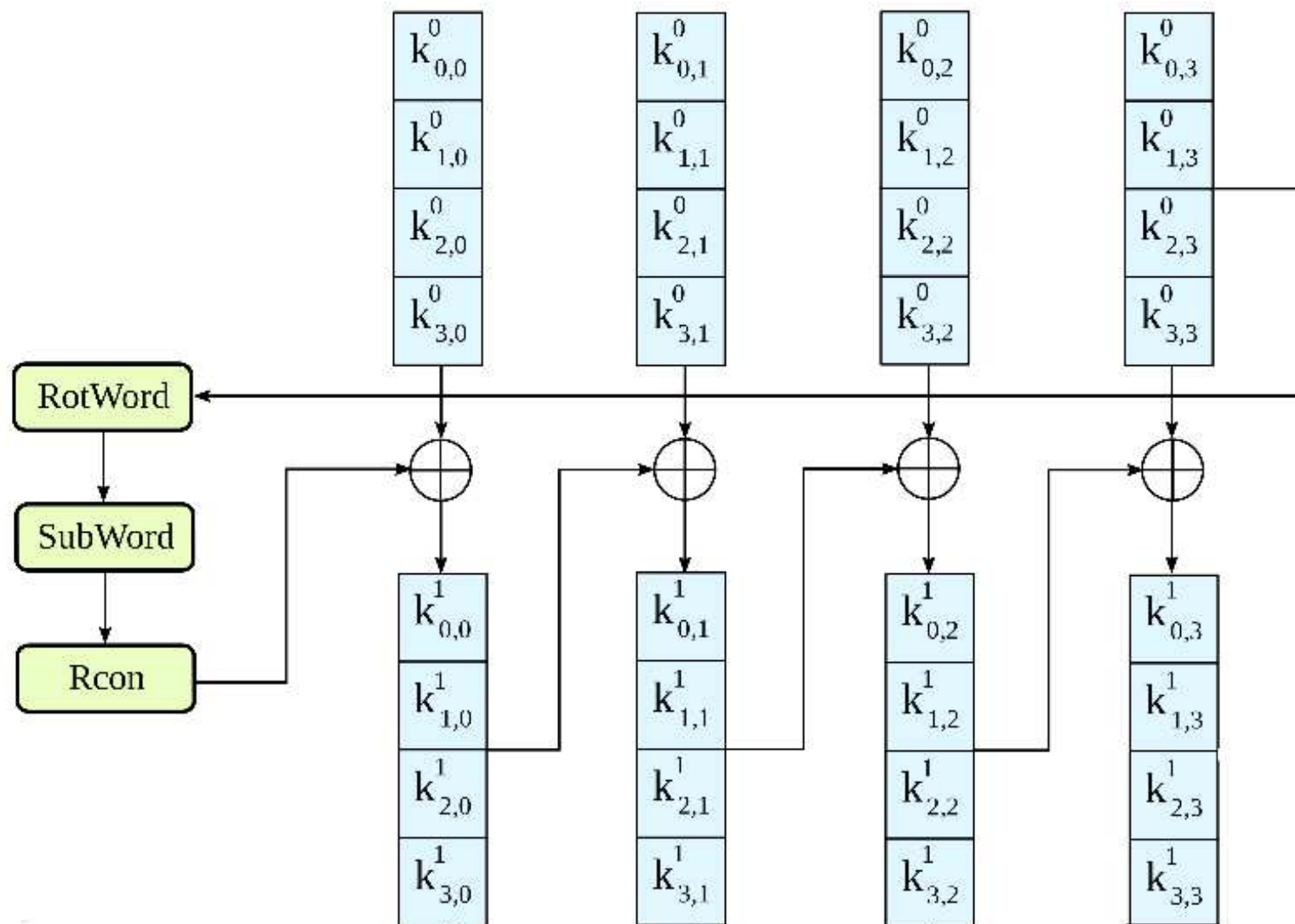
Tego klucza nie stosujemy bezpośrednio, tylko tworzymy z niego klucze do kolejnych 10 rund:

$$k^0 \rightarrow k^1 \rightarrow k^2 \rightarrow \dots \rightarrow k^{10}$$

Jak z klucza k^{i-1} zrobić klucz k^i ?

AES — „rozmnożenie” klucza (*key expansion*)

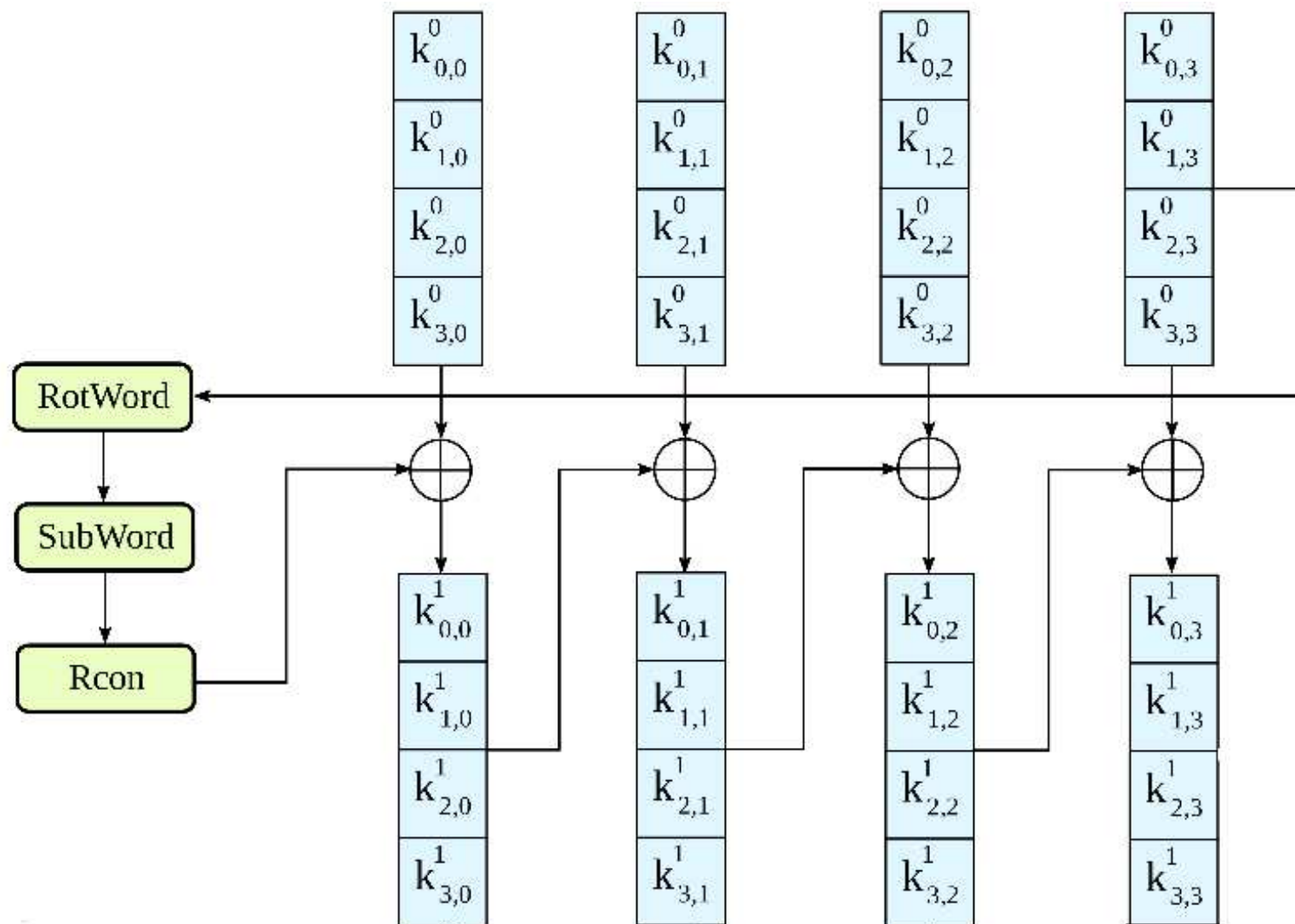
Przekształcać kolumny klucza wg schematu:



(rysunek z https://en.wikipedia.org/wiki/AES_key_schedule)

AES — „rozmnożenie” klucza (*key expansion*)

Przekształcać kolumny klucza wg schematu:



*RotWord, SubWord, Rcon
wymagają ogólny schemat
wyjaśnień*

(rysunek z https://en.wikipedia.org/wiki/AES_key_schedule)

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

→

54
47
54
41

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41



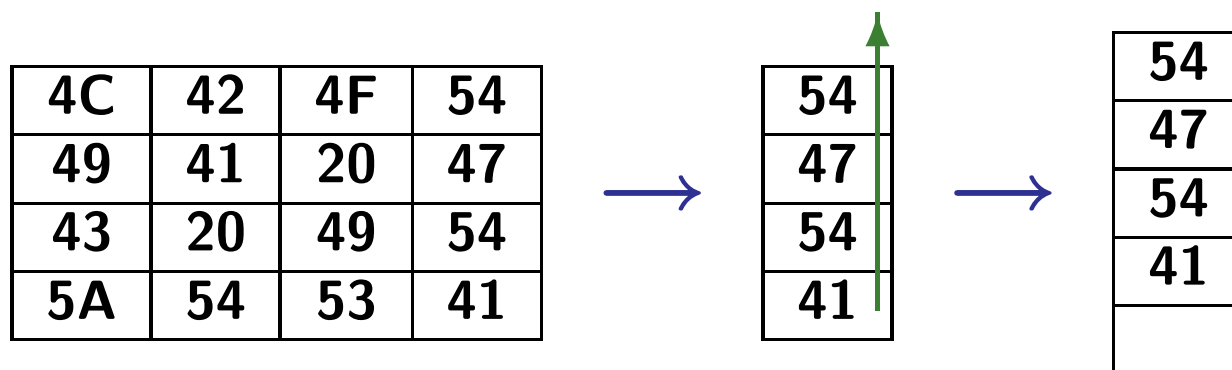
54
47
54
41



AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

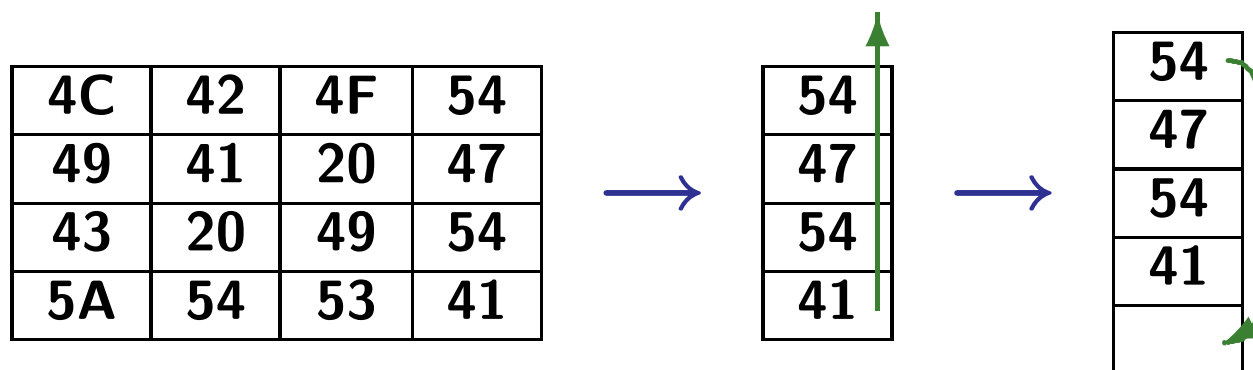
RotWord — cykliczne przesunięcie ostatniej kolumny:



AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

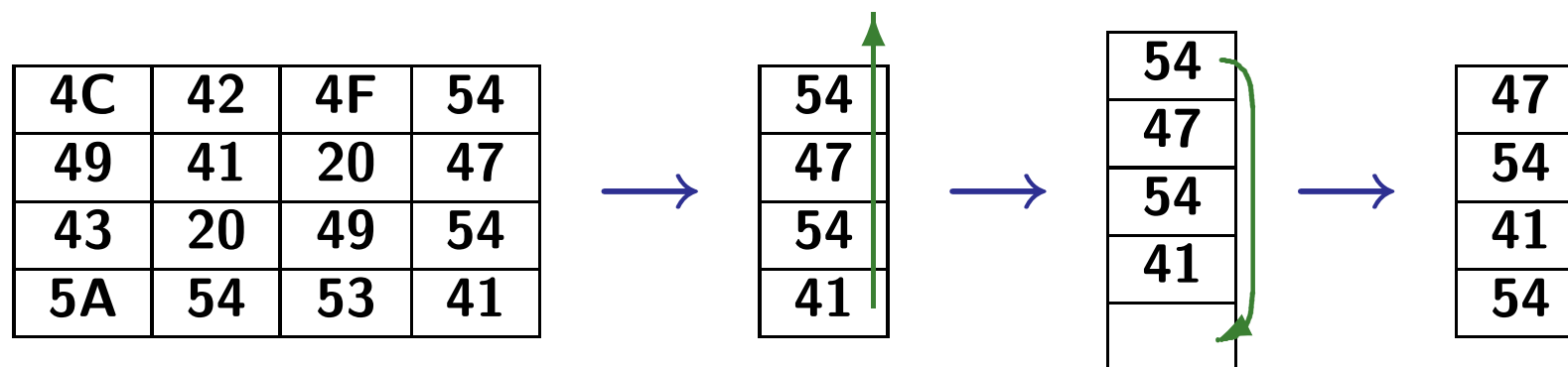
RotWord — cykliczne przesunięcie ostatniej kolumny:



AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

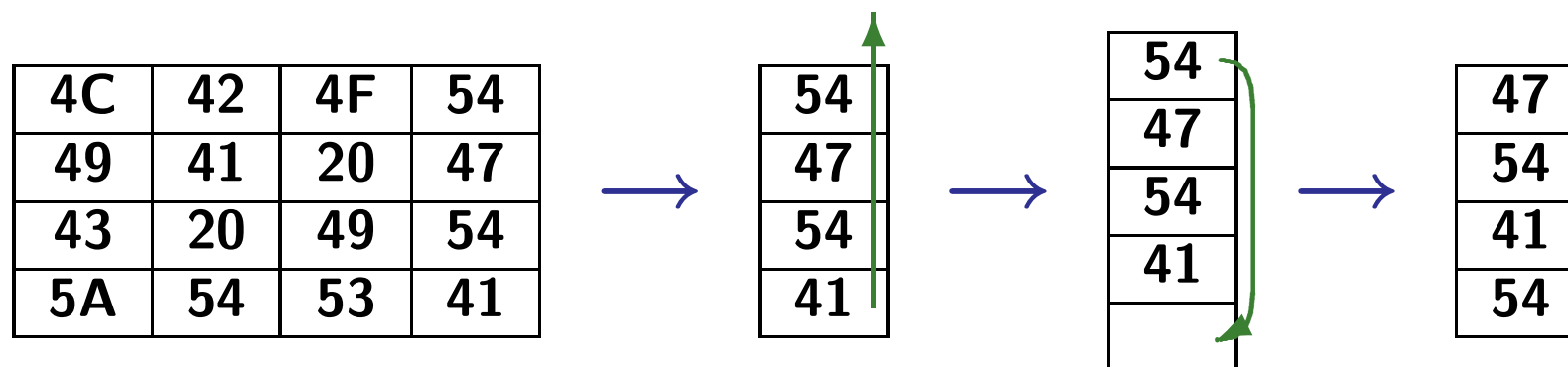
RotWord — cykliczne przesunięcie ostatniej kolumny:



AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:

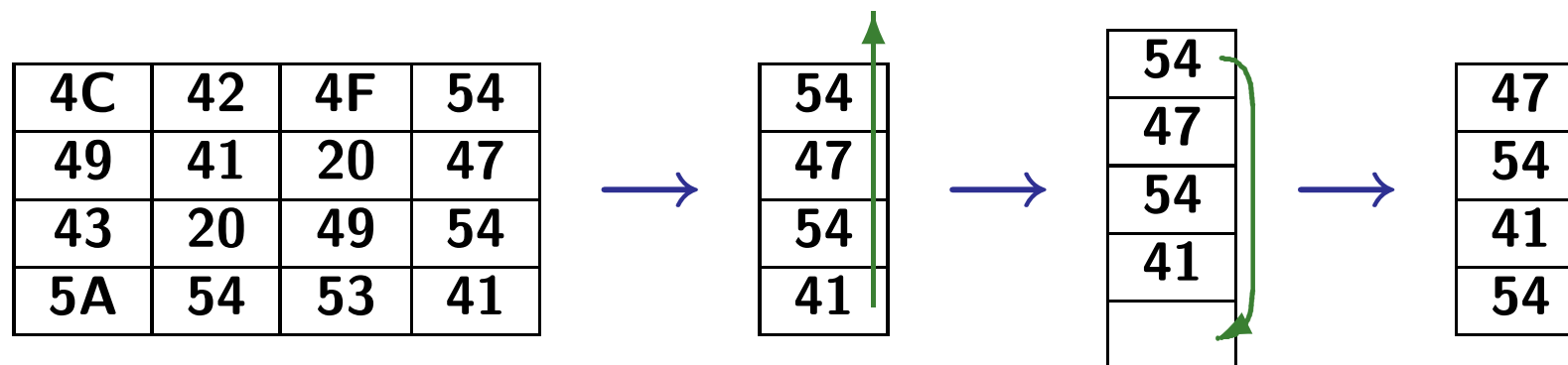


SubWord — *S-Box subst.* na ostatniej kolumnie (por. str. 10)

AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:



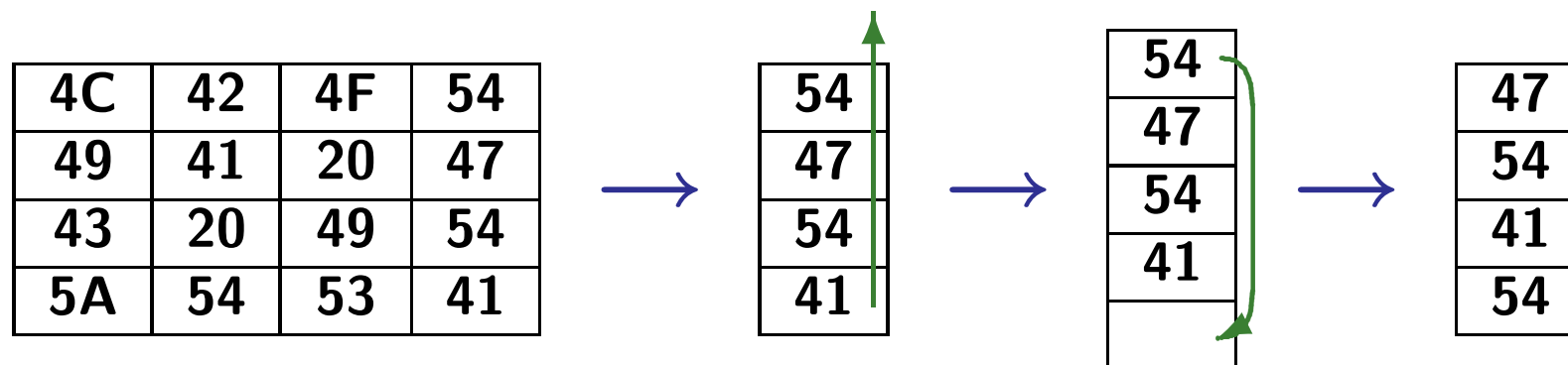
SubWord — *S-Box subst.* na ostatniej kolumnie (por. str. 10):

47
54
41
54

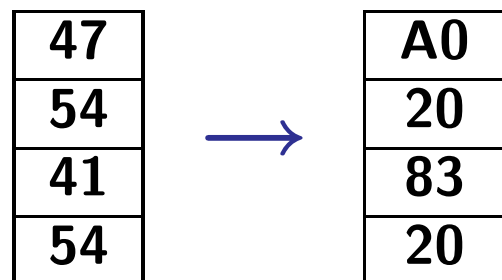
AES — „rozmnożenie” klucza (*key expansion*)

Na razie działamy tylko na ostatniej kolumnie klucza...

RotWord — cykliczne przesunięcie ostatniej kolumny:



SubWord — *S-Box subst.* na ostatniej kolumnie (por. str. 10):



AES — „rozmnożenie” klucza (*key expansion*)

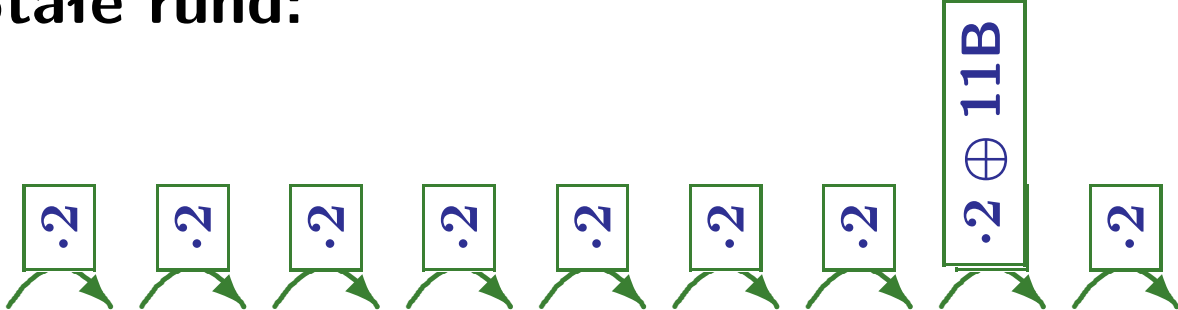
AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

1									
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:



	1	2	4	8	10	20	40	80	1B	36
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
runda	1	2	3	4	5	6	7	8	9	10

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
	1	2	4	8	10	20	40	80	1B
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
runda 1									
runda 2									
runda 3									
runda 4									
runda 5									
runda 6									
runda 7									
runda 8									
runda 9									
runda 10									

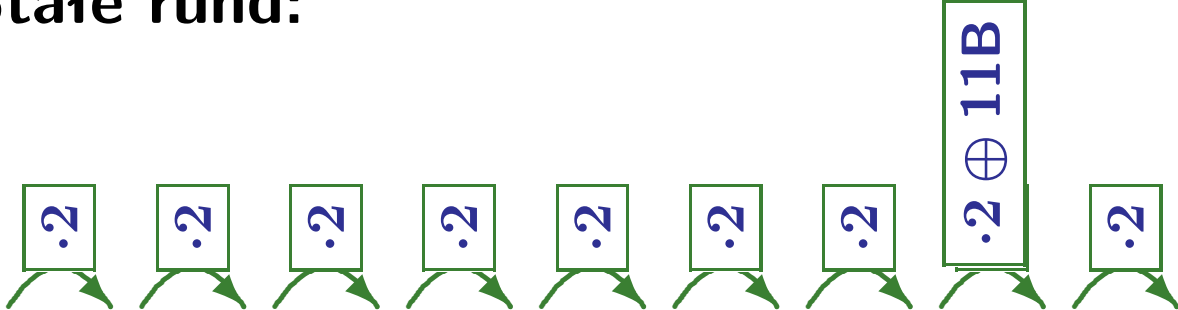
$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:



	1	2	4	8	10	20	40	80	1B	36
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
runda 1										
runda 2										
runda 3										
runda 4										
runda 5										
runda 6										
runda 7										
runda 8										
runda 9										
runda 10										

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10 \quad \leftarrow !$$

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10$$

$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$



AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10$$

$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$

$$80 \cdot 2 \oplus 11B$$



AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10$$

$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$

$$80 \cdot 2 \oplus 11B = 100 \oplus 11B$$

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10$$



$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$

$$80 \cdot 2 \oplus 11B = 100 \oplus 11B = 1B$$

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10$$



$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$

$$80 \cdot 2 \oplus 11B = 100 \oplus 11B = 1B$$

$$1B \cdot 2 = 36$$

AES — „rozmnożenie” klucza (*key expansion*)

Stałe rund:

	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2$	$\cdot 2 \oplus 11B$	$\cdot 2$	
1	2	4	8	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
runda 1	runda 2	runda 3	runda 4	runda 5	runda 6	runda 7	runda 8	runda 9	runda 10

$$1 \cdot 2 = 2$$

$$2 \cdot 2 = 4$$

$$4 \cdot 2 = 8$$

$$8 \cdot 2 = 10 \quad \leftarrow !$$

$$10 \cdot 2 = 20$$

$$20 \cdot 2 = 40$$

$$40 \cdot 2 = 80$$

$$80 \cdot 2 \oplus 11B = 100 \oplus 11B$$

$$= 1B$$

$$1B \cdot 2 = 36$$

W rundzie nr i używamy 4-bajtowej stałej rundy i

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0	01
20	00
83	00
20	00

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0
20
83
20

 $\oplus$

01
00
00
00

 $=$

A1
20
83
20

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0
20
83
20

 $\oplus$

01
00
00
00

 $=$

A1
20
83
20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0	01	A1
20	00	20
83	00	83
20	00	20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...	4C	42	4F	54
20	...	49	41	20	47
83	...	43	20	49	54
20	...	5A	54	53	41

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0	01	A1
20	00	20
83	00	83
20	00	20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...	4C	42	4F	54	EB	42	4F	54
20	...	49	41	20	47	69	41	20	47
83	...	43	20	49	54	C0	20	49	54
20	...	5A	54	53	41	7A	54	53	41

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0
20
83
20

 $\oplus$

01
00
00
00

 $=$

A1
20
83
20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...
20	...
83	...
20	...

 $\oplus$

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

 $=$

EB	42	4F	54
69	41	20	47
C0	20	49	54
7A	54	53	41

XOR-owanie pierwszej kolumny z drugą kolumną klucza

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0
20
83
20

 $\oplus$

01
00
00
00

 $=$

A1
20
83
20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...
20	...
83	...
20	...

 $\oplus$

4C	42	4F	54
49	41	20	47
43	20	49	54
5A	54	53	41

 $=$

EB	42	4F	54
69	41	20	47
C0	20	49	54
7A	54	53	41

XOR-owanie pierwszej kolumny z drugą kolumną klucza:

EB	42	4F	54
69	41	20	47
C0	20	49	54
7A	54	53	41

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0	01	A1
20	00	20
83	00	83
20	00	20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...	4C	42	4F	54	EB	42	4F	54
20	...	49	41	20	47	69	41	20	47
83	...	43	20	49	54	C0	20	49	54
20	...	5A	54	53	41	7A	54	53	41

XOR-owanie pierwszej kolumny z drugą kolumną klucza:

EB	42	4F	54	EB	A9	4F	54
69	41	20	47	69	28	20	47
C0	20	49	54	C0	E0	49	54
7A	54	53	41	7A	2E	53	41

AES — „rozmnożenie” klucza (*key expansion*)

Rcon — XOR-owanie przekształconej ostatniej kolumny ze stałą rundy:

A0	01	A1
20	00	20
83	00	83
20	00	20

XOR-owanie tak otrzymanej kolumny z pierwszą kolumną klucza:

A1	...	4C	42	4F	54	EB	42	4F	54
20	...	49	41	20	47	69	41	20	47
83	...	43	20	49	54	C0	20	49	54
20	...	5A	54	53	41	7A	54	53	41

XOR-owanie pierwszej kolumny z drugą kolumną klucza:

EB	42	4F	54	EB	A9	4F	54
69	41	20	47	69	28	20	47
C0	20	49	54	C0	E0	49	54
7A	54	53	41	7A	2E	53	41

dalej →

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41



EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41

→

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

XOR-owanie trzeciej kolumny z czwartą kolumną klucza

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41

→

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

XOR-owanie trzeciej kolumny z czwartą kolumną klucza:

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41

→

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

XOR-owanie trzeciej kolumny z czwartą kolumną klucza:

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

→

EB	A9	E6	B2
69	28	08	4F
C0	E0	A9	FB
7A	2E	7B	3A

AES — „rozmnożenie” klucza (*key expansion*)

XOR-owanie drugiej kolumny z trzecią kolumną klucza:

EB	A9	4F	54
69	28	20	47
C0	E0	49	54
7A	2E	53	41

→

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

XOR-owanie trzeciej kolumny z czwartą kolumną klucza:

EB	A9	E6	54
69	28	08	47
C0	E0	A9	54
7A	2E	7B	41

→

EB	A9	E6	B2
69	28	08	4F
C0	E0	A9	FB
7A	2E	7B	3A

gotowy klucz
dla pierwszej rundy