Kryptografia

z elementami kryptografii kwantowej

Ryszard Tanaś

http://zon8.physd.amu.edu.pl/~tanas

Wykład 4

Spis treści

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7 IDEA — International Data Encryption Algorithm

- IDEA jest algorytmem blokowym wprowadzonym w latach 90-tych
- IDEA używa kluczy 128 bitowych
- IDEA jest używana w pakiecie PGP
- IDEA jest algorytmem opatentowanym; można go używać bezpłatnie do celów niekomercyjnych
- IDEA działa na blokach 64 bitowych i wykorzystuje 3 różne operacje: xor ($\oplus$), dodawanie modulo 2^{16} ($\boxplus$) oraz mnożenie modulo $2^{16}+1$ ($\odot$)

7.1 Szyfrowanie

- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

7.1 Szyfrowanie

- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

7.1 Szyfrowanie

- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

7.1 Szyfrowanie

- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

7.1 Szyfrowanie

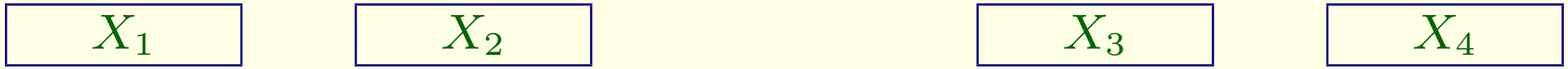
- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

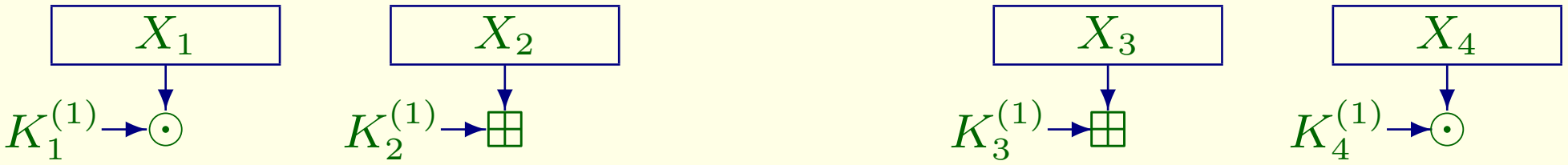
7.1 Szyfrowanie

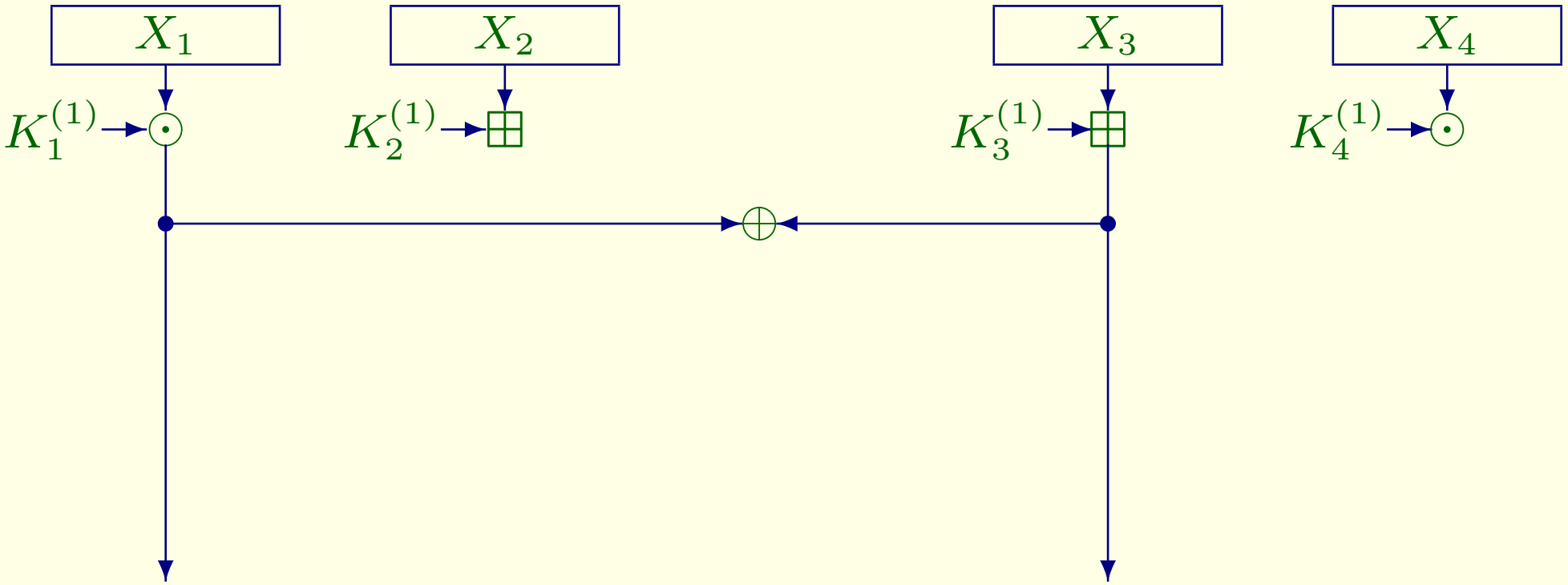
- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

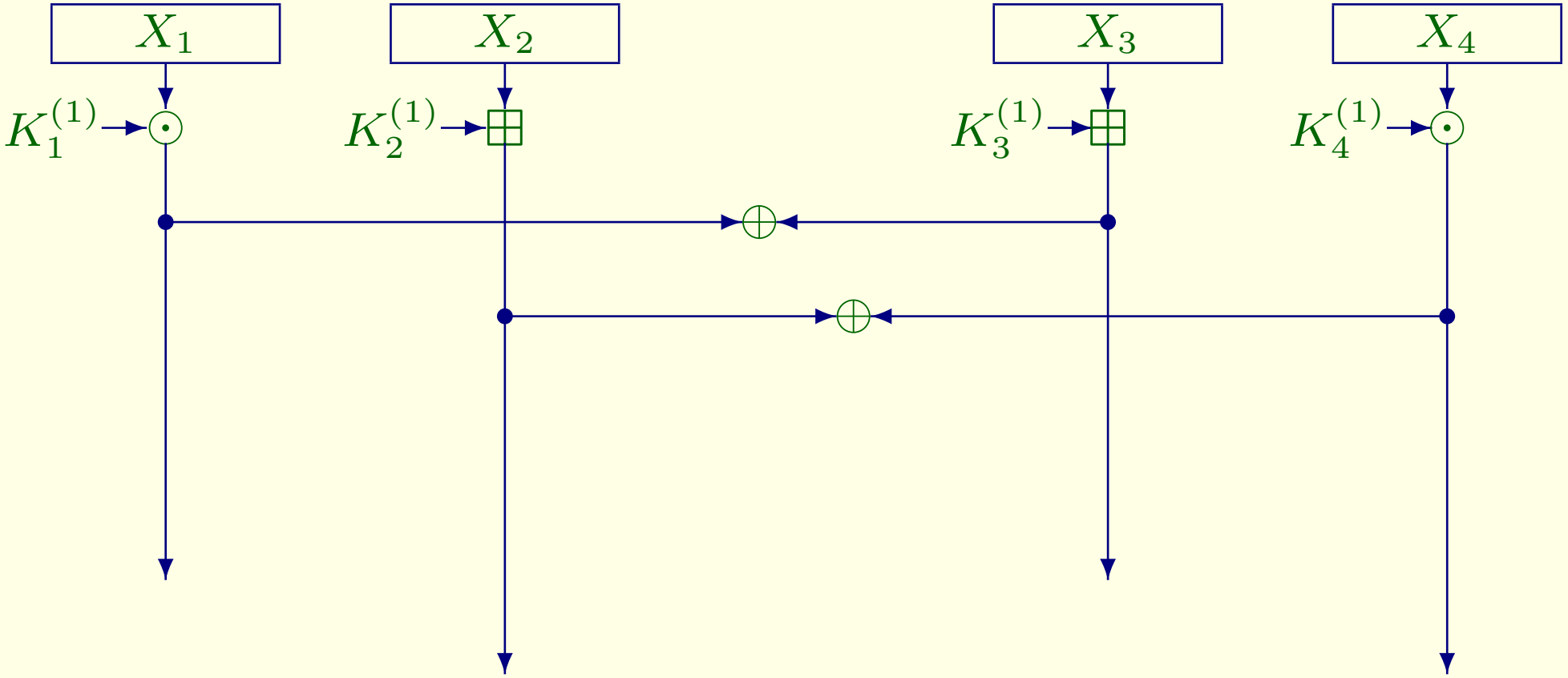
7.1 Szyfrowanie

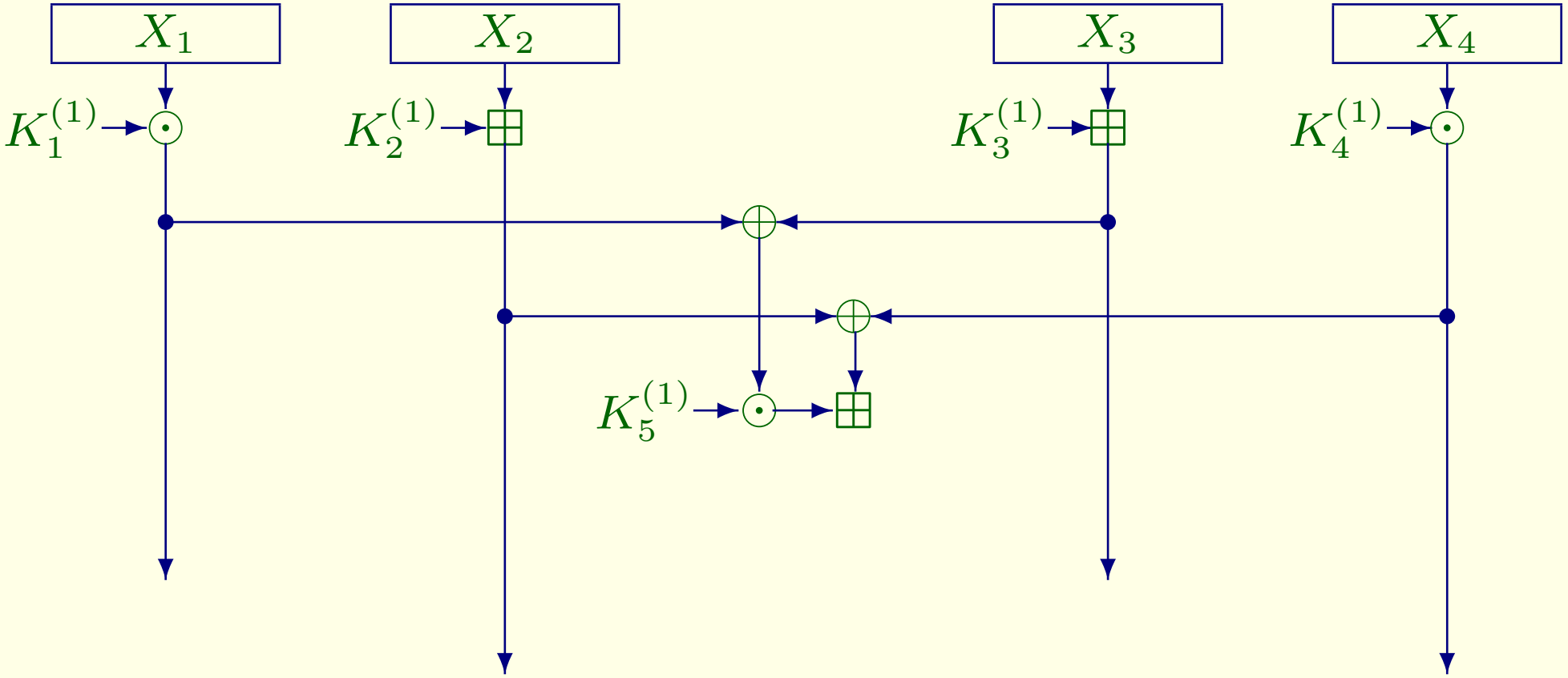
- 64 bitowy blok tekstu jawnego jest dzielony na 4 bloki po 16 bitów: X_1, X_2, X_3, X_4
- Algorytm składa się z 8 rund
- W każdej rundzie wykonywane są wymienione wyżej 3 typy operacji na 16 bitowych blokach z 16 bitowymi podkluczami (każda runda wymaga 6 podkluczy)
- Pomiędzy rundami blok 2 i 3 są zamieniane
- Algorytm kończy przekształcenie końcowe, które wymaga 4 podkluczy
- W wyniku otrzymuje się 4 bloki kryptogramu po 16 bitów: Y_1, Y_2, Y_3, Y_4

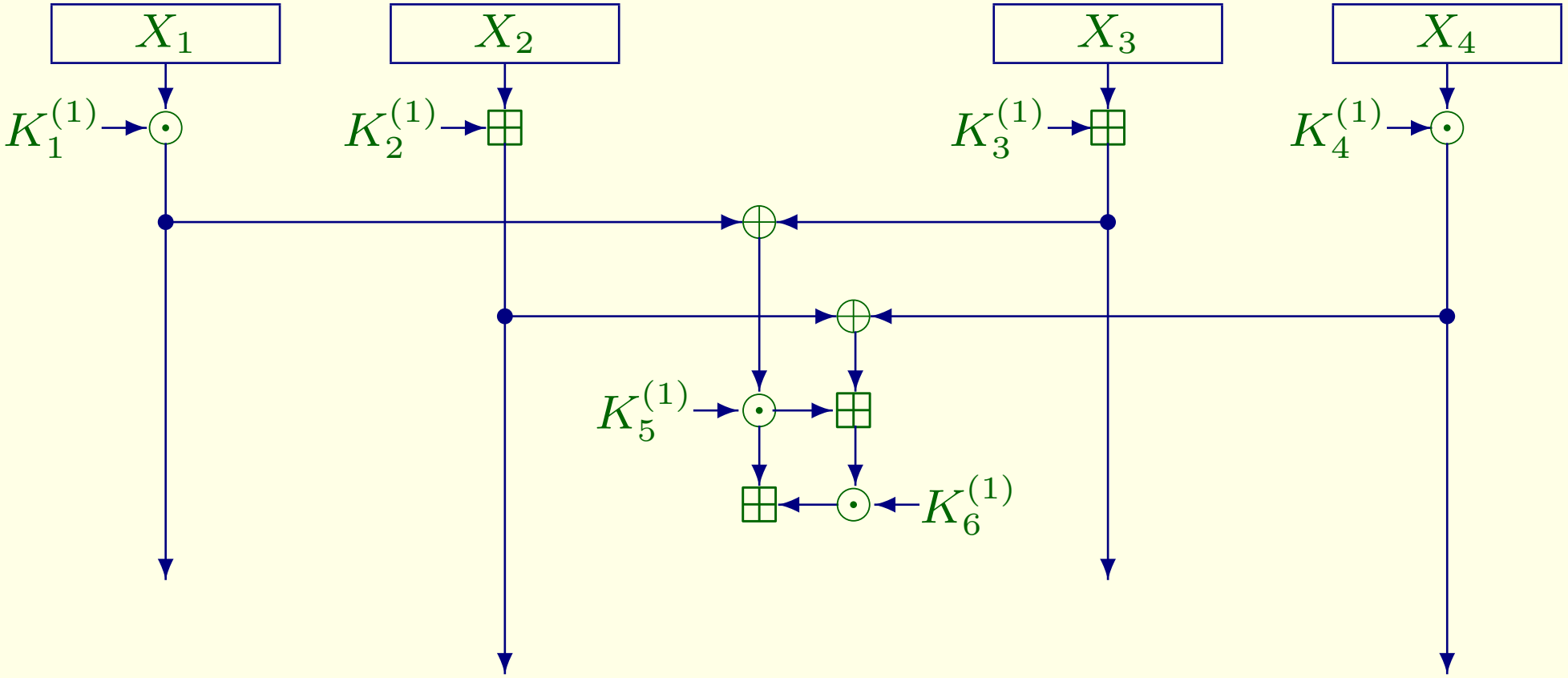

X_1
X_2
X_3
X_4

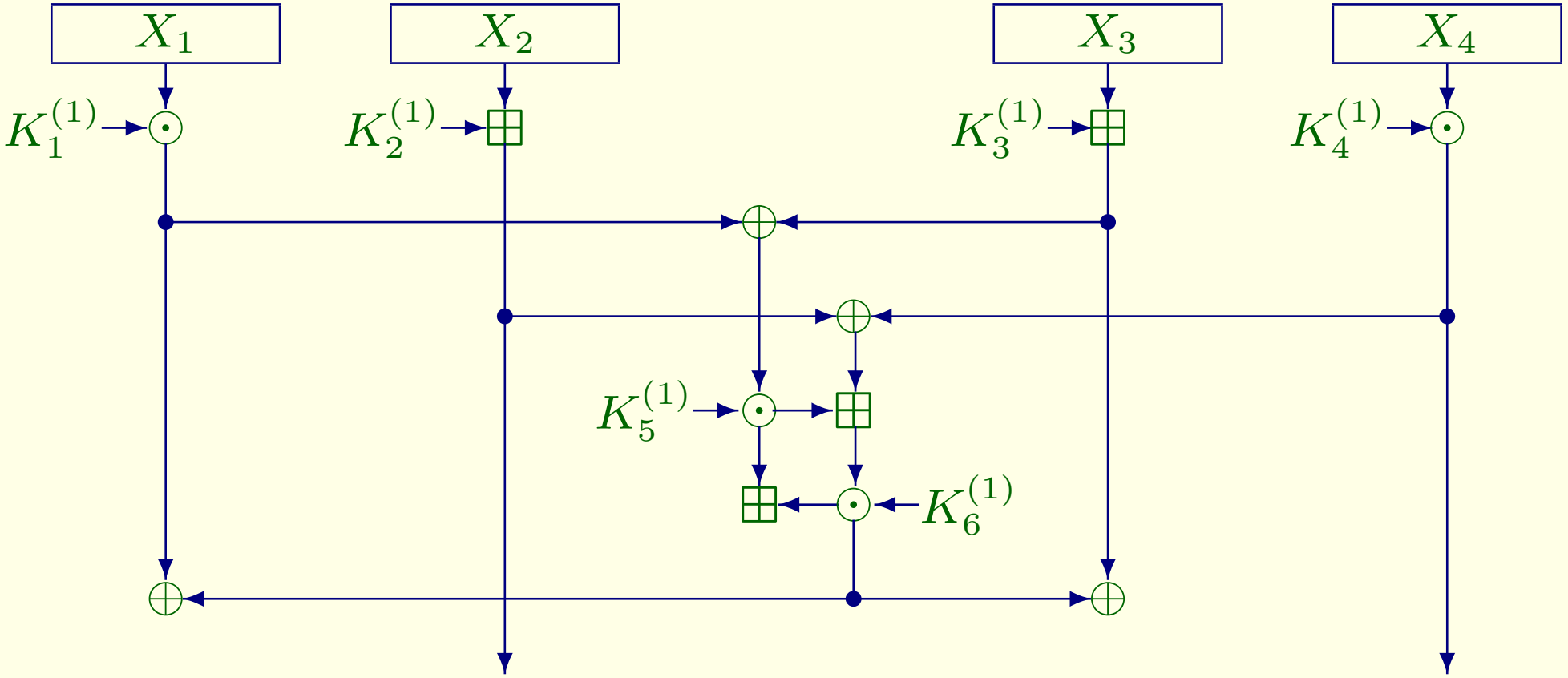

X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$

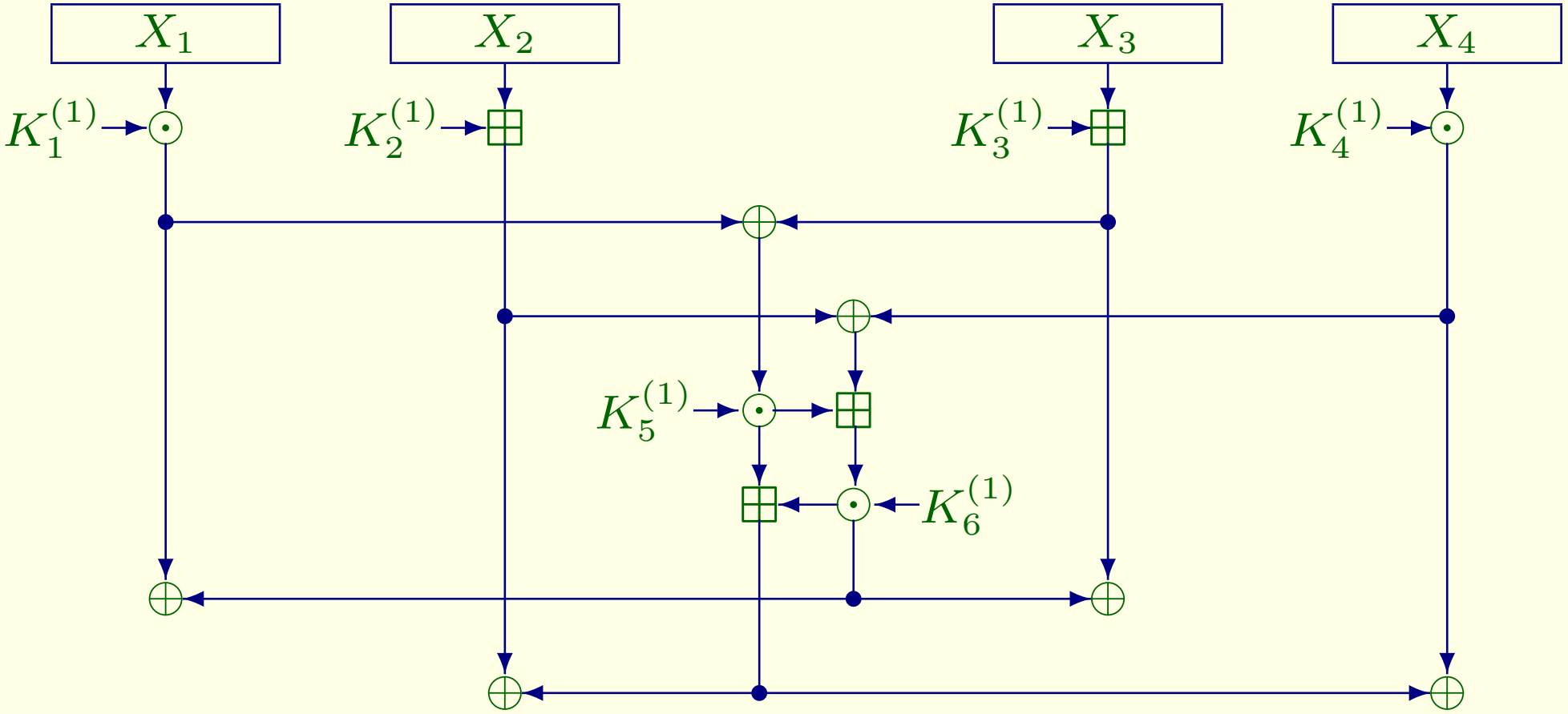

X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$

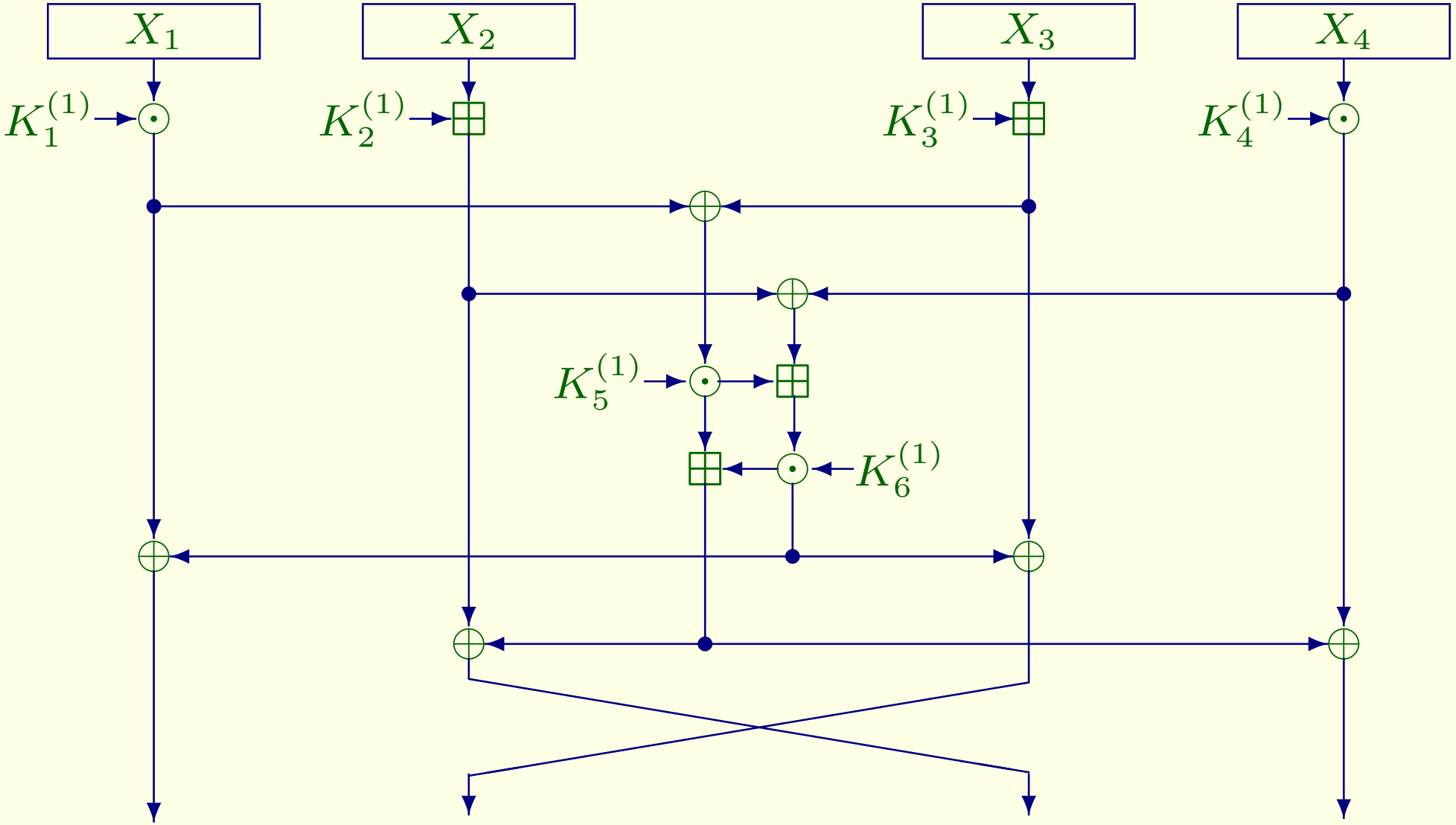

X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$

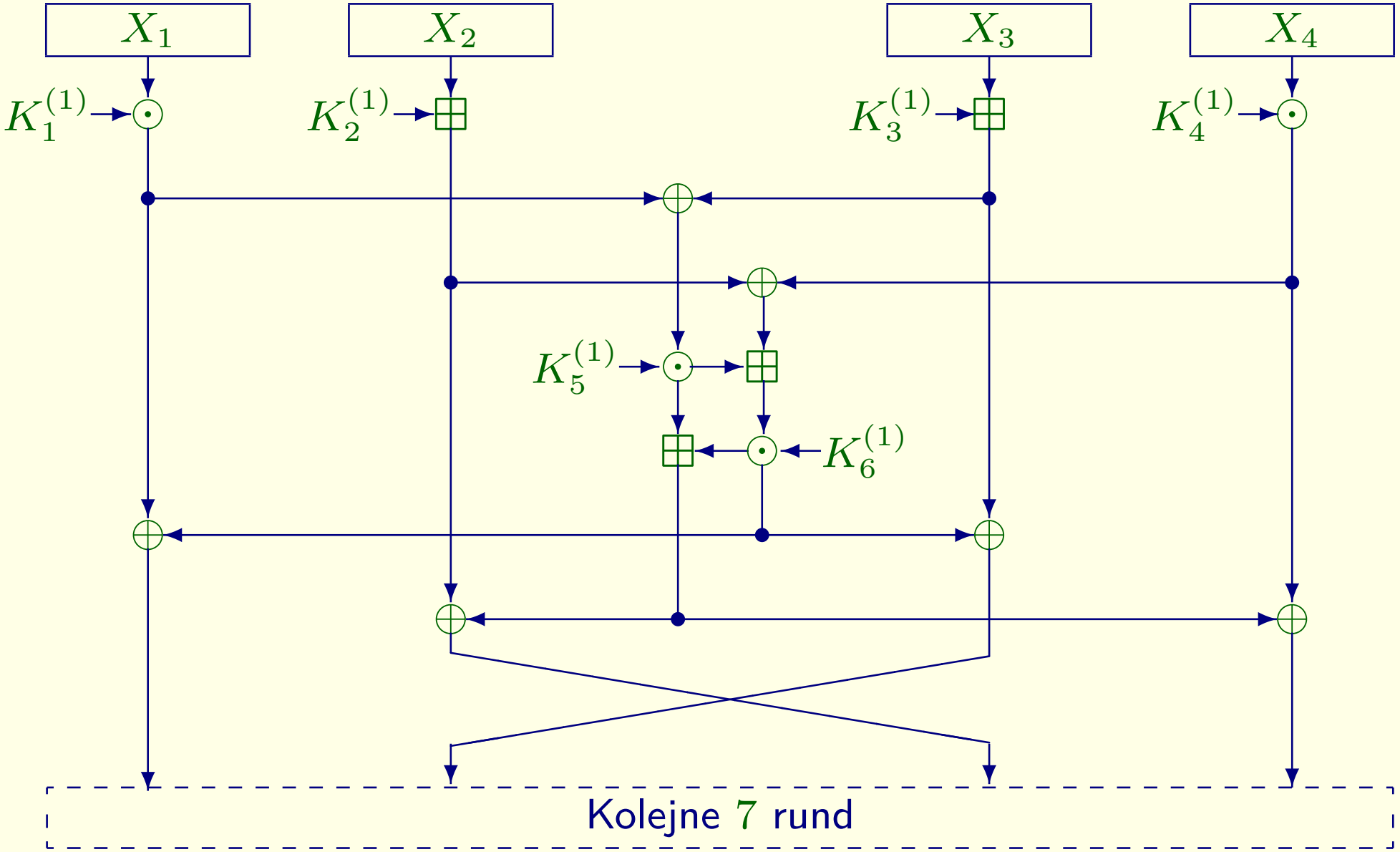

X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$

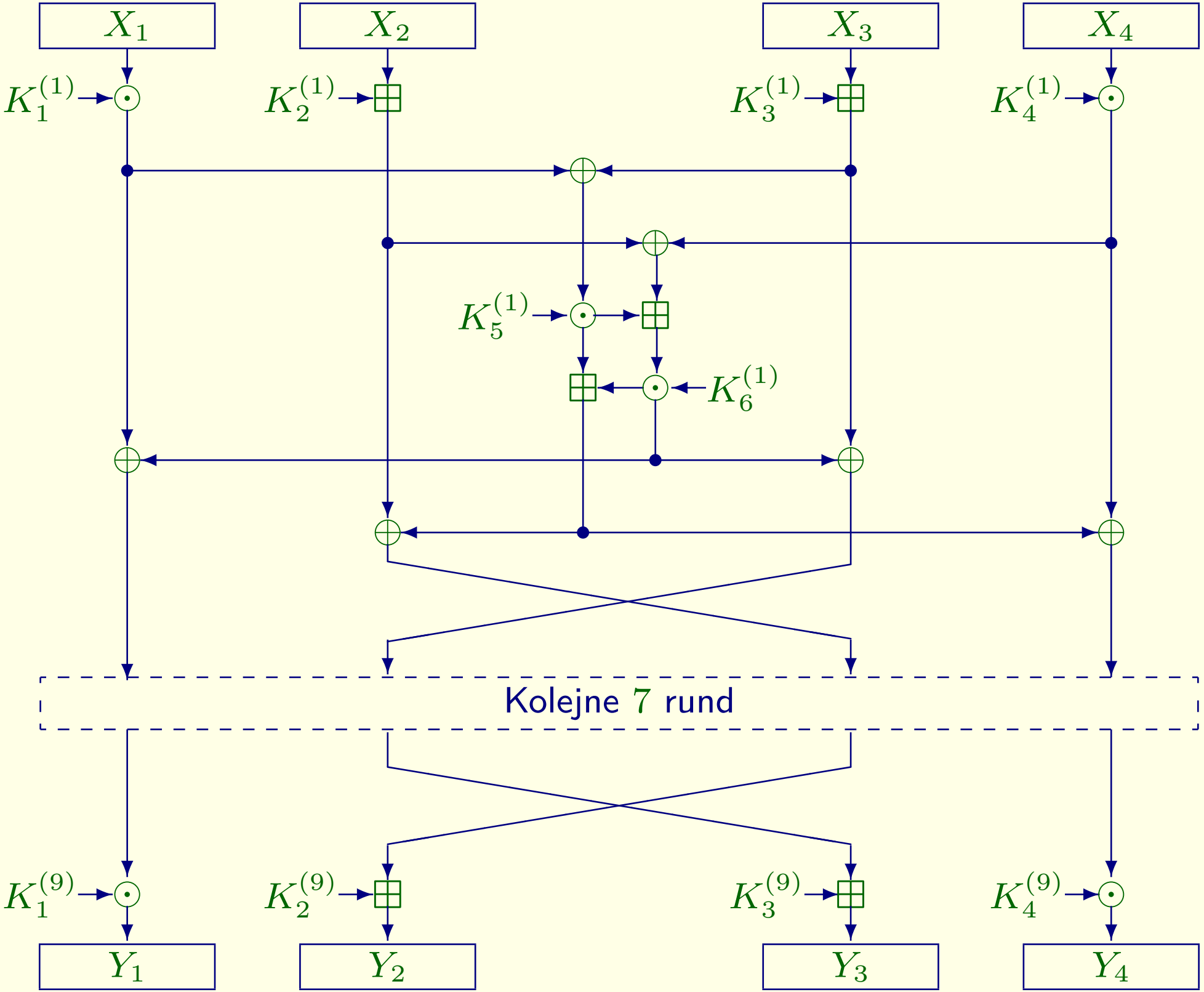

X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$


X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$


X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$


X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$


X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$
Kolejne 7 rund


X_1
X_2
X_3
X_4
$K_1^{(1)}$
$K_2^{(1)}$
$K_3^{(1)}$
$K_4^{(1)}$
$K_5^{(1)}$
$K_6^{(1)}$
Kolejne 7 rund
$K_1^{(9)}$
$K_2^{(9)}$
$K_3^{(9)}$
$K_4^{(9)}$
Y_1
Y_2
Y_3
Y_4

7.2 Generowanie podkluczy

- IDEA używa klucza 128 bitowego i wymaga $8 \times 6 + 4 = 52$ podklucze
  1. 128 bitowy klucz jest dzielony na bloki 16 bitowe, co daje 8 podkluczy
  2. Na kluczu wykonuje się przesunięcie cykliczne o 25 pozycji i znowu dzieli na bloki 16 bitowe, co daje kolejne 8 podkluczy
  3. Operację z punktu 2 powtarza się tak długo aż wygeneruje się wszystkie podklucze

7.2 Generowanie podkluczy

- IDEA używa klucza 128 bitowego i wymaga $8 \times 6 + 4 = 52$ podklucze
  1. 128 bitowy klucz jest dzielony na bloki 16 bitowe, co daje 8 podkluczy
  2. Na kluczu wykonuje się przesunięcie cykliczne o 25 pozycji i znowu dzieli na bloki 16 bitowe, co daje kolejne 8 podkluczy
  3. Operację z punktu 2 powtarza się tak długo aż wygeneruje się wszystkie podklucze

7.2 Generowanie podkluczy

- IDEA używa klucza 128 bitowego i wymaga $8 \times 6 + 4 = 52$ podklucze
  1. 128 bitowy klucz jest dzielony na bloki 16 bitowe, co daje 8 podkluczy
  2. Na kluczu wykonuje się przesunięcie cykliczne o 25 pozycji i znowu dzieli na bloki 16 bitowe, co daje kolejne 8 podkluczy
  3. Operację z punktu 2 powtarza się tak długo aż wygeneruje się wszystkie podklucze

7.2 Generowanie podkluczy

- IDEA używa klucza 128 bitowego i wymaga $8 \times 6 + 4 = 52$ podklucze
  1. 128 bitowy klucz jest dzielony na bloki 16 bitowe, co daje 8 podkluczy
  2. Na kluczu wykonuje się przesunięcie cykliczne o 25 pozycji i znowu dzieli na bloki 16 bitowe, co daje kolejne 8 podkluczy
  3. Operację z punktu 2 powtarza się tak długo aż wygeneruje się wszystkie podklucze

7.2 Generowanie podkluczy

- IDEA używa klucza 128 bitowego i wymaga $8 \times 6 + 4 = 52$ podklucze
  1. 128 bitowy klucz jest dzielony na bloki 16 bitowe, co daje 8 podkluczy
  2. Na kluczu wykonuje się przesunięcie cykliczne o 25 pozycji i znowu dzieli na bloki 16 bitowe, co daje kolejne 8 podkluczy
  3. Operację z punktu 2 powtarza się tak długo aż wygeneruje się wszystkie podklucze

7.3 Deszyfrowanie

- Deszyfrowanie algorytmem IDEA przebiega tak jak szyfrowanie, zamiast X_1, X_2, X_3, X_4 na wejściu podaje się bloki Y_1, Y_2, Y_3, Y_4 kryptogramu oraz klucz K (ten sam co przy szyfrowaniu)
- Z klucza K generuje się podklucze $K_i^{(r)}$
- Generuje się podklucze deszyfrujące $K'^{(r)}_i$ wg schematu przedstawionego w tablicy

7.3 Deszyfrowanie

- Deszyfrowanie algorytmem IDEA przebiega tak jak szyfrowanie, zamiast X_1, X_2, X_3, X_4 na wejściu podaje się bloki Y_1, Y_2, Y_3, Y_4 kryptogramu oraz klucz K (ten sam co przy szyfrowaniu)
- Z klucza K generuje się podklucze $K_i^{(r)}$
- Generuje się podklucze deszyfrujące $K'^{(r)}_i$ wg schematu przedstawionego w tablicy

7.3 Deszyfrowanie

- Deszyfrowanie algorytmem IDEA przebiega tak jak szyfrowanie, zamiast X_1, X_2, X_3, X_4 na wejściu podaje się bloki Y_1, Y_2, Y_3, Y_4 kryptogramu oraz klucz K (ten sam co przy szyfrowaniu)
- Z klucza K generuje się podklucze $K_i^{(r)}$
- Generuje się podklucze deszyfrujące ${K'}_i^{(r)}$ wg schematu przedstawionego w tablicy

7.3 Deszyfrowanie

- Deszyfrowanie algorytmem IDEA przebiega tak jak szyfrowanie, zamiast X_1, X_2, X_3, X_4 na wejściu podaje się bloki Y_1, Y_2, Y_3, Y_4 kryptogramu oraz klucz K (ten sam co przy szyfrowaniu)
- Z klucza K generuje się podklucze $K_i^{(r)}$
- Generuje się podklucze deszyfrujące $K'^{(r)}_i$ wg schematu przedstawionego w tablicy

Runda	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$
$r = 1$	$\left(K_1^{(10-r)}\right)^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$\left(K_4^{(10-r)}\right)^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$2 \leq r \leq 8$	$\left(K_1^{(10-r)}\right)^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$	$\left(K_4^{(10-r)}\right)^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$r = 9$	$\left(K_1^{(10-r)}\right)^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$	$\left(K_4^{(10-r)}\right)^{-1}$	—	—

Tworzenie podkluczy deszyfrujących $K'^{(r)}_i$ na podstawie podkluczy szyfrujących $K_i^{(r)}$ w algorytmie IDEA (dla $K_i = 0$ przyjmuje się $(K_i)^{-1} = 0$; $2^{16} \equiv -1 \bmod 2^{16} + 1$)

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

8 AES — Advanced Encryption Standard — Rijndael

- Nowy standard przyjęty w 2001 r. w USA
- Algorytm blokowy, który zaprojektowali Joan Daemen i Vincent Rijmen
- Zarówno długość bloku jak i klucza może być wybrana jako 128, 192 lub 256 bitów
- Rijndael jest ogólnie dostępny
- Liczba rund zależy od długości bloku

- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

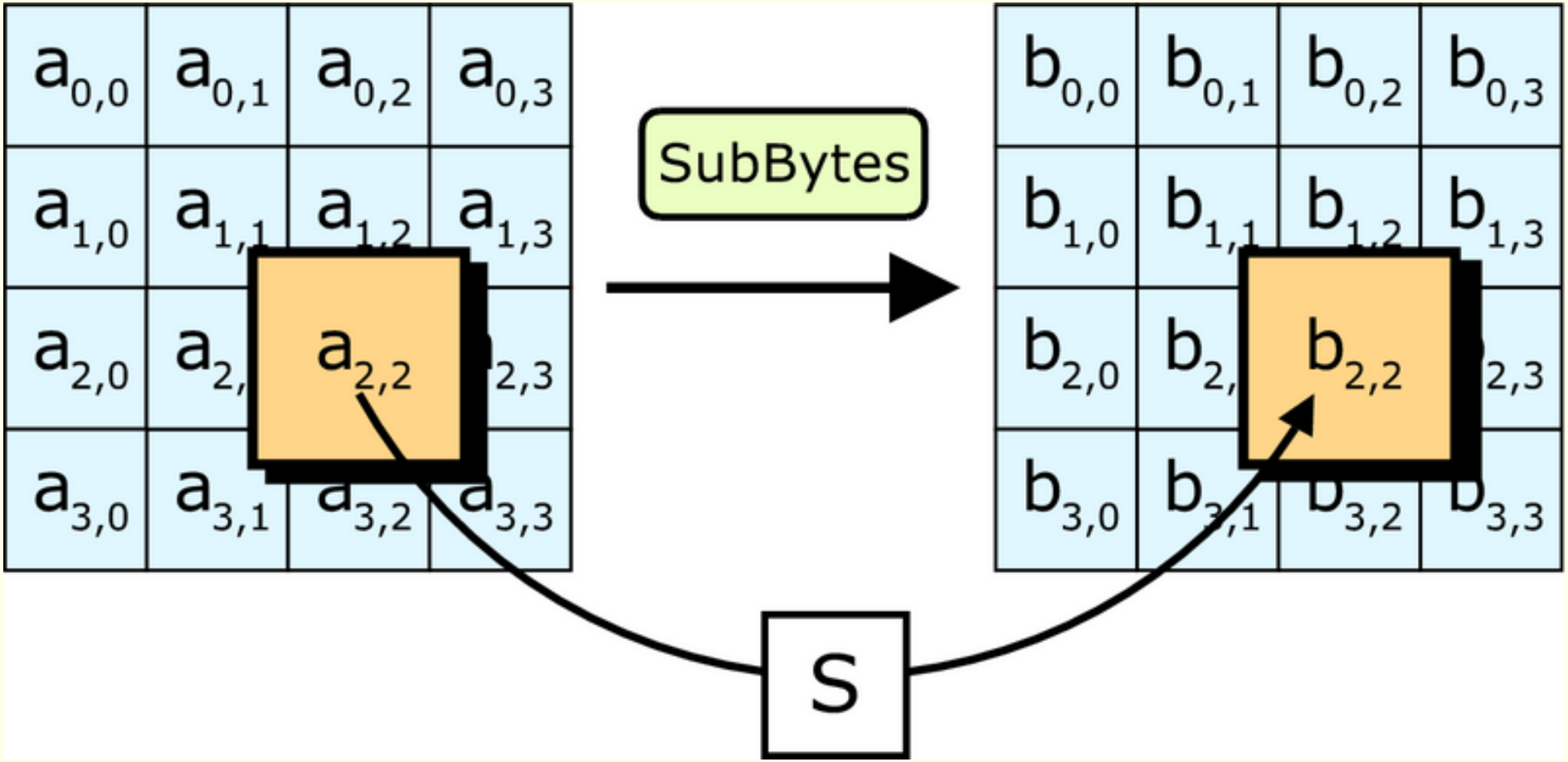
- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

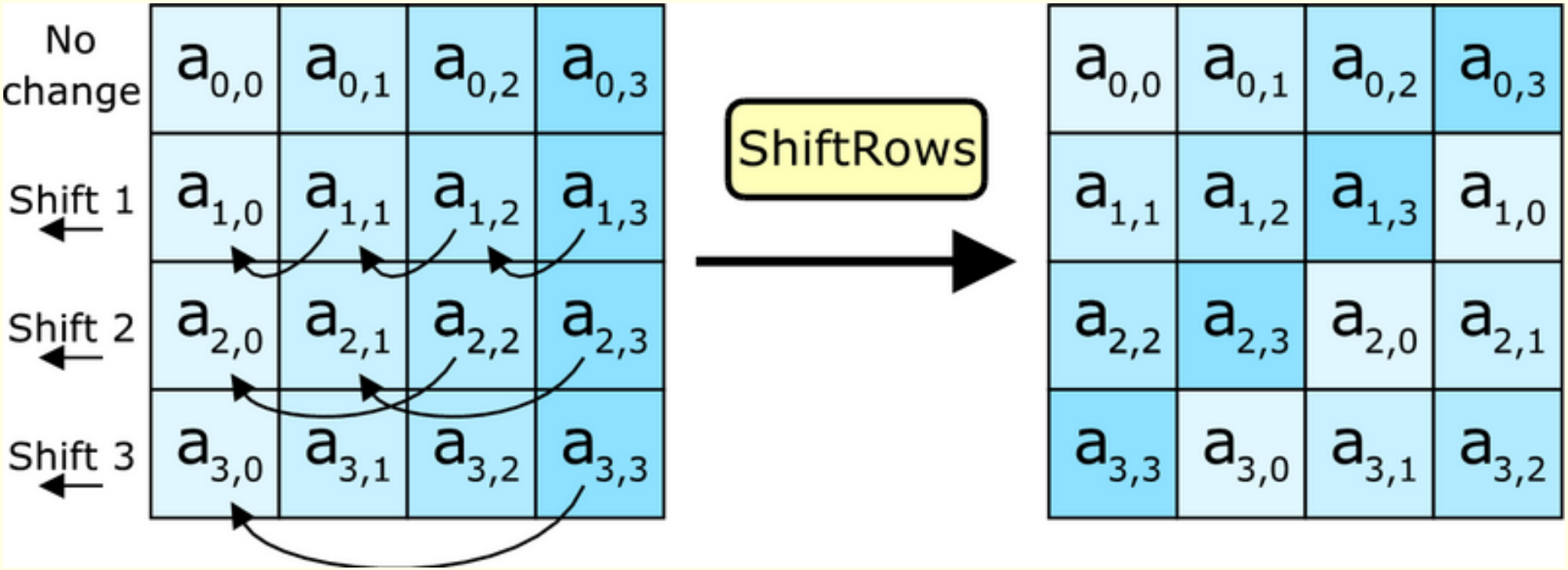
- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

- W każdej rundzie wykonywane są 4 operacje (macierzowe):
  - podstawienie w S-boksie
  - przesunięcie wierszy
  - mieszanie kolumn
  - xor z podkluczem
- Podklucze są generowane algorytmem, który zależy od rundy

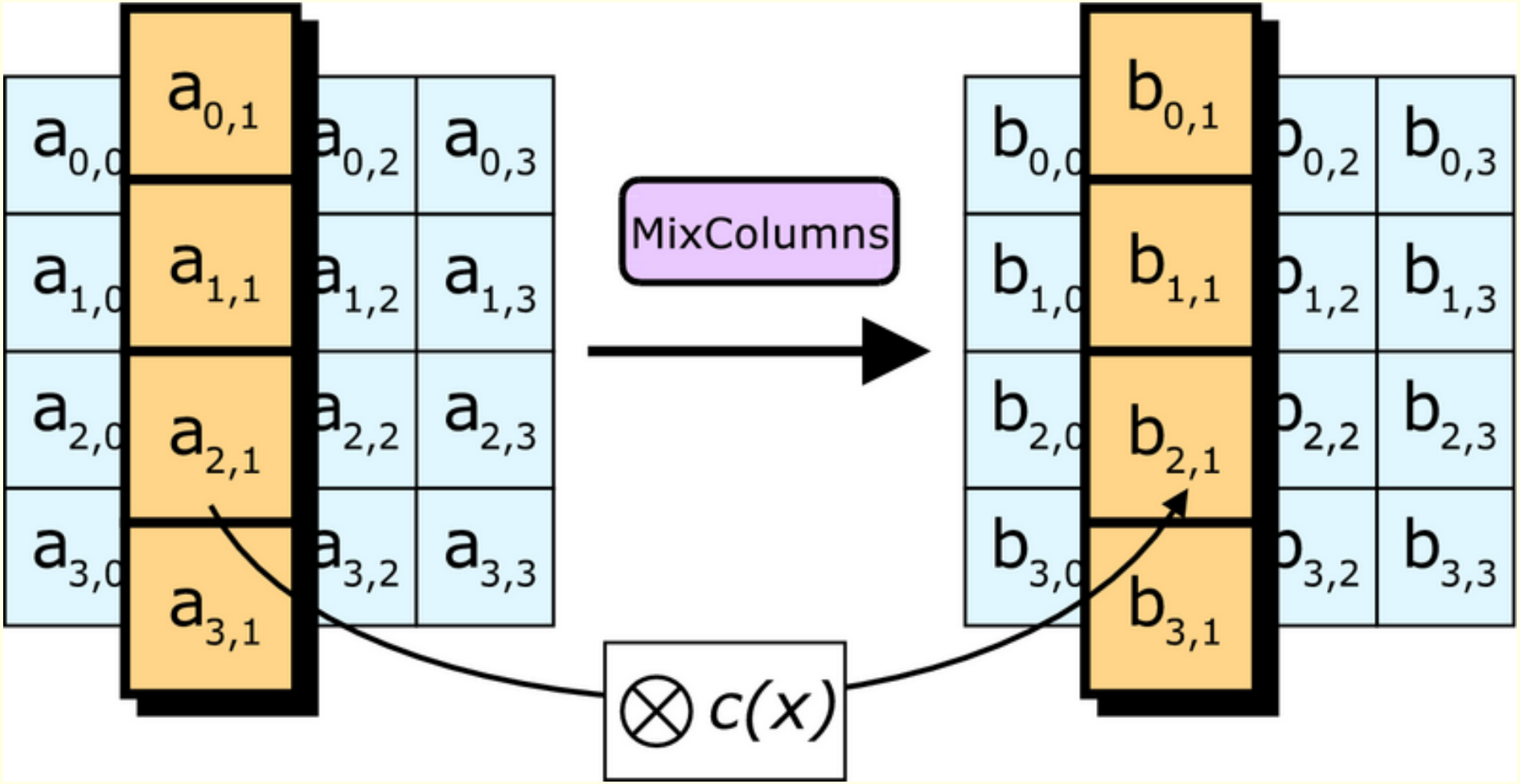
Podstawienie w S-boksie



Przesunięcie wierszy



Mieszanie kolumn



Operacja xor z kluczem