# Kryptografia

## z elementami kryptografii kwantowej

**Ryszard Tanaś**

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

**Wykład 2**

# Spis treści

# 5 DES — Data Encryption Standard

- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

# 5 DES — Data Encryption Standard

- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

# 5 DES — Data Encryption Standard

- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

# 5 DES — Data Encryption Standard

- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

# 5 DES — Data Encryption Standard

- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

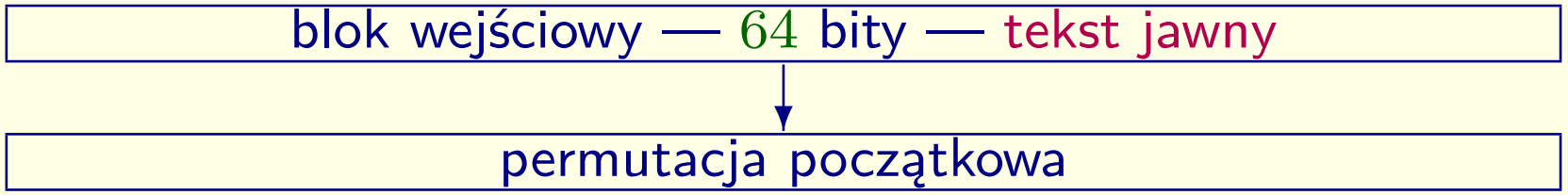
# 5 DES — Data Encryption Standard

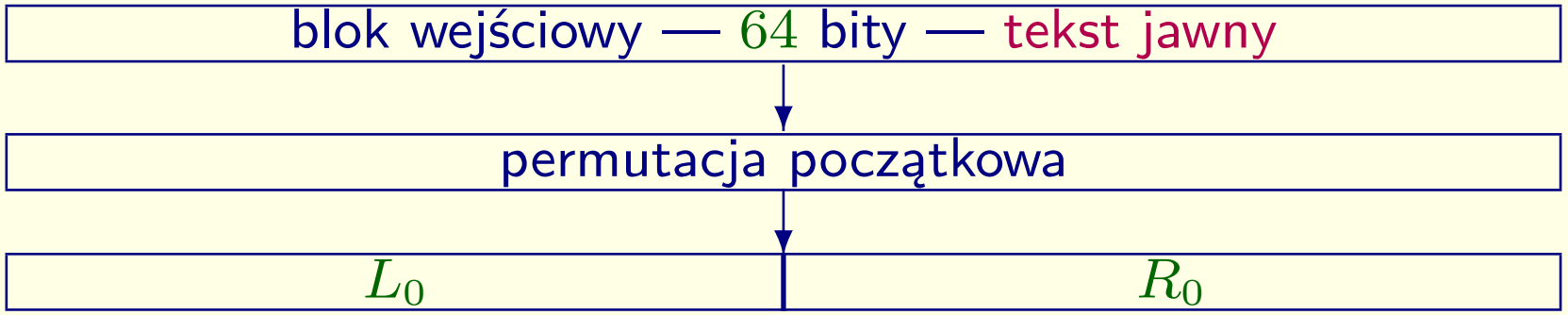
- w 1981 r. przyjęty w USA jako standard do celów cywilnych (nadal używany, choć nowym standardem od 2001 r. jest AES)

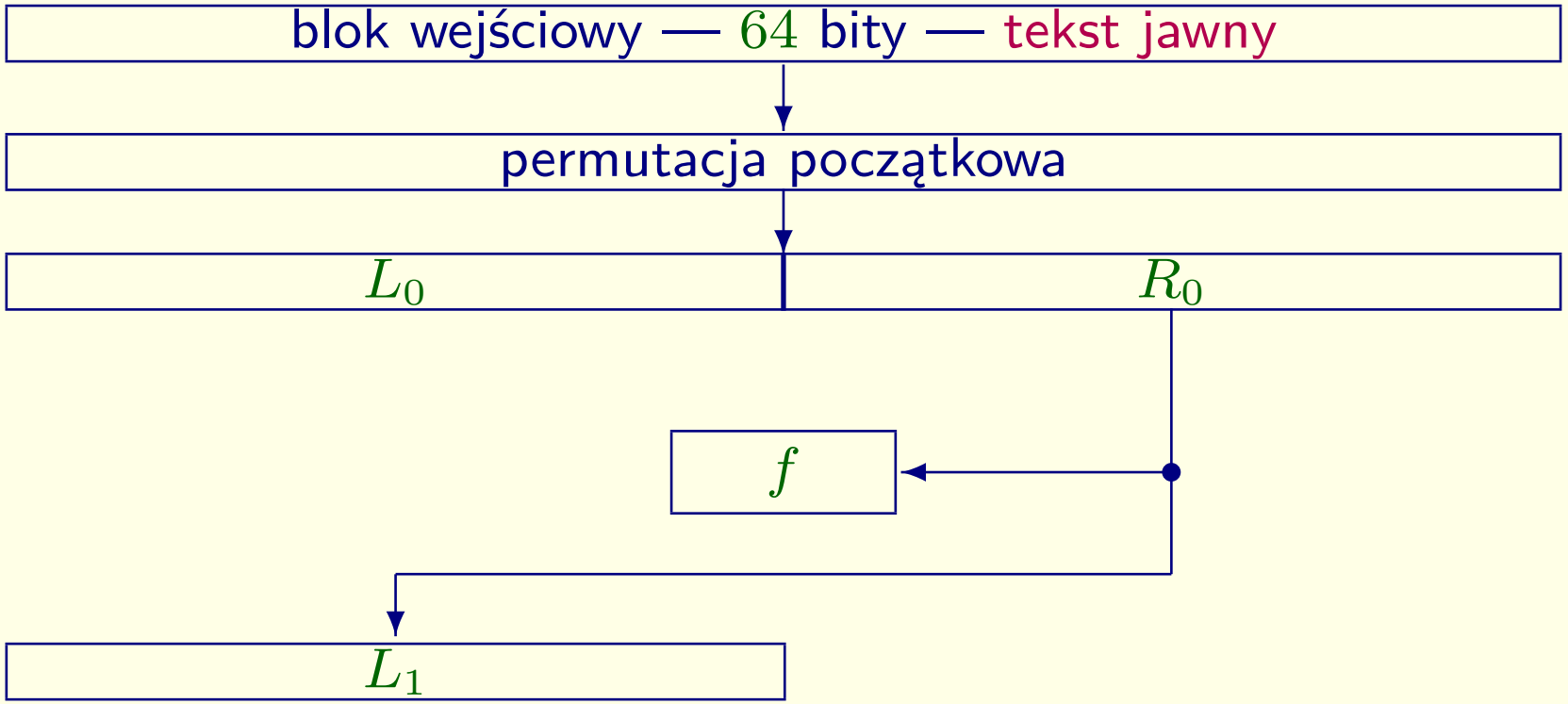
- algorytm symetryczny
- szczegóły algorytmu zostały opublikowane (podejrzenia o tylne drzwi)
- szyfruje bloki $64$ bitowe ($8$ liter ASCII z bitem parzystości)
- klucze są efektywnie $56$ bitowe ($64$ bity minus $8$ bitów parzystości); obecnie uważa się, że taka długość klucza jest zbyt mała

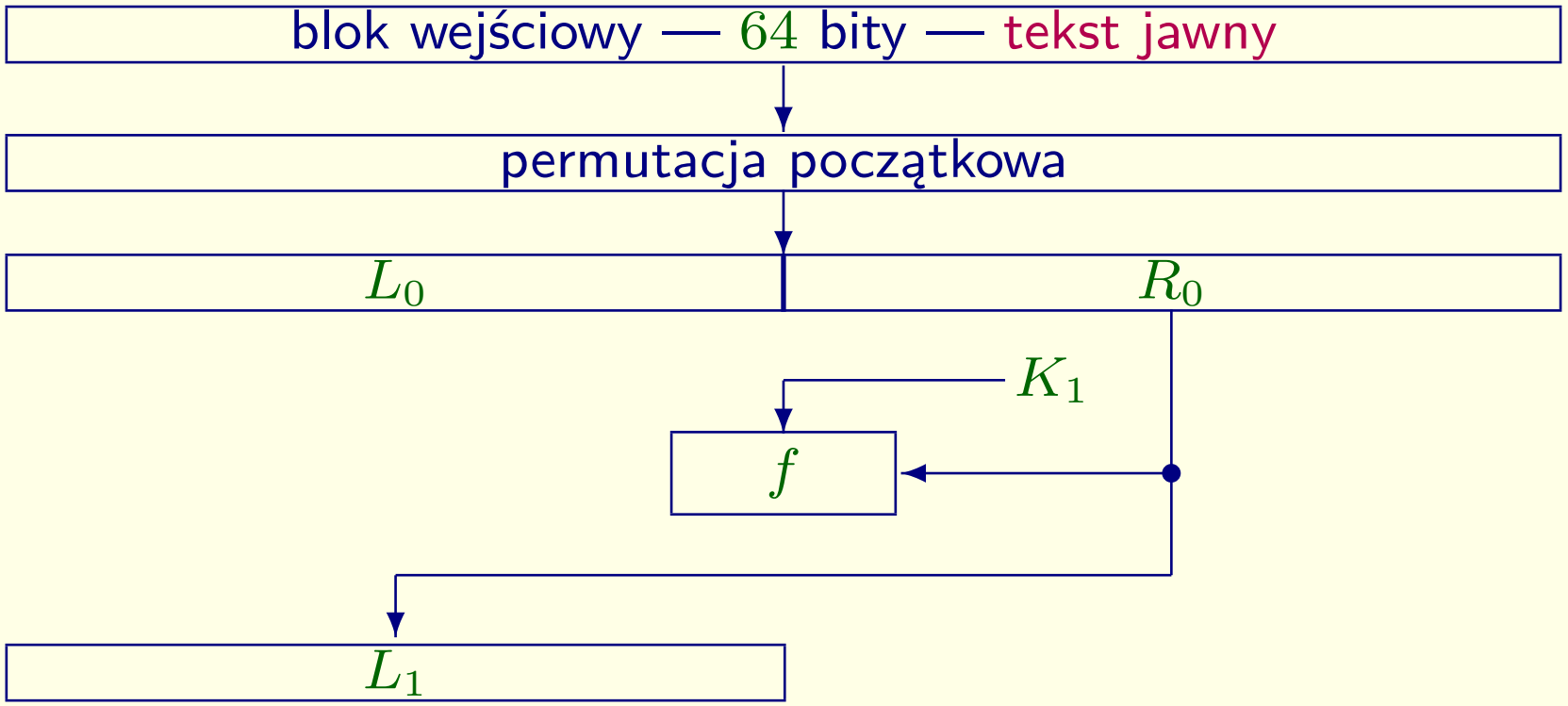
# 5.1 Etapy DES

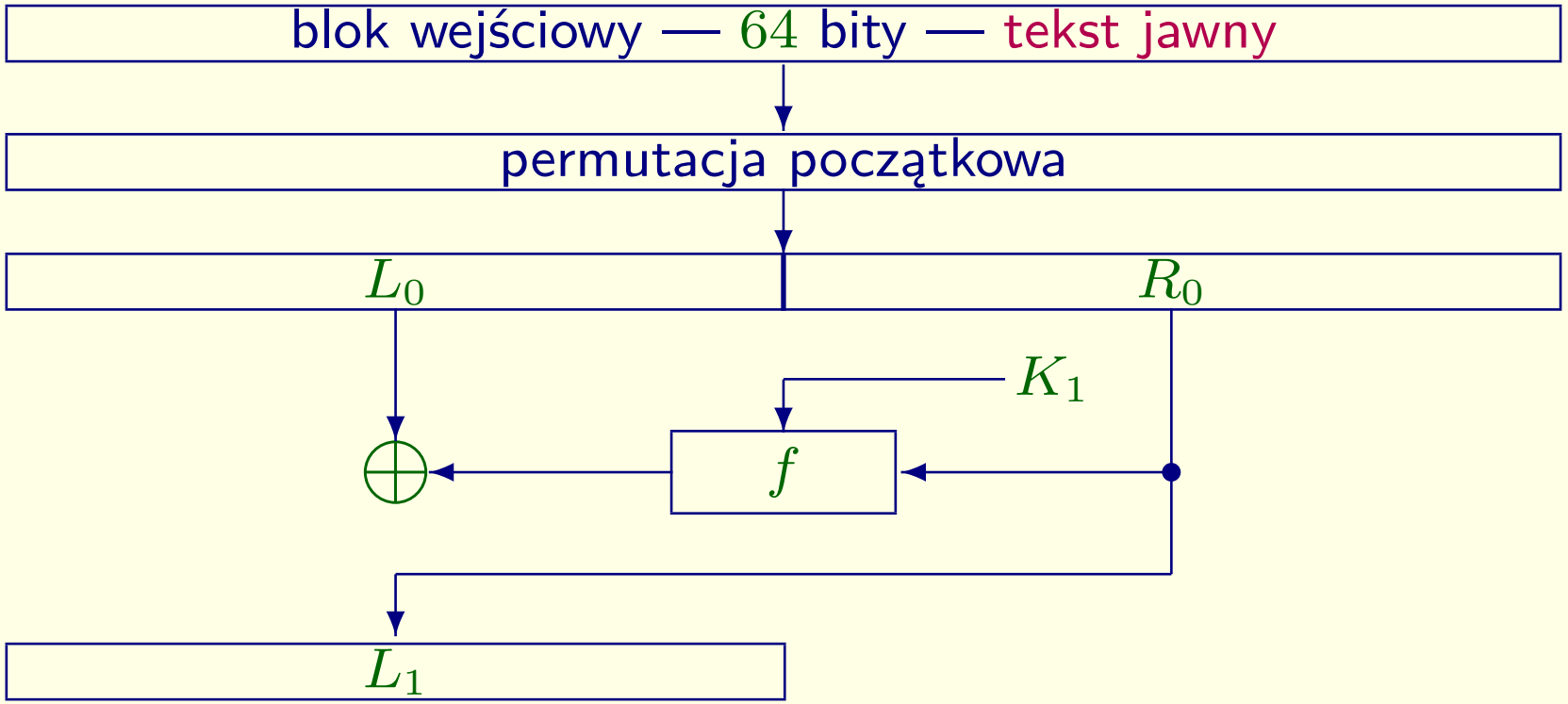

blok wejściowy — 64 bity — tekst jawny

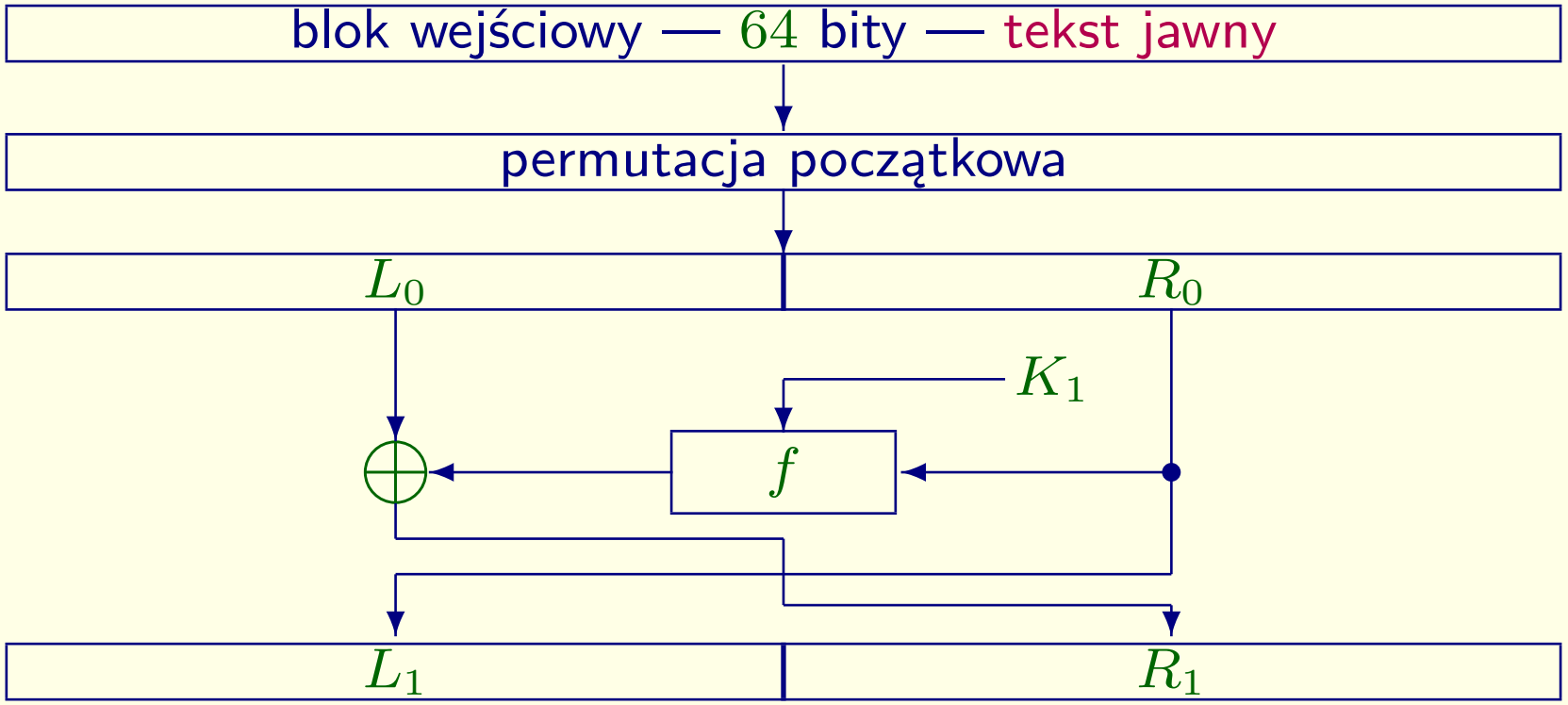

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa

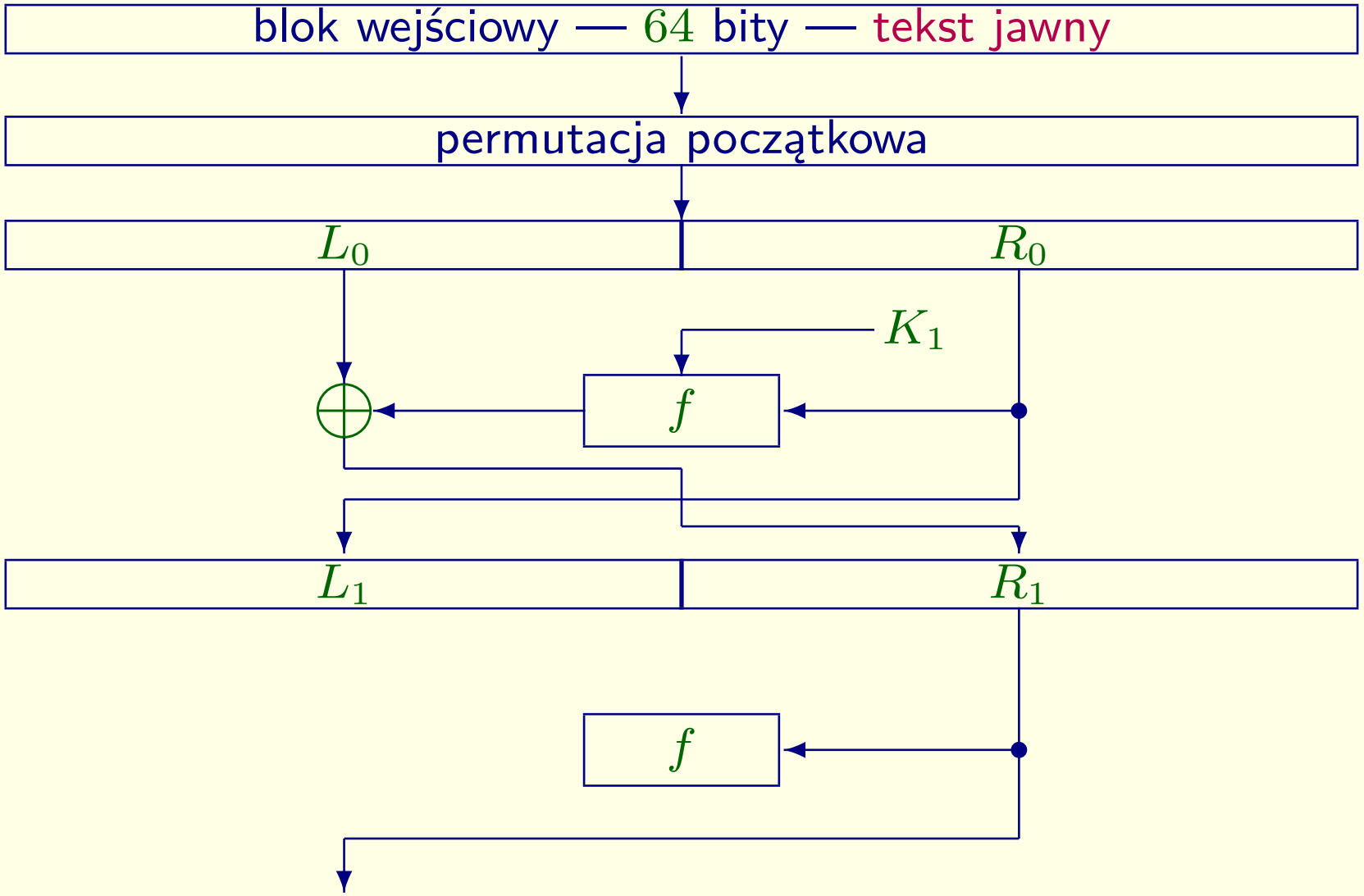

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$

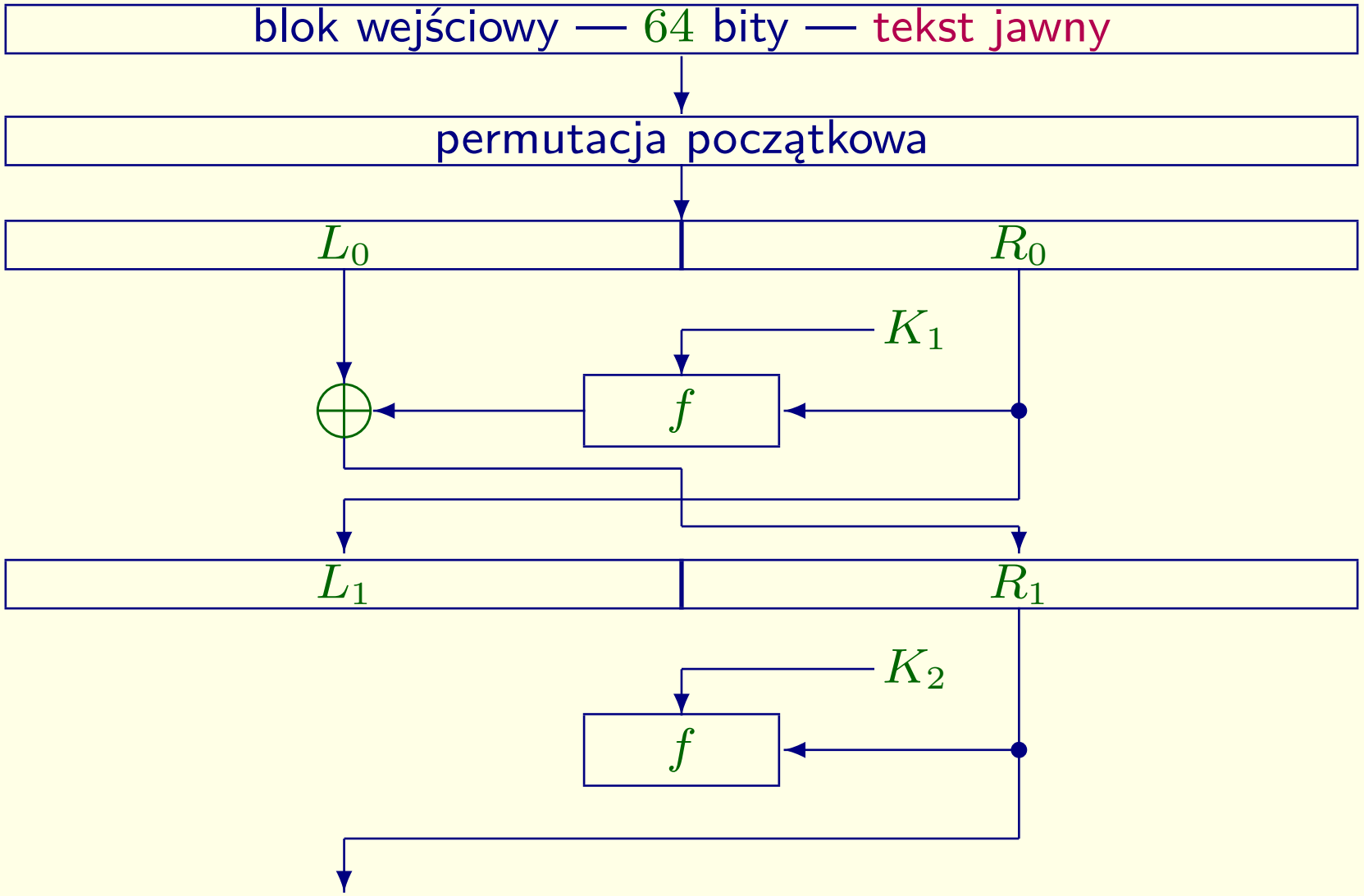

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$f$
$L_1$

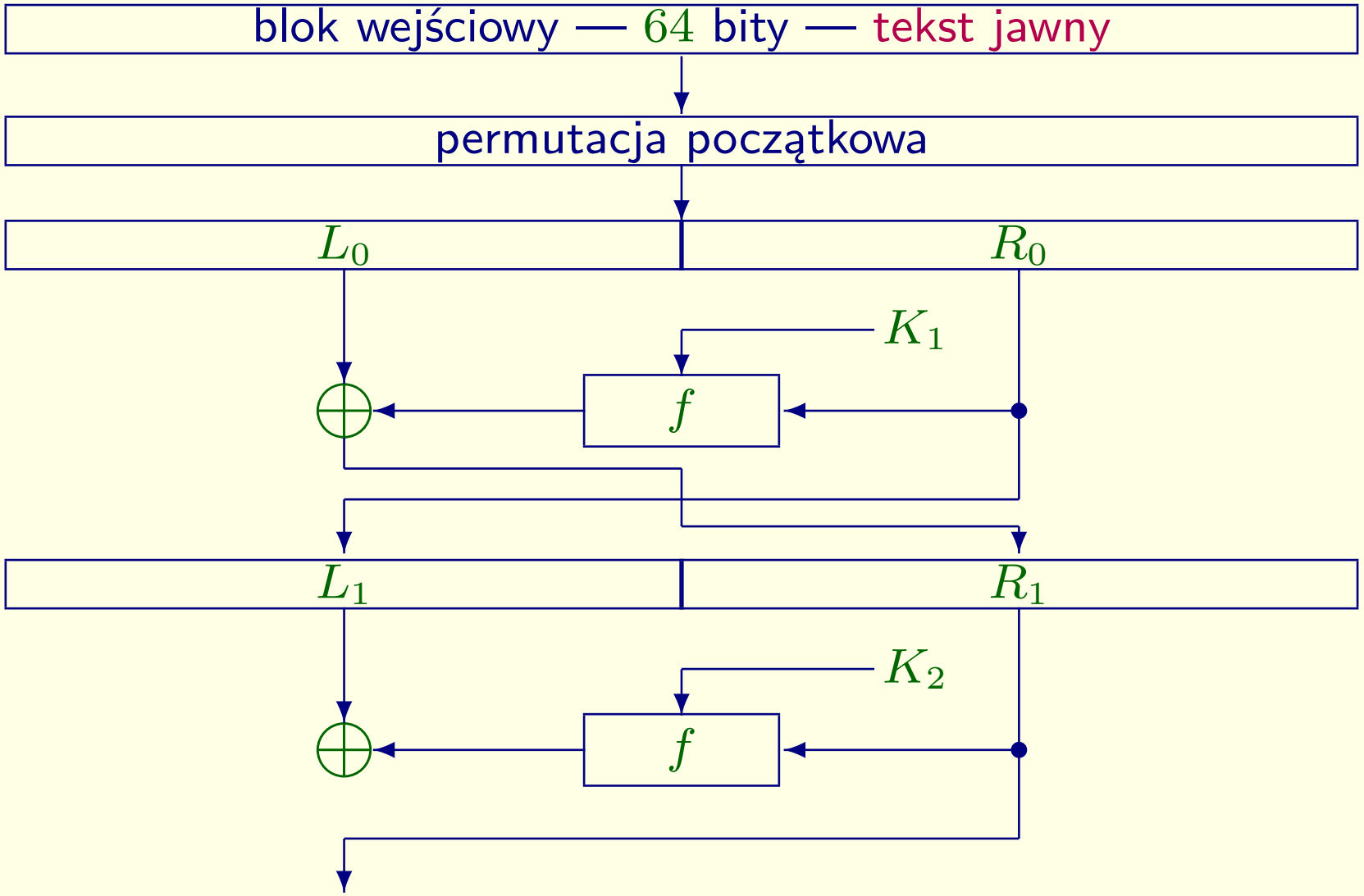

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$

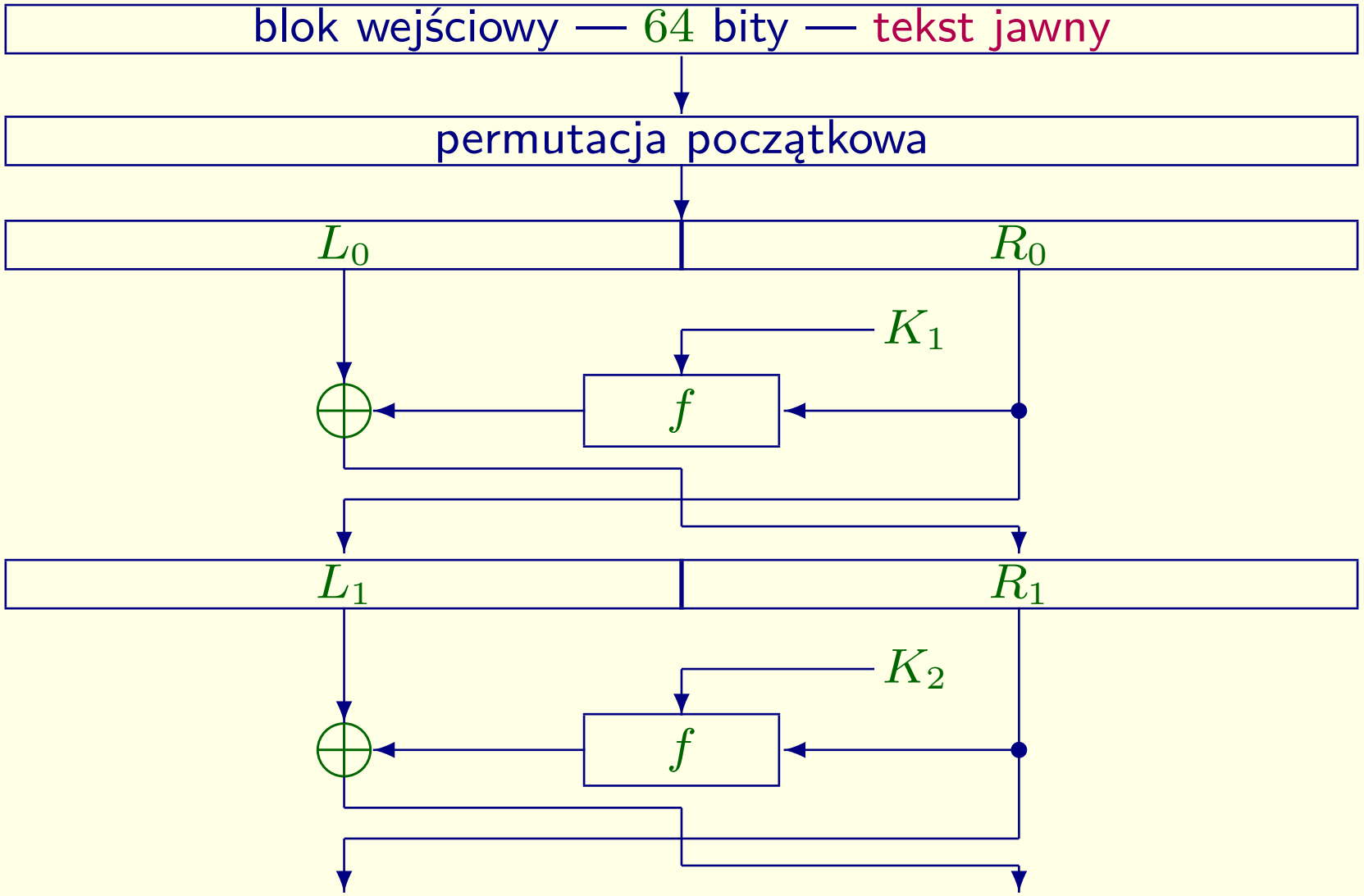

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$

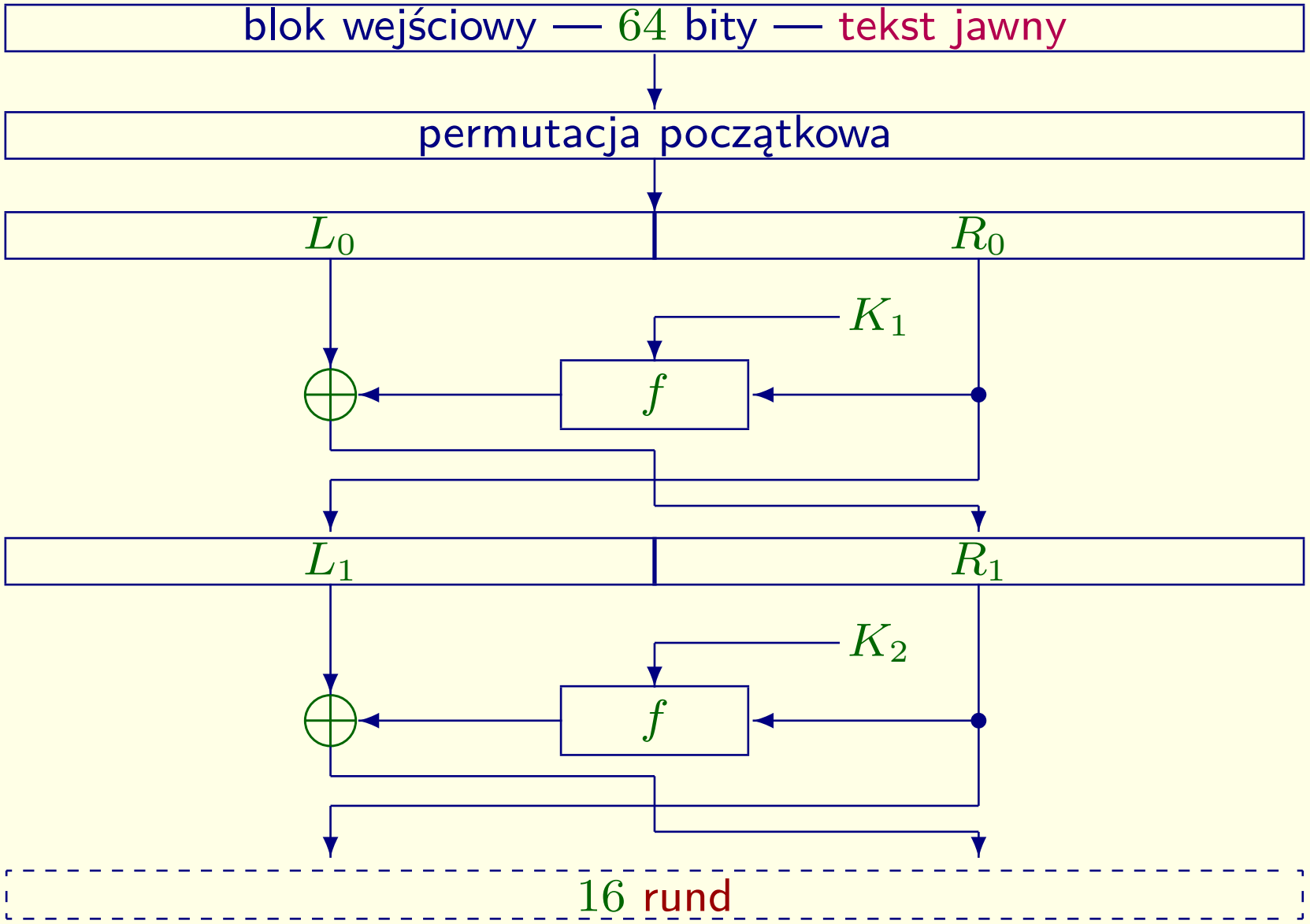

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$

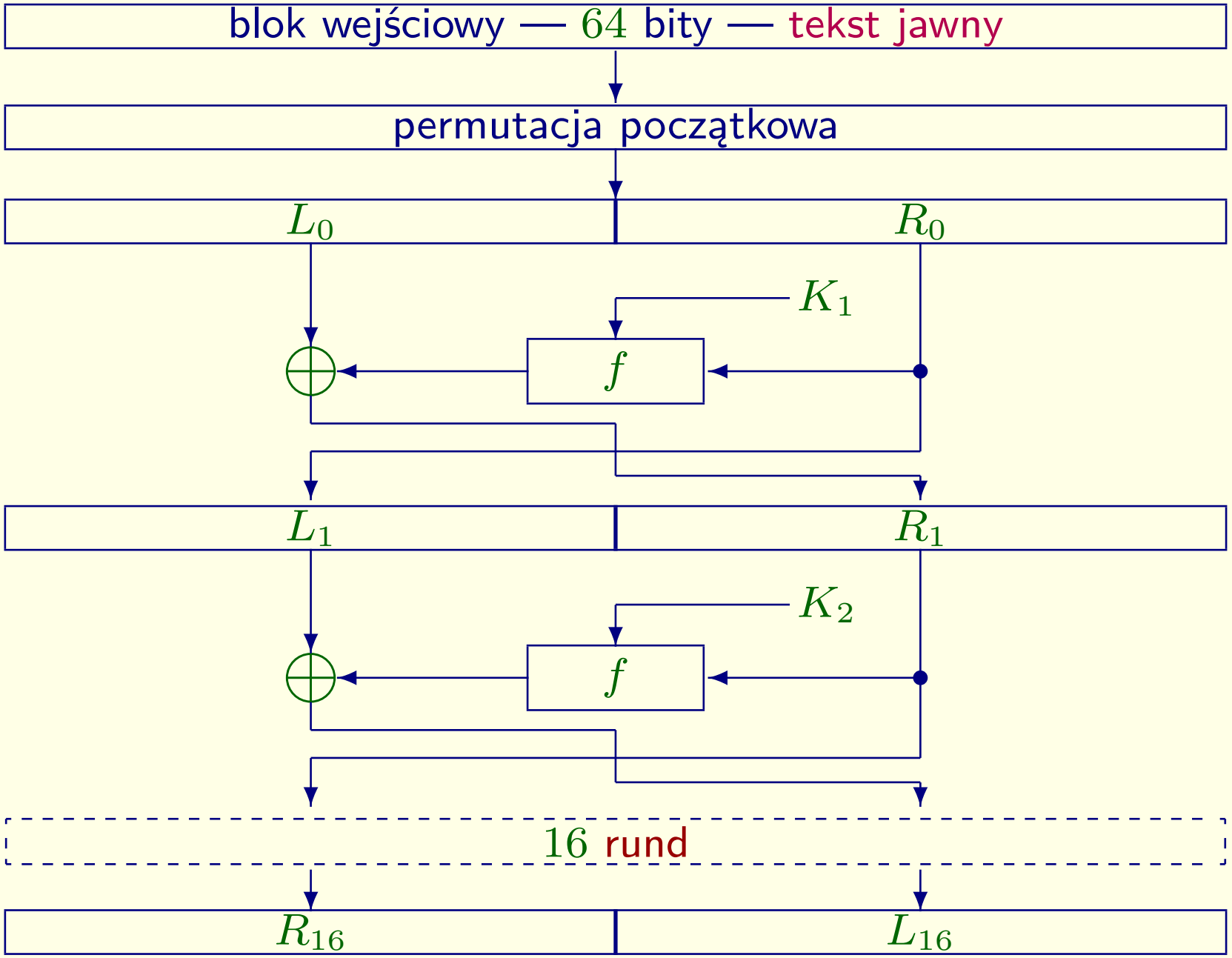

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$f$

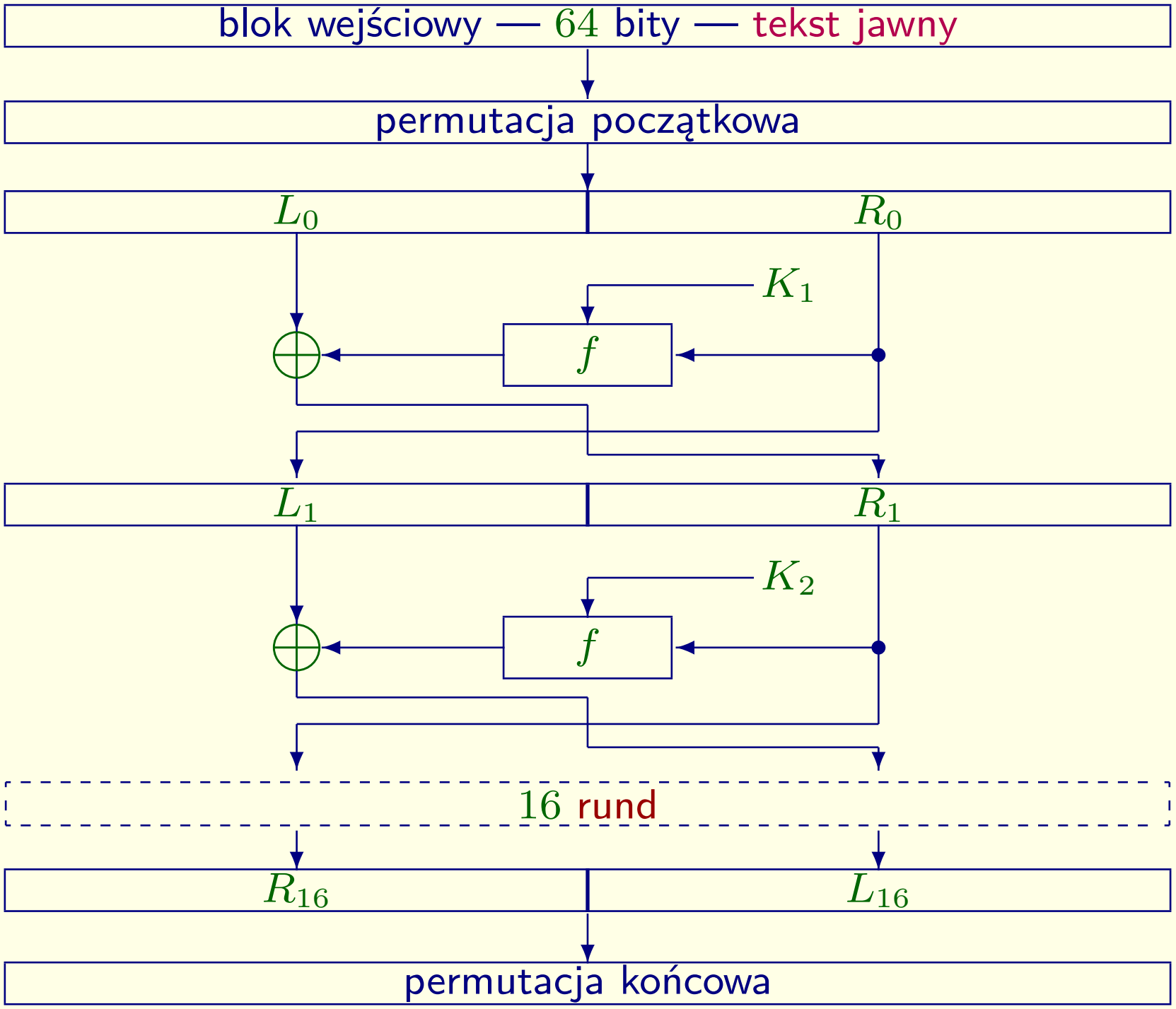

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$

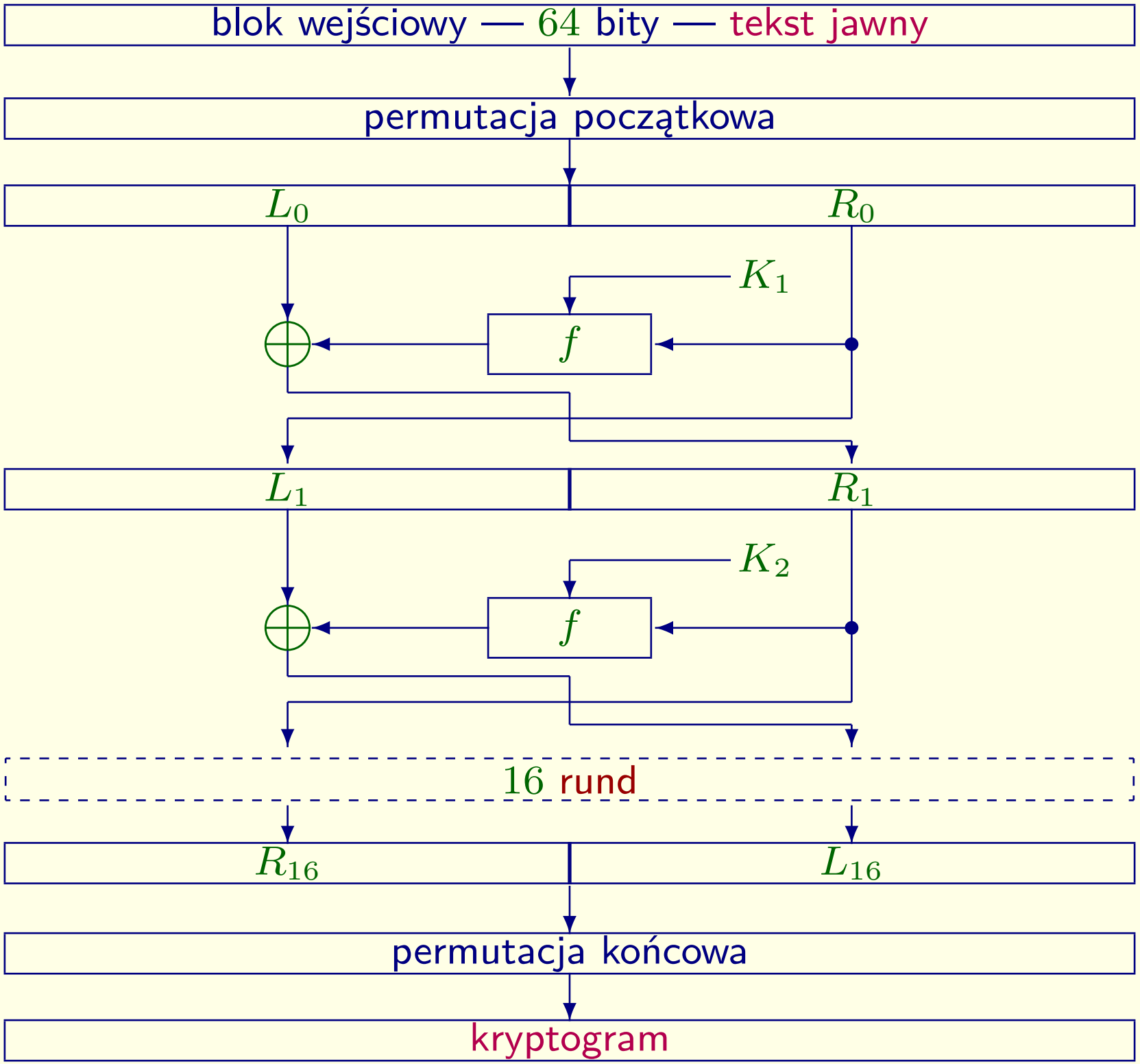

blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$


blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$


blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$
16 rund


blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$
16 rund
$R_{16}$
$L_{16}$


blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$
16 rund
$R_{16}$
$L_{16}$
permutacja końcowa


blok wejściowy — 64 bity — tekst jawny
permutacja początkowa
$L_0$
$R_0$
$K_1$
$f$
$L_1$
$R_1$
$K_2$
$f$
16 rund
$R_{16}$
$L_{16}$
permutacja końcowa
kryptogram

- wejście — $64$ bitowy blok
- permutacja początkowa
- blok zostaje podzielony na lewą i prawą połowę po $32$ bity każda
- $16$ rund identycznych operacji opisanych funkcją $f$, w czasie których dane prawej połowy są przekształcane z użyciem klucza

- wejście — $64$ bitowy blok
- permutacja początkowa
- blok zostaje podzielony na lewą i prawą połowę po $32$ bity każda
- $16$ rund identycznych operacji opisanych funkcją $f$, w czasie których dane prawej połowy są przekształcane z użyciem klucza

- wejście — $64$ bitowy blok
- permutacja początkowa
- blok zostaje podzielony na lewą i prawą połowę po $32$ bity każda
- $16$ rund identycznych operacji opisanych funkcją $f$, w czasie których dane prawej połowy są przekształcane z użyciem klucza

- wejście — $64$ bitowy blok
- permutacja początkowa
- blok zostaje podzielony na lewą i prawą połowę po $32$ bity każda
- $16$ rund identycznych operacji opisanych funkcją $f$, w czasie których dane prawej połowy są przekształcane z użyciem klucza

- jedna runda — funkcja $f$
  - w czasie każdej rundy bity klucza są przesuwane, a następnie $48$ bitowy podklucz jest wybierany z $56$ bitowego klucza
  - prawa część danych jest rozszerzana do $48$ bitów za pomocą permutacji rozszerzającej a następnie podlega operacji xor z $48$ bitami podklucza
  - wynik wysyłany jest do $8$ $S$-boksów, które produkują nowe $32$ bity
  - otrzymane $32$ bity są permutowane w $P$-boksie

- jedna runda — funkcja $f$
  - w czasie każdej rundy bity klucza są przesuwane, a następnie $48$ bitowy podklucz jest wybierany z $56$ bitowego klucza
  - prawa część danych jest rozszerzana do $48$ bitów za pomocą permutacji rozszerzającej a następnie podlega operacji xor z $48$ bitami podklucza
  - wynik wysyłany jest do $8$ $S$-boksów, które produkują nowe $32$ bity
  - otrzymane $32$ bity są permutowane w $P$-boksie

- jedna runda — funkcja $f$
  - w czasie każdej rundy bity klucza są przesuwane, a następnie $48$ bitowy podklucz jest wybierany z $56$ bitowego klucza
  - prawa część danych jest rozszerzana do $48$ bitów za pomocą permutacji rozszerzającej a następnie podlega operacji xor z $48$ bitami podklucza
  - wynik wysyłany jest do $8$ $S$-boksów, które produkują nowe $32$ bity
  - otrzymane $32$ bity są permutowane w $P$-boksie

- jedna runda — funkcja $f$
  - w czasie każdej rundy bity klucza są przesuwane, a następnie $48$ bitowy podklucz jest wybierany z $56$ bitowego klucza
  - prawa część danych jest rozszerzana do $48$ bitów za pomocą permutacji rozszerzającej a następnie podlega operacji xor z $48$ bitami podklucza
  - wynik wysyłany jest do $8$ $S$-boksów, które produkują nowe $32$ bity
  - otrzymane $32$ bity są permutowane w $P$-boksie

- jedna runda — funkcja $f$
  - w czasie każdej rundy bity klucza są przesuwane, a następnie $48$ bitowy podklucz jest wybierany z $56$ bitowego klucza
  - prawa część danych jest rozszerzana do $48$ bitów za pomocą permutacji rozszerzającej a następnie podlega operacji xor z $48$ bitami podklucza
  - wynik wysyłany jest do $8$ $S$-boksów, które produkują nowe $32$ bity
  - otrzymane $32$ bity są permutowane w $P$-boksie

- wynik tych $4$ operacji stanowiących funkcję $f$ podlega operacji xor z lewą połową i staje się nową prawą połową
- stara prawa połowa staje się nową lewą połową, i tak $16$ razy
- permutacja końcowa
- kryptogram

- wynik tych $4$ operacji stanowiących funkcję $f$ podlega operacji xor z lewą połową i staje się nową prawą połową
- stara prawa połowa staje się nową lewą połową, i tak $16$ razy
- permutacja końcowa
- kryptogram

- wynik tych $4$ operacji stanowiących funkcję $f$ podlega operacji xor z lewą połową i staje się nową prawą połową
- stara prawa połowa staje się nową lewą połową, i tak $16$ razy
- permutacja końcowa
- kryptogram

- wynik tych $4$ operacji stanowiących funkcję $f$ podlega operacji xor z lewą połową i staje się nową prawą połową
- stara prawa połowa staje się nową lewą połową, i tak $16$ razy
- permutacja końcowa
- kryptogram

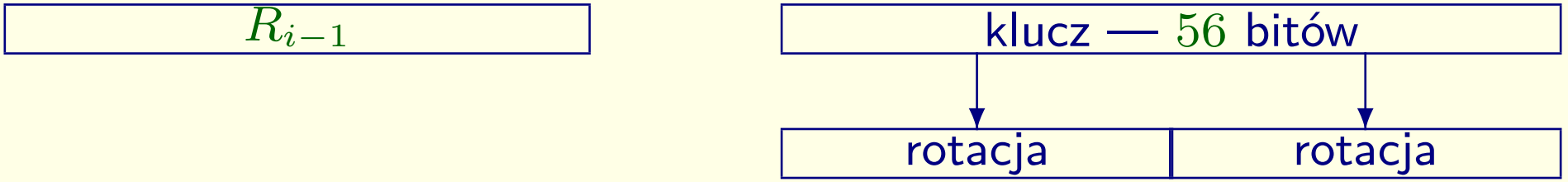
# Jedna runda DES'a

# Jedna runda DES'a

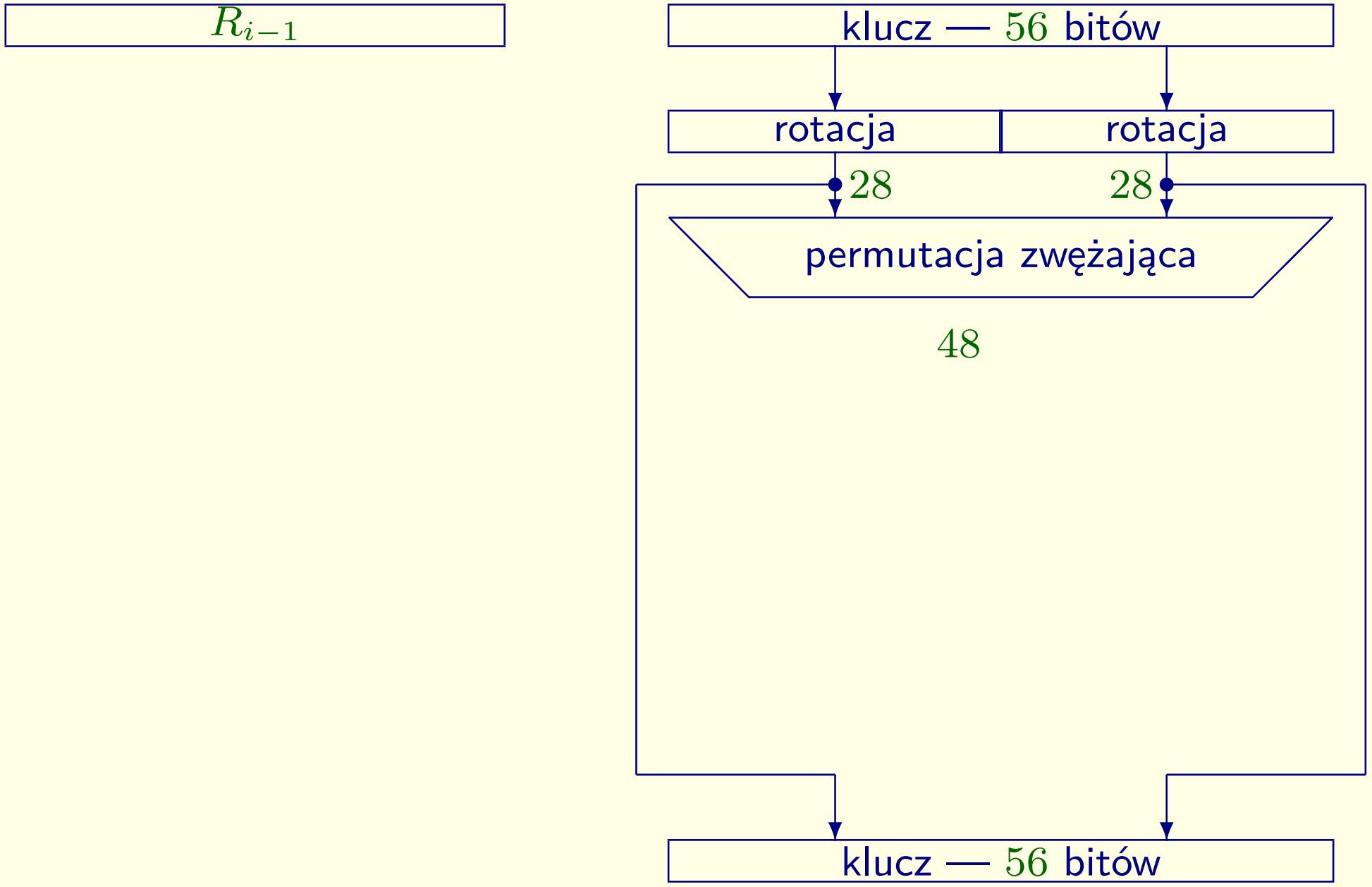
$R_{i-1}$

klucz — $56$ bitów

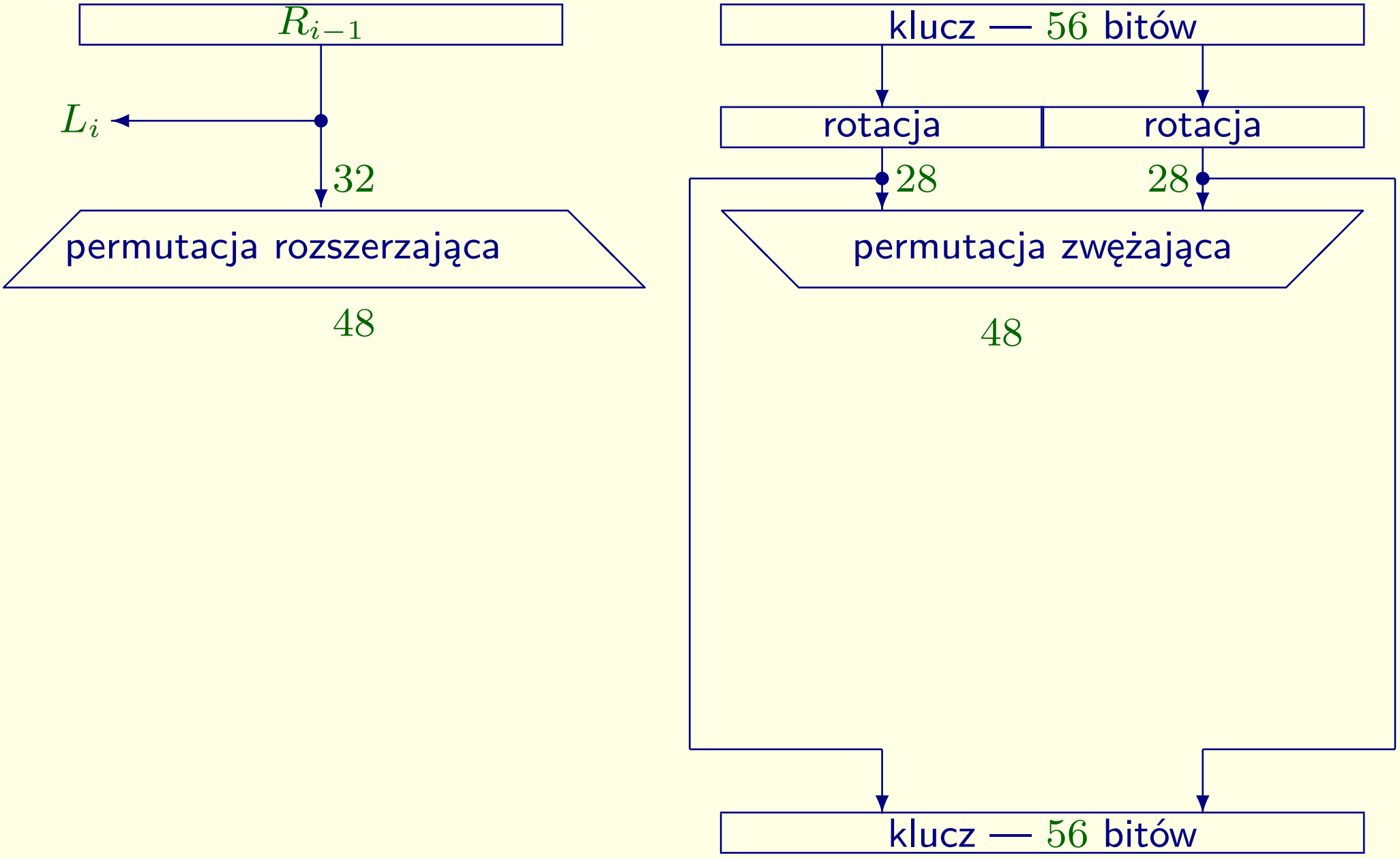
# Jedna runda DES'a



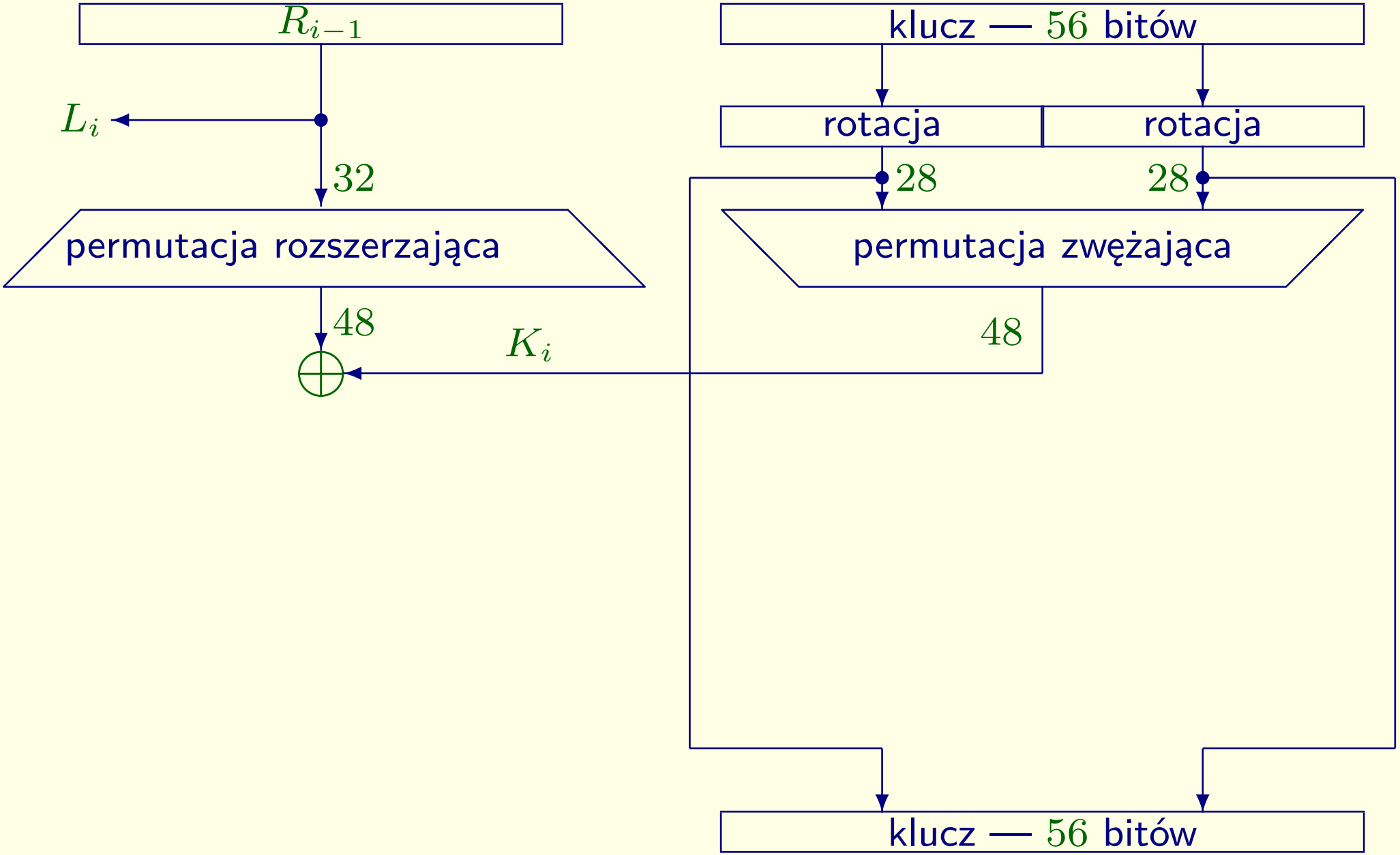
# Jedna runda DES'a



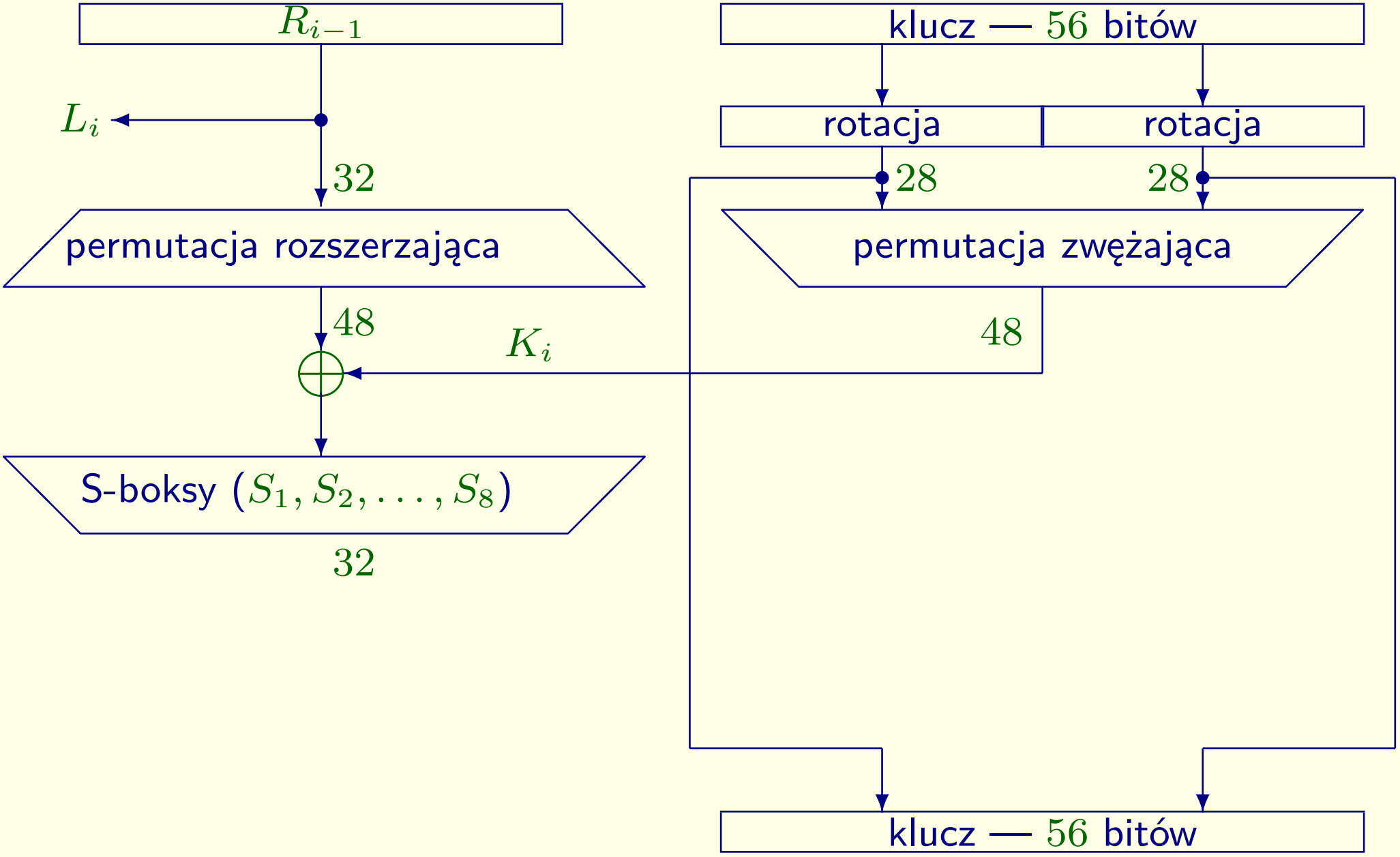
# Jedna runda DES'a



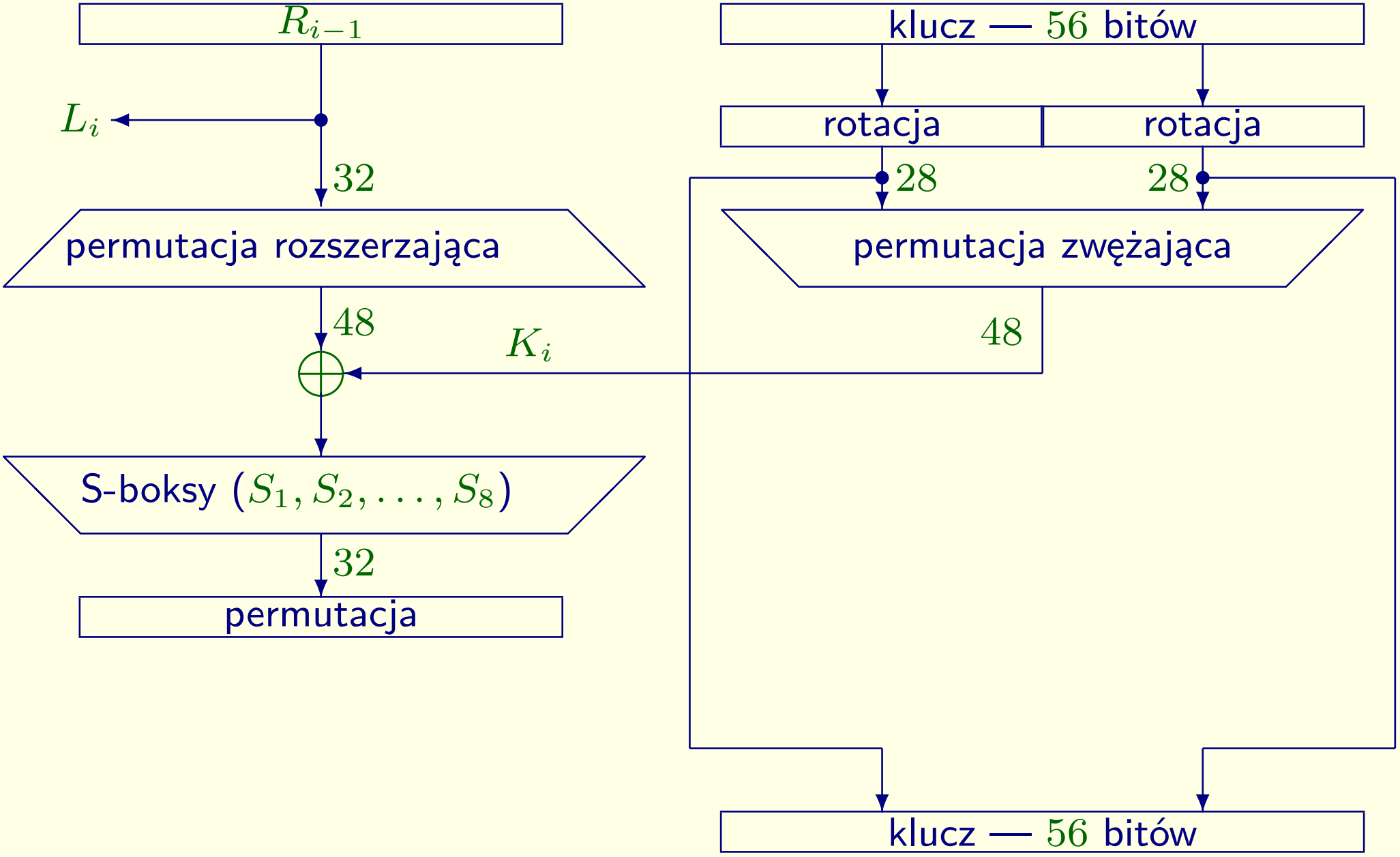
# Jedna runda DES'a



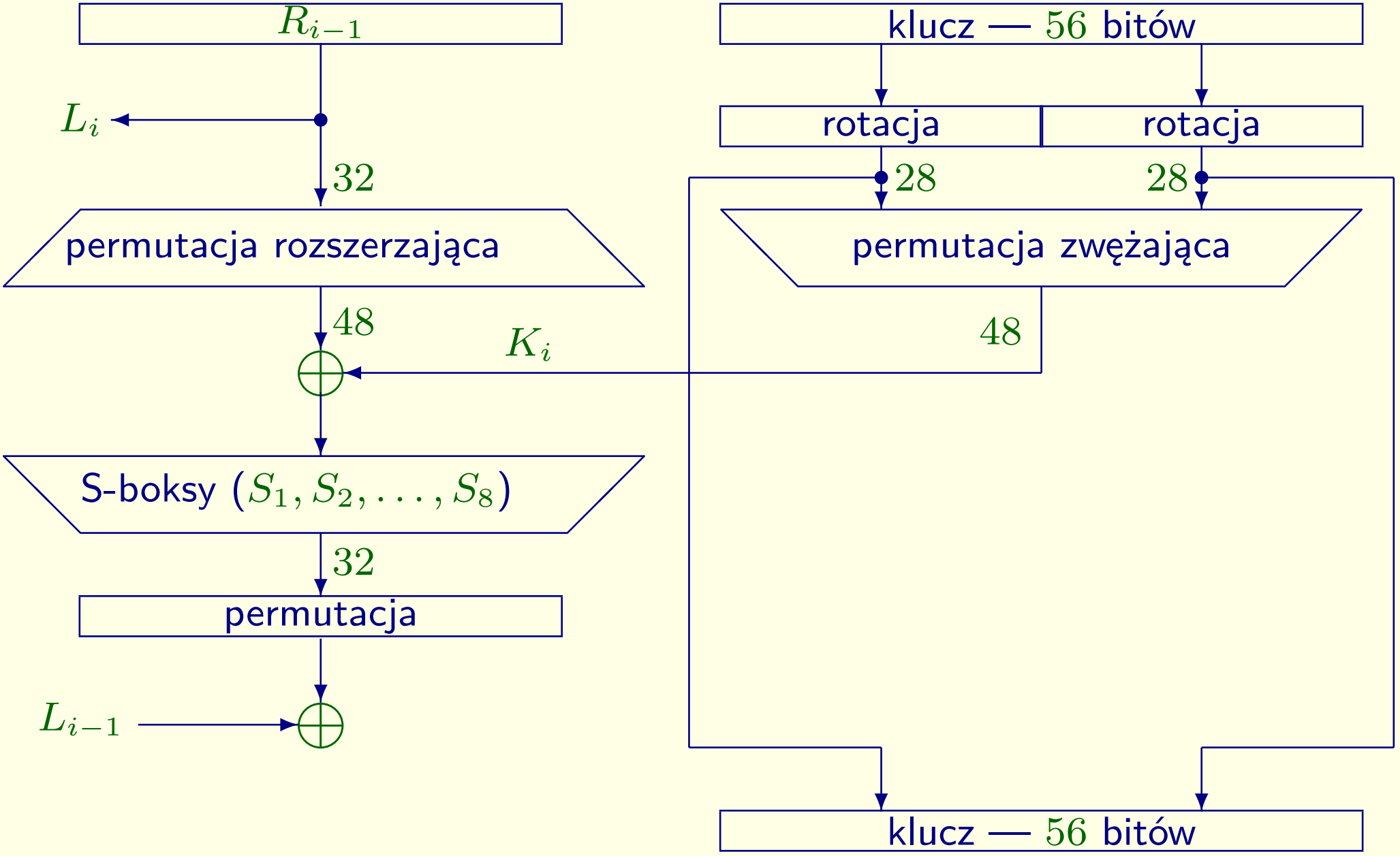
# Jedna runda DES'a



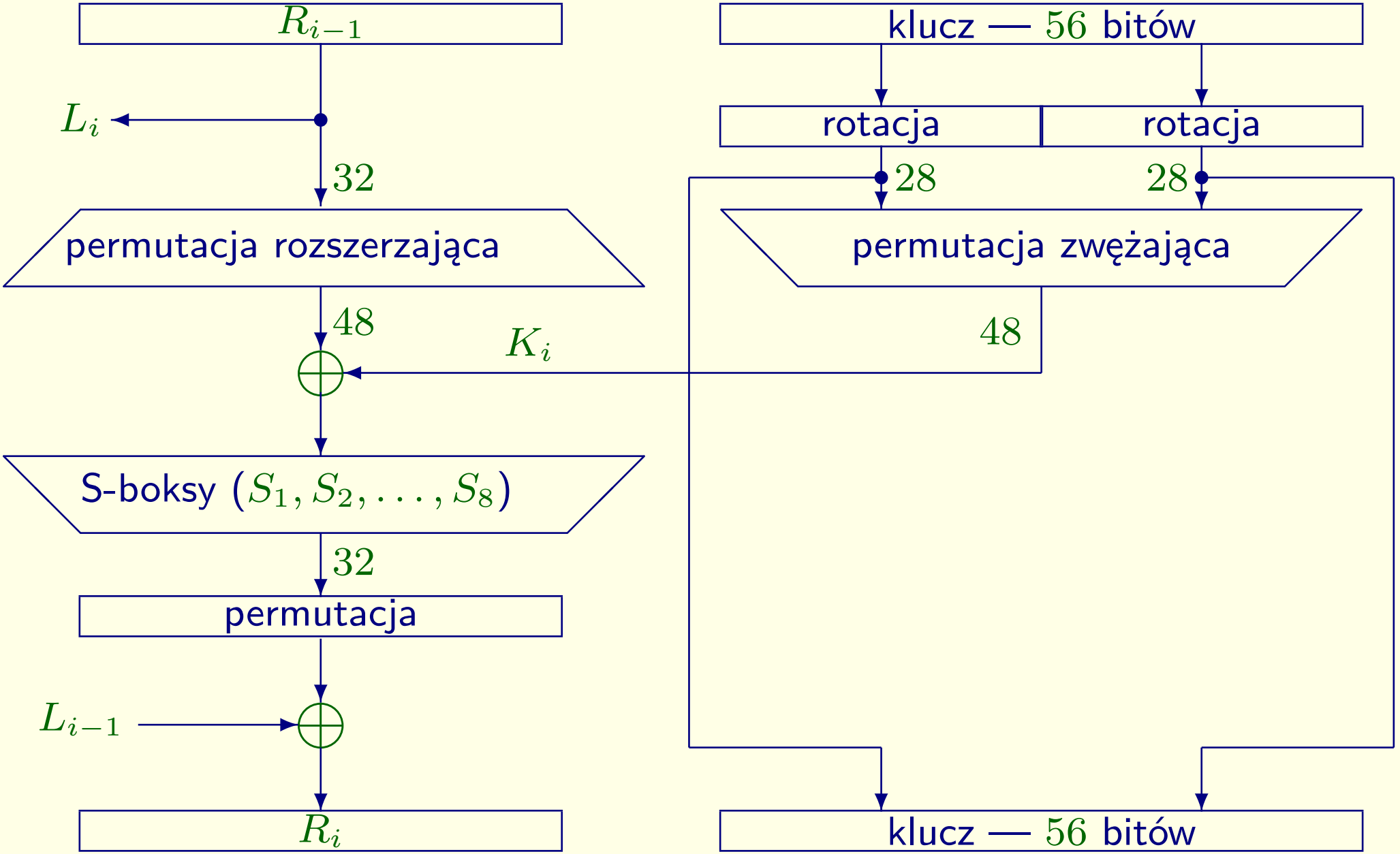
# Jedna runda DES'a



# Jedna runda DES'a



# Jedna runda DES'a



- szyfrowanie DES'em ($i$-ta runda)

  Jeśli $L_i$ i $R_i$ są lewą i prawą połową dla $i$-tej rundy, $K_i$ jest podkluczem dla tej rundy, to dla tej rundy mamy

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

- deszyfrowanie DES'em polega na przeprowadzeniu tych samych operacji co dla szyfrowania tylko podklucze występują w odwrotnej kolejności Ponieważ

$$\begin{aligned} R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i) \end{aligned}$$

  to znając $L_i$, $R_i$ oraz $K_i$ możemy obliczyć $L_{i-1}$ i $R_{i-1}$.

- szyfrowanie DES'em ($i$-ta runda)

  Jeśli $L_i$ i $R_i$ są lewą i prawą połową dla $i$-tej rundy, $K_i$ jest podkluczem dla tej rundy, to dla tej rundy mamy

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \end{aligned}$$

- deszyfrowanie DES'em polega na przeprowadzeniu tych samych operacji co dla szyfrowania tylko podklucze występują w odwrotnej kolejności Ponieważ

$$\begin{aligned} R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f(R_{i-1}, K_i) = R_i \oplus f(L_i, K_i) \end{aligned}$$

  to znając $L_i$, $R_i$ oraz $K_i$ możemy obliczyć $L_{i-1}$ i $R_{i-1}$.

# 5.2 Elementy DES

- permutacje początkowa i końcowa nie mają znaczenia kryptograficznego (ułatwiają operowanie danymi w bajtach)

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $IP$ | 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| | 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |
| $IP^{-1}$ | 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 | 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| | 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 | 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| | 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 | 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| | 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 | 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

Permutacja początkowa $IP$ i końcowa $IP^{-1}$ (tablice te czytamy od lewej do prawej, od góry w dół odczytując kolejne bity wyniku, a numery umieszczone na kolejnych pozycjach tabeli oznaczają numer bitu na wejściu, np. bit 58 wejścia jest pierwszym bitem wyjścia, itd.)

- generowanie podkluczy
  - z $64$ bitowego losowego klucza otrzymuje się $56$ bitowy ignorując co ósmy bit i dokonując permutacji $KP$

| $KP$ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| | 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| | 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Permutacja klucza

  - $56$ bitowy klucz dzieli się na dwie połowy po $28$ bitów
  - połowy są przesuwane cyklicznie w lewo o $1$ lub $2$ bity w zależności od rundy wg reguły

| runda | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| przesunięcie | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Przesunięcia połówek klucza

- generowanie podkluczy
  - z $64$ bitowego losowego klucza otrzymuje się $56$ bitowy ignorując co ósmy bit i dokonując permutacji $KP$

| $KP$ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| | 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| | 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Permutacja klucza

  - $56$ bitowy klucz dzieli się na dwie połowy po $28$ bitów
  - połowy są przesuwane cyklicznie w lewo o $1$ lub $2$ bity w zależności od rundy wg reguły

| runda | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| przesunięcie | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Przesunięcia połówek klucza

- generowanie podkluczy
  - z $64$ bitowego losowego klucza otrzymuje się $56$ bitowy ignorując co ósmy bit i dokonując permutacji $KP$

| $KP$ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| | 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| | 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Permutacja klucza

  - $56$ bitowy klucz dzieli się na dwie połowy po $28$ bitów
  - połowy są przesuwane cyklicznie w lewo o $1$ lub $2$ bity w zależności od rundy wg reguły

| runda | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| przesunięcie | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Przesunięcia połówek klucza

- generowanie podkluczy
  - z $64$ bitowego losowego klucza otrzymuje się $56$ bitowy ignorując co ósmy bit i dokonując permutacji $KP$

| $KP$ | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| | 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| | 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

Permutacja klucza

  - $56$ bitowy klucz dzieli się na dwie połowy po $28$ bitów
  - połowy są przesuwane cyklicznie w lewo o $1$ lub $2$ bity w zależności od rundy wg reguły

| runda | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| przesunięcie | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Przesunięcia połówek klucza

- permutacja z kompresją $CP$ (permutowany wybór) daje $48$ bitów podklucza

| $CP$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 | 15 | 6 | 21 | 10 |
| | 23 | 19 | 12 | 4 | 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 |
| | 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 | 51 | 45 | 33 | 48 |
| | 44 | 49 | 39 | 56 | 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

Permutacja zwężająca

- permutacja z rozszerzeniem rozszerza $32$ bity $R_i$ do $48$ bitów

| $EP$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 32 | 1 | 2 | 3 | 4 | 5 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 12 | 13 | 14 | 15 | 16 | 17 |
| | 16 | 17 | 18 | 19 | 20 | 21 | 20 | 21 | 22 | 23 | 24 | 25 |
| | 24 | 25 | 26 | 27 | 28 | 29 | 28 | 29 | 30 | 31 | 32 | 1 |

Permutacja z rozszerzeniem

- permutacja z kompresją $CP$ (permutowany wybór) daje $48$ bitów podklucza

| $CP$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 | 15 | 6 | 21 | 10 |
| | 23 | 19 | 12 | 4 | 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 |
| | 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 | 51 | 45 | 33 | 48 |
| | 44 | 49 | 39 | 56 | 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

Permutacja zwężająca

- permutacja z rozszerzeniem rozszerza $32$ bity $R_i$ do $48$ bitów

| $EP$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 32 | 1 | 2 | 3 | 4 | 5 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 12 | 13 | 14 | 15 | 16 | 17 |
| | 16 | 17 | 18 | 19 | 20 | 21 | 20 | 21 | 22 | 23 | 24 | 25 |
| | 24 | 25 | 26 | 27 | 28 | 29 | 28 | 29 | 30 | 31 | 32 | 1 |

Permutacja z rozszerzeniem

- $S$-boksy
  - wynik operacji xor na rozszerzonym $R_i$ i $K_i$ dzielony jest na $8$ części po $6$ bitów, z których każda przechodzi do oddzielnego $S$-boksu $(S_1, \ldots, S_8)$
  - $6$ bitów wchodzących do $S$-boksu przekształcanych jest w $4$ bity wyjściowe w specjalny sposób pierwszy i ostatni bit $6$ bitów wejściowych daje liczbę dwubitową od $0 - 3$ oznaczającą wiersz $S$-boksu, zaś bity $2 - 5$ dają liczbę $4$-bitową od $0 - 15$, która odpowiada kolumnie tabeli.

- $S$-boksy
  - wynik operacji xor na rozszerzonym $R_i$ i $K_i$ dzielony jest na $8$ części po $6$ bitów, z których każda przechodzi do oddzielnego $S$-boksu $(S_1, \ldots, S_8)$
  - $6$ bitów wchodzących do $S$-boksu przekształcanych jest w $4$ bity wyjściowe w specjalny sposób pierwszy i ostatni bit $6$ bitów wejściowych daje liczbę dwubitową od $0 - 3$ oznaczającą wiersz $S$-boksu, zaś bity $2 - 5$ dają liczbę $4$-bitową od $0 - 15$, która odpowiada kolumnie tabeli.

- $S$-boksy
  - wynik operacji xor na rozszerzonym $R_i$ i $K_i$ dzielony jest na $8$ części po $6$ bitów, z których każda przechodzi do oddzielnego $S$-boksu $(S_1, \ldots, S_8)$
  - $6$ bitów wchodzących do $S$-boksu przekształcanych jest w $4$ bity wyjściowe w specjalny sposób pierwszy i ostatni bit $6$ bitów wejściowych daje liczbę dwubitową od $0-3$ oznaczającą wiersz $S$-boksu, zaś bity $2-5$ dają liczbę $4$-bitową od $0-15$, która odpowiada kolumnie tabeli.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |
| $S_2$ | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |
| $S_3$ | 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| | 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| | 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| | 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |
| $S_4$ | 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| | 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| | 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| | 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |
| $S_5$ | 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| | 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| | 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| | 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |
| $S_6$ | 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| | 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| | 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| | 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |
| $S_7$ | 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| | 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| | 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| | 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |
| $S_8$ | 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| | 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| | 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| | 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

- Weźmy dla przykładu $S$-boks $S_1$

| $S_1$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

$S$-boks $S_1$

Jeśli na wejściu mamy ciąg bitów $1\underline{10011}1$, to skrajne bity $1$ i $6$ dają liczbę $11_2 = 3_{10}$, zaś bity $2-5$ dają liczbę $1001_2 = 9_{10}$, na przecięciu wiersza $3$ i kolumny $9$ boksu $S_1$ mamy liczbę $11_{10} = 1011_2$ (wiersze i kolumny liczymy od zera). W wyniku otrzymujemy ciąg bitów $1011$.

- wyniki z $8$ $S$-boksów są łączone w $32$ bitową liczbę
- na tych $32$ bitach dokonuje się permutacji $P$ wg poniższej tablicy oraz operacji xor z $32$ bitami lewej połowy otrzymując nową prawą połowę, która przekazywana jest do następnej rundy

| $P$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 | 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| | 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 | 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

Pertmutacja ($P$-boks)

- wyniki z $8$ $S$-boksów są łączone w $32$ bitową liczbę
- na tych $32$ bitach dokonuje się permutacji $P$ wg poniższej tablicy oraz operacji xor z $32$ bitami lewej połowy otrzymując nową prawą połowę, która przekazywana jest do następnej rundy

| $P$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 | 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| | 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 | 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

Pertmutacja ($P$-boks)

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$

## 5.3 Trzykrotny DES

Rozszerzenie algorytmu DES, w którym stosuje się dwa klucze $K_1$ i $K_2$

- Szyfrowanie
  1. wiadomość szyfrowana jest kluczem $K_1$
  2. wynik kroku 1. deszyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest ponownie szyfrowany kluczem $K_1$
- Deszyfrowanie
  1. kryptogram deszyfrowany jest kluczem $K_1$
  2. wynik kroku 1. szyfrowany jest kluczem $K_2$
  3. wynik kroku 2. jest powtórnie deszyfrowany kluczem $K_1$