Kryptografia

z elementami kryptografii kwantowej

Ryszard Tanaś

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

Wykład 3

Spis treści

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6 Tryby szyfrowania: szyfrowanie blokowe

6.1 ECB — Electronic Codebook (Elektroniczna książka kodowa)

- Tekst jawny dzielony jest na bloki o długości 64 bity
- Każdy blok jest oddzielnie szyfrowany tym samym kluczem
- ECB — zalety
  - Utrata lub uszkodzenie pojedynczych bloków nie ma wpływu na możliwość deszyfrowania pozostałych
  - Nadaje się do szyfrowania baz danych
- ECB — wady
  - Możliwa jest modyfikacja kryptogramu bez znajomości klucza

6.2 CBC — Cipher Block Chaining (Wiązanie bloków)

- Szyfrowanie kolejnego bloku zależy od wyniku szyfrowania poprzedniego bloku
- Taki sam blok tekstu jawnego jest w różnych miejscach szyfrowany inaczej
- Kolejny blok tekstu jawnego jest poddawany operacji XOR z kryptogramem poprzedniego bloku.

6.2 CBC — Cipher Block Chaining (Wiązanie bloków)

- Szyfrowanie kolejnego bloku zależy od wyniku szyfrowania poprzedniego bloku
- Taki sam blok tekstu jawnego jest w różnych miejscach szyfrowany inaczej
- Kolejny blok tekstu jawnego jest poddawany operacji XOR z kryptogramem poprzedniego bloku.

6.2 CBC — Cipher Block Chaining (Wiązanie bloków)

- Szyfrowanie kolejnego bloku zależy od wyniku szyfrowania poprzedniego bloku
- Taki sam blok tekstu jawnego jest w różnych miejscach szyfrowany inaczej
- Kolejny blok tekstu jawnego jest poddawany operacji XOR z kryptogramem poprzedniego bloku.

6.2 CBC — Cipher Block Chaining (Wiązanie bloków)

- Szyfrowanie kolejnego bloku zależy od wyniku szyfrowania poprzedniego bloku
- Taki sam blok tekstu jawnego jest w różnych miejscach szyfrowany inaczej
- Kolejny blok tekstu jawnego jest poddawany operacji XOR z kryptogramem poprzedniego bloku.

Matematycznie wygląda to następująco:

Matematycznie wygląda to następująco:

$$C_1 \quad = \quad E_K(M_1 \oplus I)$$

Matematycznie wygląda to następująco:

$$\begin{aligned} C_1 &= E_K(M_1 \oplus I) \\ C_i &= E_K(M_i \oplus C_{i-1}) \end{aligned}$$

Matematycznie wygląda to następująco:

$$
\begin{aligned}
C_1 &= E_K(M_1 \oplus I) \\
C_i &= E_K(M_i \oplus C_{i-1}) \\
\\
M_1 &= D_K(C_1 \oplus I)
\end{aligned}
$$

Matematycznie wygląda to następująco:

$$
\begin{aligned}
C_1 &= E_K(M_1 \oplus I) \\
C_i &= E_K(M_i \oplus C_{i-1}) \\
\\
M_1 &= D_K(C_1 \oplus I) \\
M_i &= D_K(C_i) \oplus C_{i-1}
\end{aligned}
$$

Matematycznie wygląda to następująco:

$$
\begin{aligned}
C_1 &= E_K(M_1 \oplus I) \\
C_i &= E_K(M_i \oplus C_{i-1}) \\
\\
M_1 &= D_K(C_1 \oplus I) \\
M_i &= D_K(C_i) \oplus C_{i-1}
\end{aligned}
$$

gdzie M_i jest i-tym blokiem wiadomości, C_i jest i-tym blokiem kryptogramu, zaś I jest losowym ciągiem bitów, który jest przesyłany bez szyfrowania

Matematycznie wygląda to następująco:

$$\begin{aligned} C_1 &= E_K(M_1 \oplus I) \\ C_i &= E_K(M_i \oplus C_{i-1}) \end{aligned}$$

$$\begin{aligned} M_1 &= D_K(C_1 \oplus I) \\ M_i &= D_K(C_i) \oplus C_{i-1} \end{aligned}$$

gdzie M_i jest i-tym blokiem wiadomości, C_i jest i-tym blokiem kryptogramu, zaś I jest losowym ciągiem bitów, który jest przesyłany bez szyfrowania

$$D_K(C_i) \oplus C_{i-1} = M_i \oplus C_{i-1} \oplus C_{i-1} = M_i$$

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

- CBC — zalety
  - takie same bloki tekstu jawnego mają różne kryptogramy
  - zmiana bitu (przekłamanie) wewnątrz jednego bloku prowadzi do zmiany tekstu po deszyfrowaniu tylko w danym bloku i następnym
- CBC — wady
  - nie można usunąć żadnego bloku z kryptogramu; nie nadaje się do szyfrowania baz danych
  - nieodporny na zakłócenia (dodatkowy bit lub utrata jednego bitu psują dalszy przekaz)

6.3 CFB — Cipher Feedback (Szyfrowanie ze sprzężeniem zwrotnym kryptogramu)

- Szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Tryb ważny w zastosowaniach sieciowych, np. komunikacja pomiędzy klawiaturą i serwerem.
- Istotnym elementem CFB jest rejestr przesuwający.

6.3 CFB — Cipher Feedback (Szyfrowanie ze sprzężeniem zwrotnym kryptogramu)

- Szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Tryb ważny w zastosowaniach sieciowych, np. komunikacja pomiędzy klawiaturą i serwerem.
- Istotnym elementem CFB jest rejestr przesuwający.

6.3 CFB — Cipher Feedback (Szyfrowanie ze sprzężeniem zwrotnym kryptogramu)

- Szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Tryb ważny w zastosowaniach sieciowych, np. komunikacja pomiędzy klawiaturą i serwerem.
- Istotnym elementem CFB jest rejestr przesuwający.

6.3 CFB — Cipher Feedback (Szyfrowanie ze sprzężeniem zwrotnym kryptogramu)

- Szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Tryb ważny w zastosowaniach sieciowych, np. komunikacja pomiędzy klawiaturą i serwerem.
- Istotnym elementem CFB jest rejestr przesuwający.

6.3.1 Działanie CFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.3.1 Działanie CFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.3.1 Działanie CFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.3.1 Działanie CFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.3.1 Działanie CFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.3.1 Działanie CFB

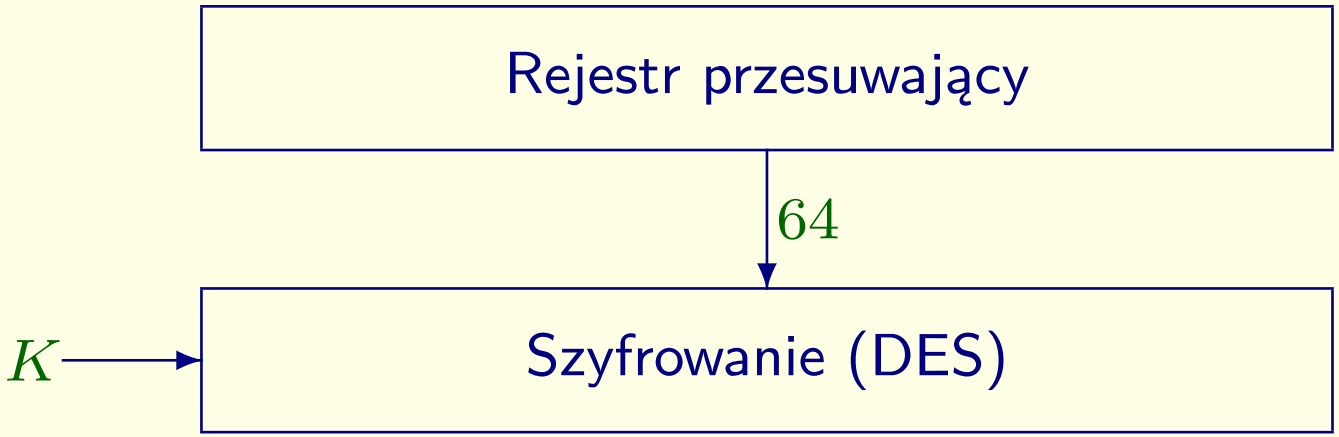
- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- C_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

CFB — szyfrowanie

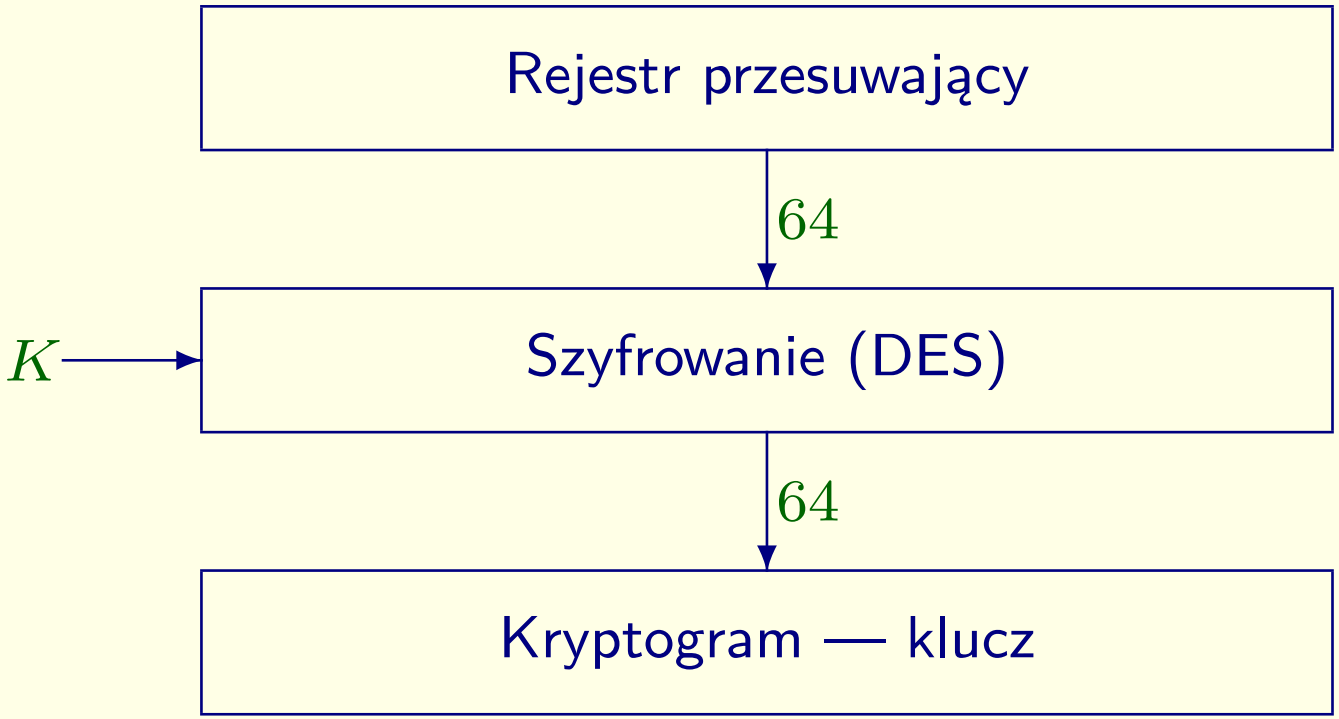
CFB — szyfrowanie

Rejestr przesuwający

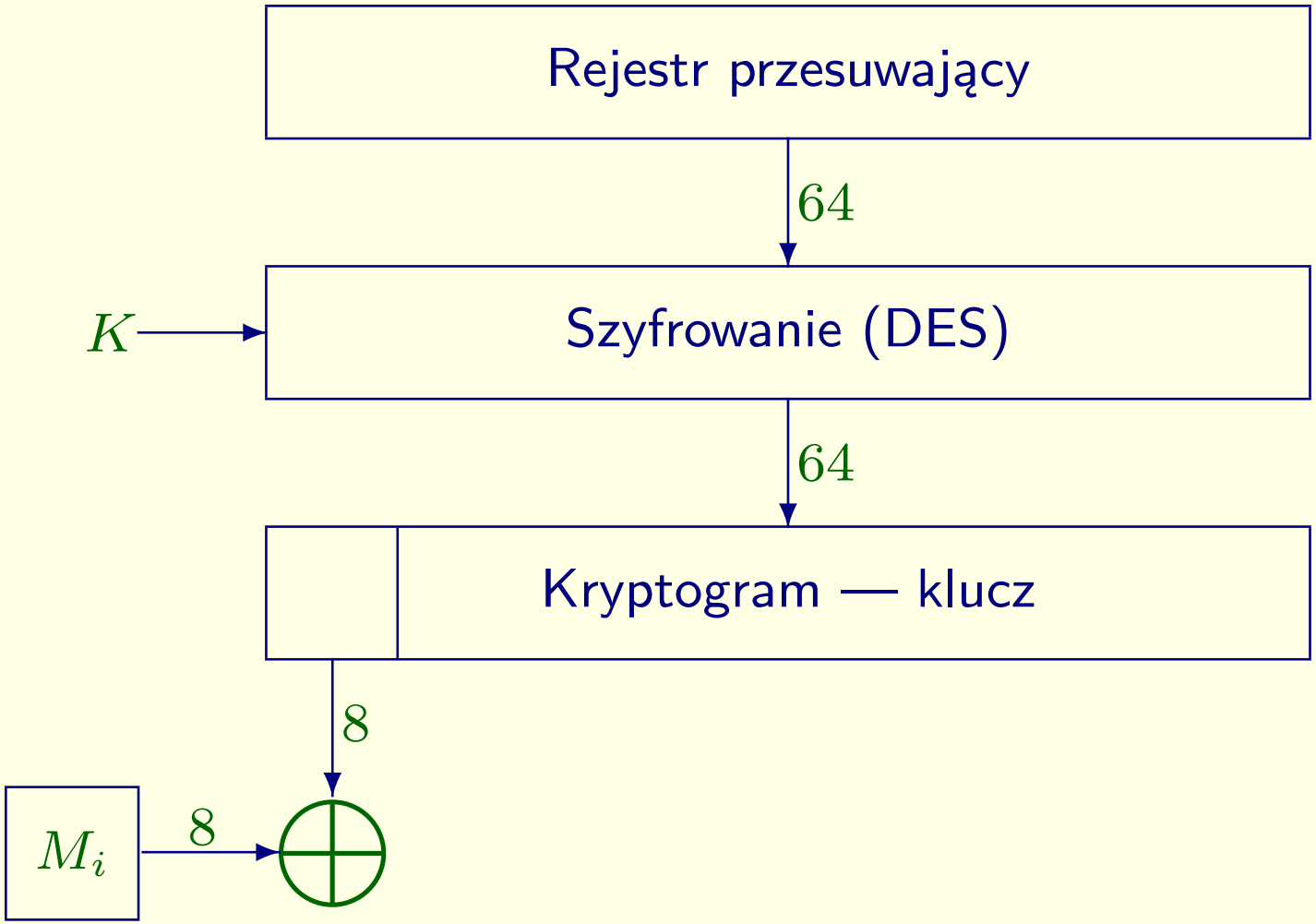
CFB — szyfrowanie



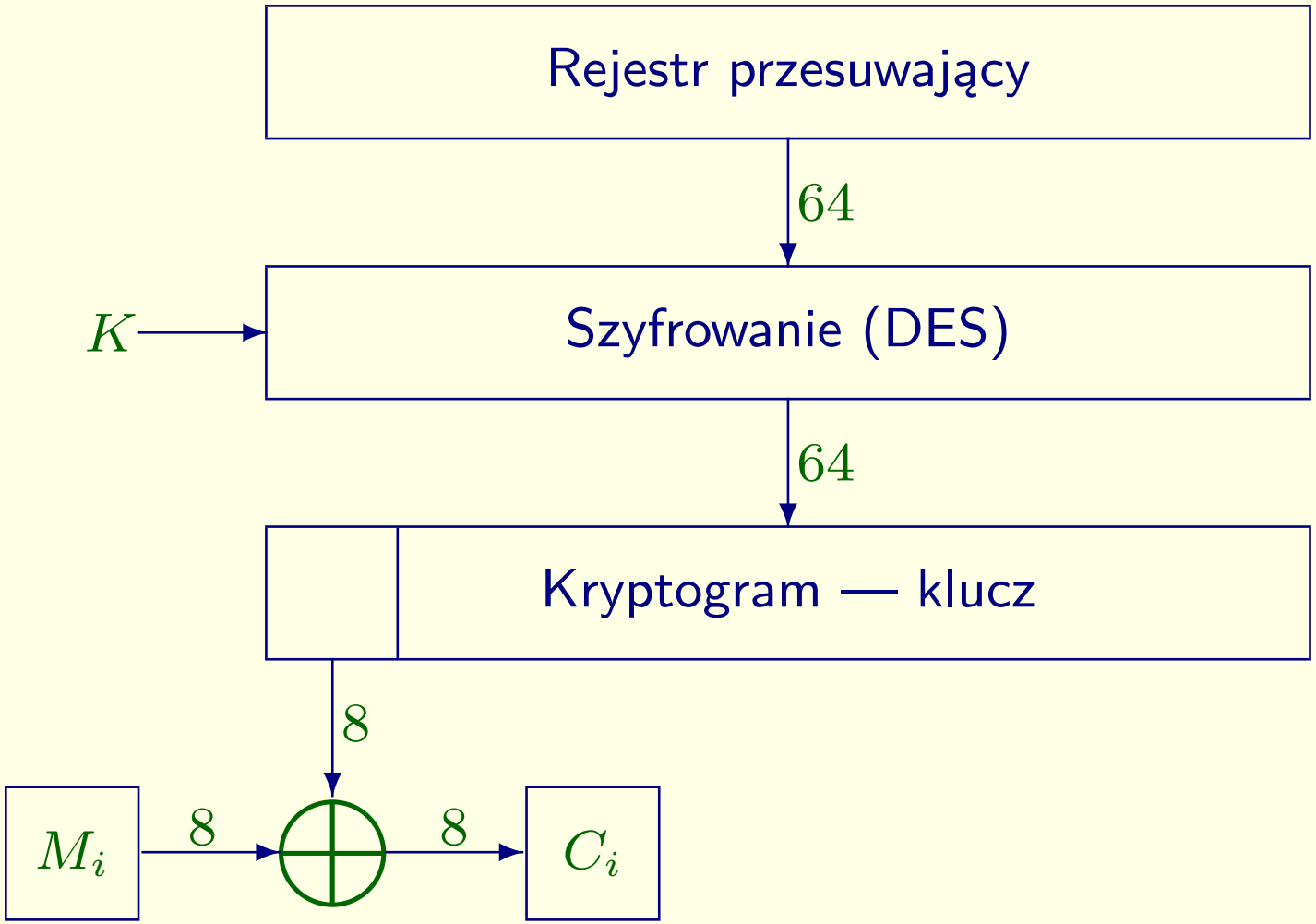
CFB — szyfrowanie



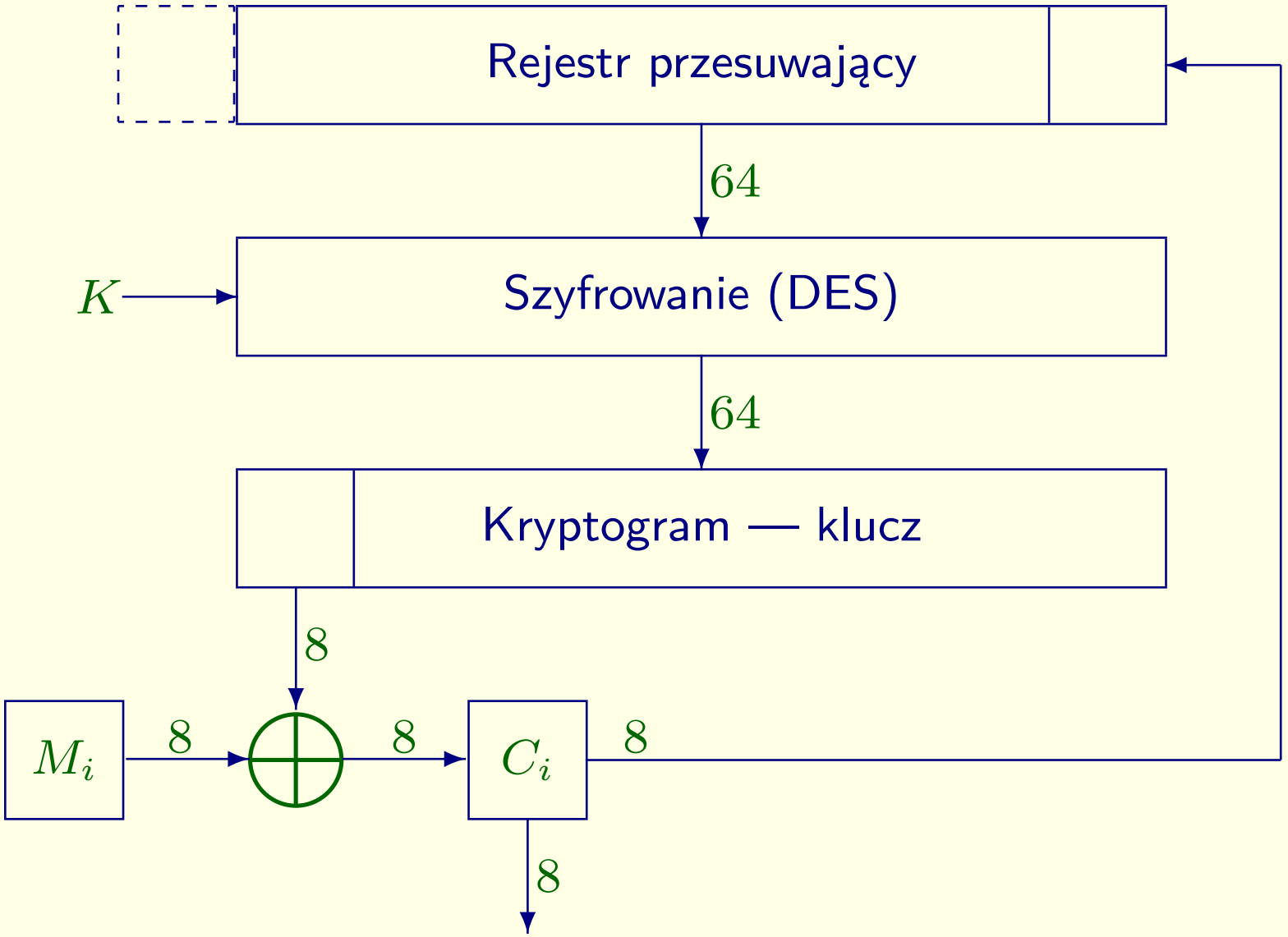
CFB — szyfrowanie



CFB — szyfrowanie



CFB — szyfrowanie

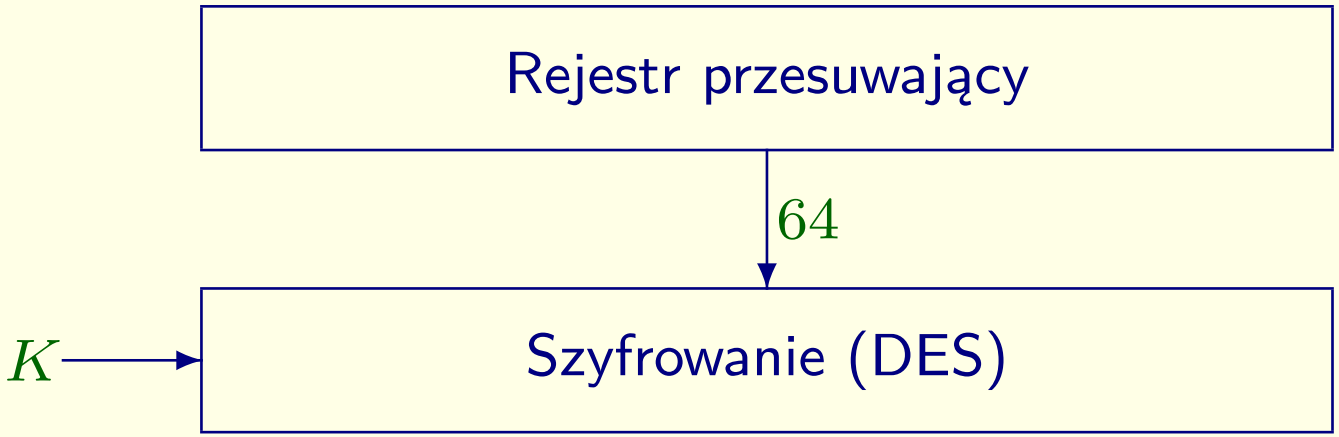


CFB — deszyfrowanie

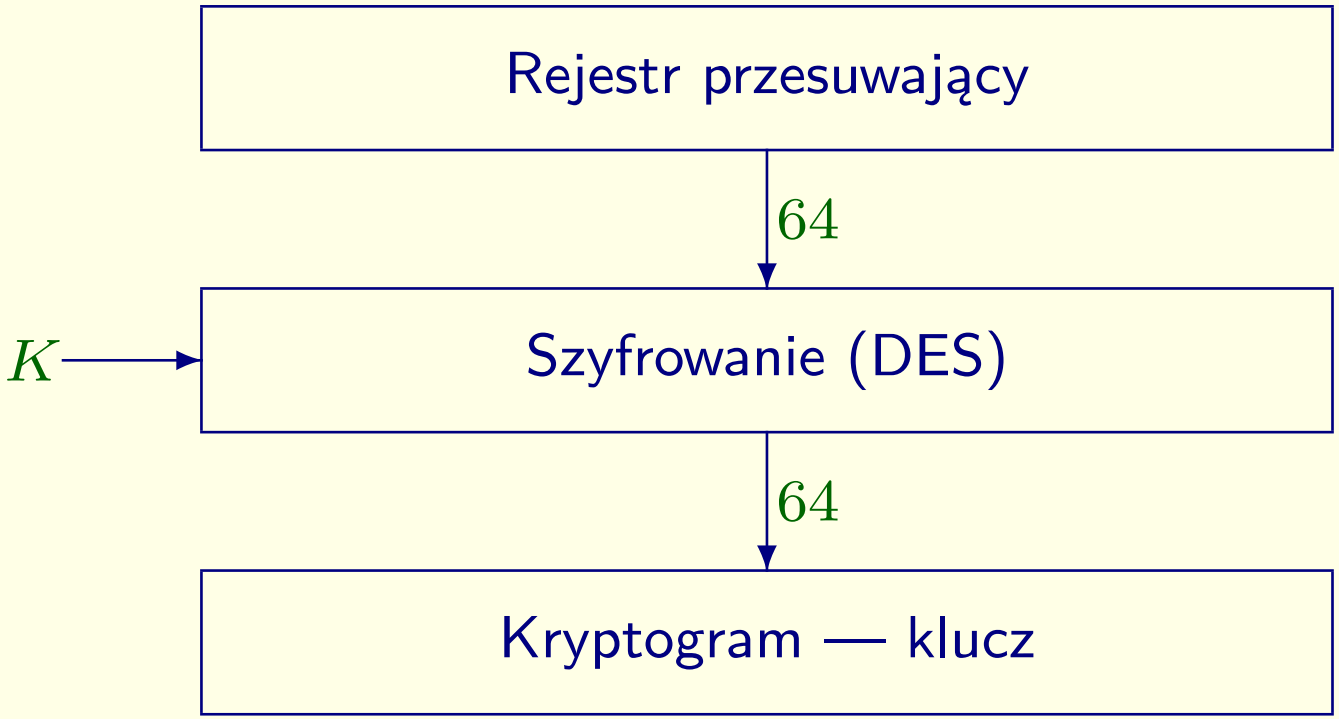
CFB — deszyfrowanie

Rejestr przesuwający

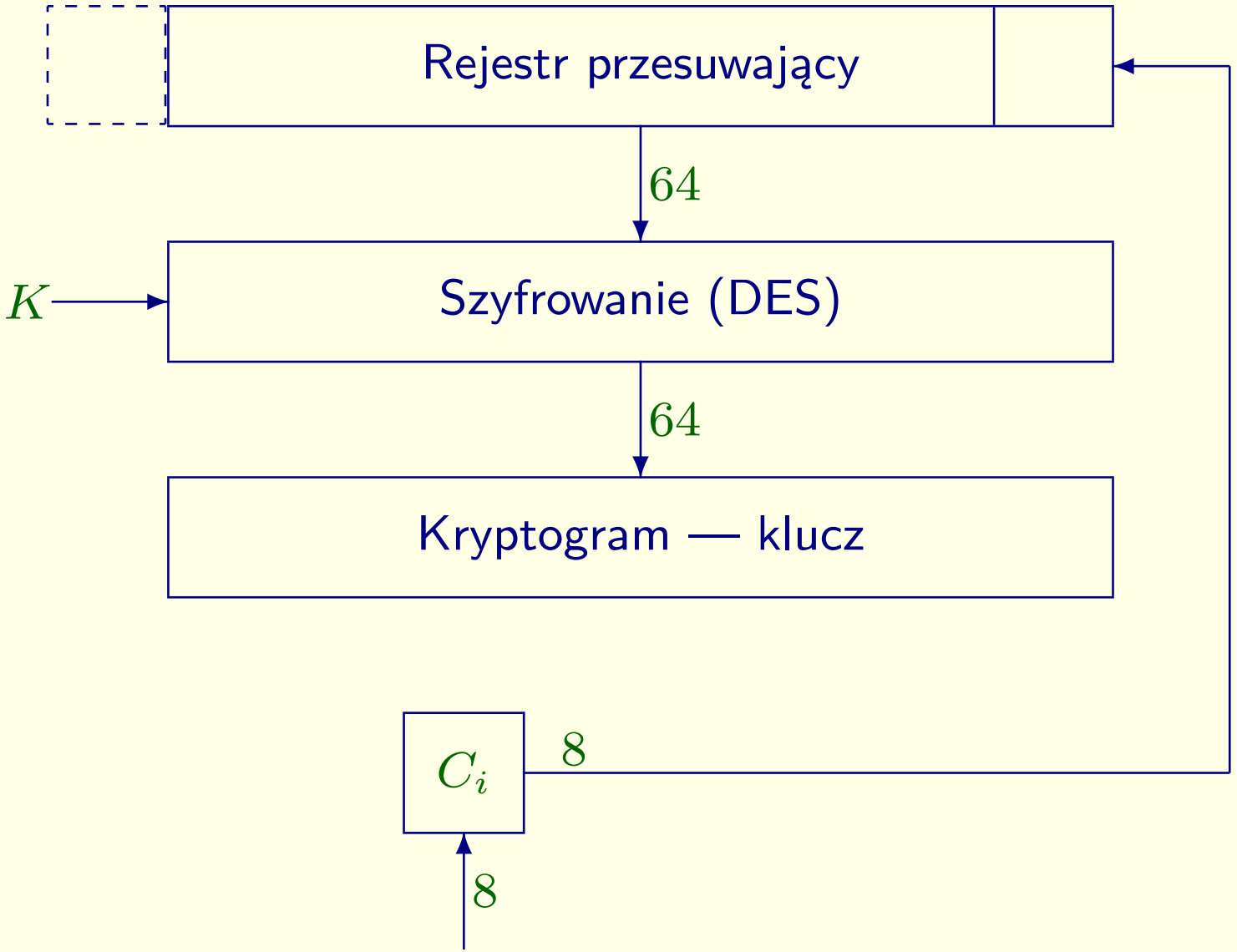
CFB — deszyfrowanie



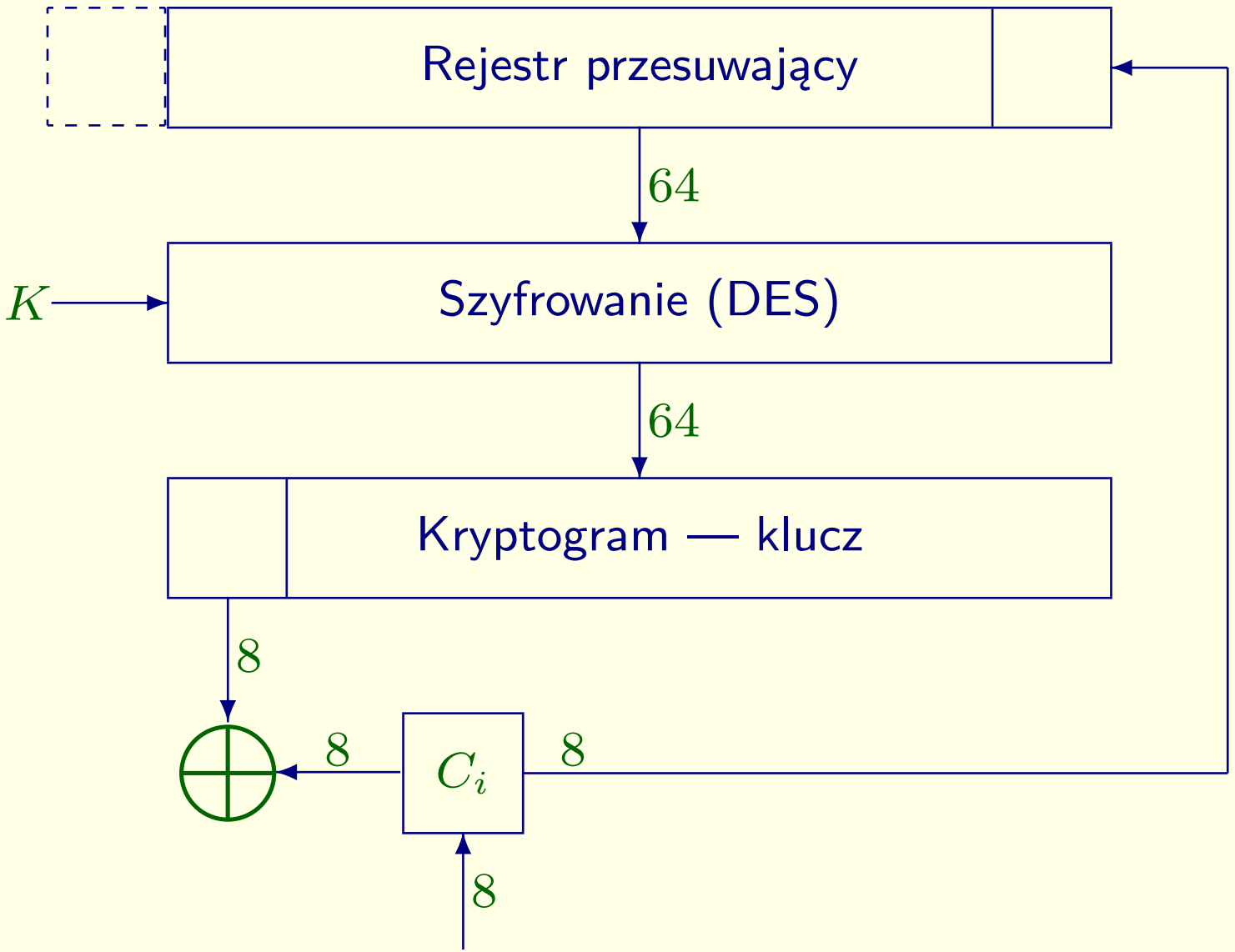
CFB — deszyfrowanie



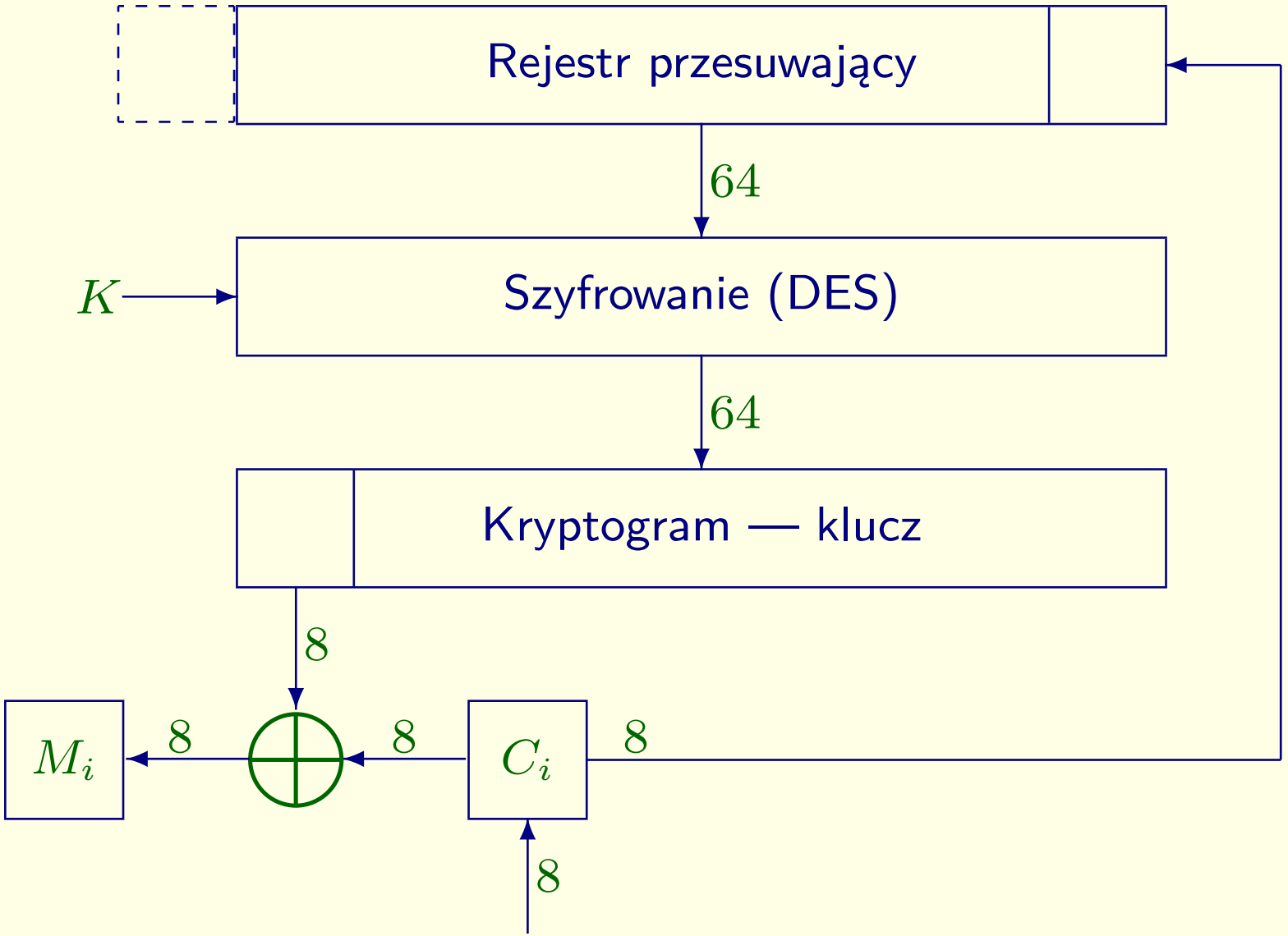
CFB — deszyfrowanie



CFB — deszyfrowanie



CFB — deszyfrowanie



6.4 OFB — Output Feedback (Szyfrowanie ze sprzężeniem zwrotnym wyjściowym)

- Podobnie jak w trybie CFB, szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Podobnie jak w trybie CFB, istotnym elementem OFB jest rejestr przesuwający.

6.4 OFB — Output Feedback (Szyfrowanie ze sprzężeniem zwrotnym wyjściowym)

- Podobnie jak w trybie CFB, szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Podobnie jak w trybie CFB, istotnym elementem OFB jest rejestr przesuwający.

6.4 OFB — Output Feedback (Szyfrowanie ze sprzężeniem zwrotnym wyjściowym)

- Podobnie jak w trybie CFB, szyfrowaniu podlegają jednostki mniejsze niż blok (64 bity), np. jeden znak ASCII (1 bajt $= 8$ bitów).
- Podobnie jak w trybie CFB, istotnym elementem OFB jest rejestr przesuwający.

6.5 Działanie OFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.5 Działanie OFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.5 Działanie OFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.5 Działanie OFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.5 Działanie OFB

- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

6.5 Działanie OFB

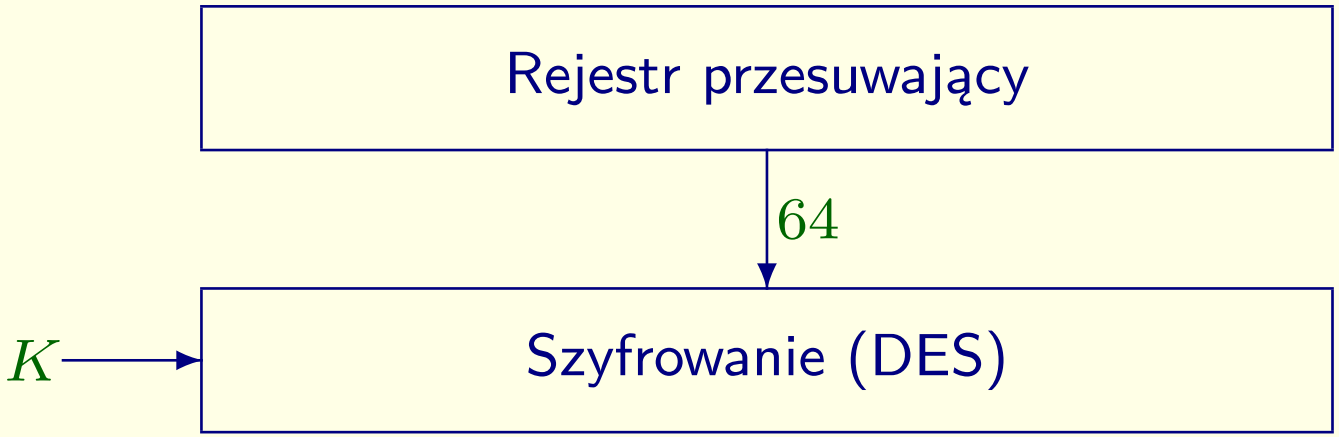
- Na początku rejestr przesuwający zawiera losowy ciąg 64 bitów
- Zawartość rejestru przesuwającego jest szyfrowana za pomocą klucza K np. algorytmem DES
- Pierwszych 8 bitów kryptogramu K_i jest dodawane modulo 2 (operacja xor) z 8 bitami reprezentującymi literę wiadomości (M_i) dając kryptogram C_i przesyłany do odbiorcy
- K_i jednocześnie przesyłane jest do rejestru przesuwającego zajmując ostatnie 8 bitów i przesuwając pozostałe bity o 8 pozycji w lewo; przesunięcie to nie jest cykliczne, tzn. pierwszych 8 bitów jest usuwanych
- Przy deszyfrowaniu rola wejścia i wyjścia zostaje zamieniona

OFB — szyfrowanie

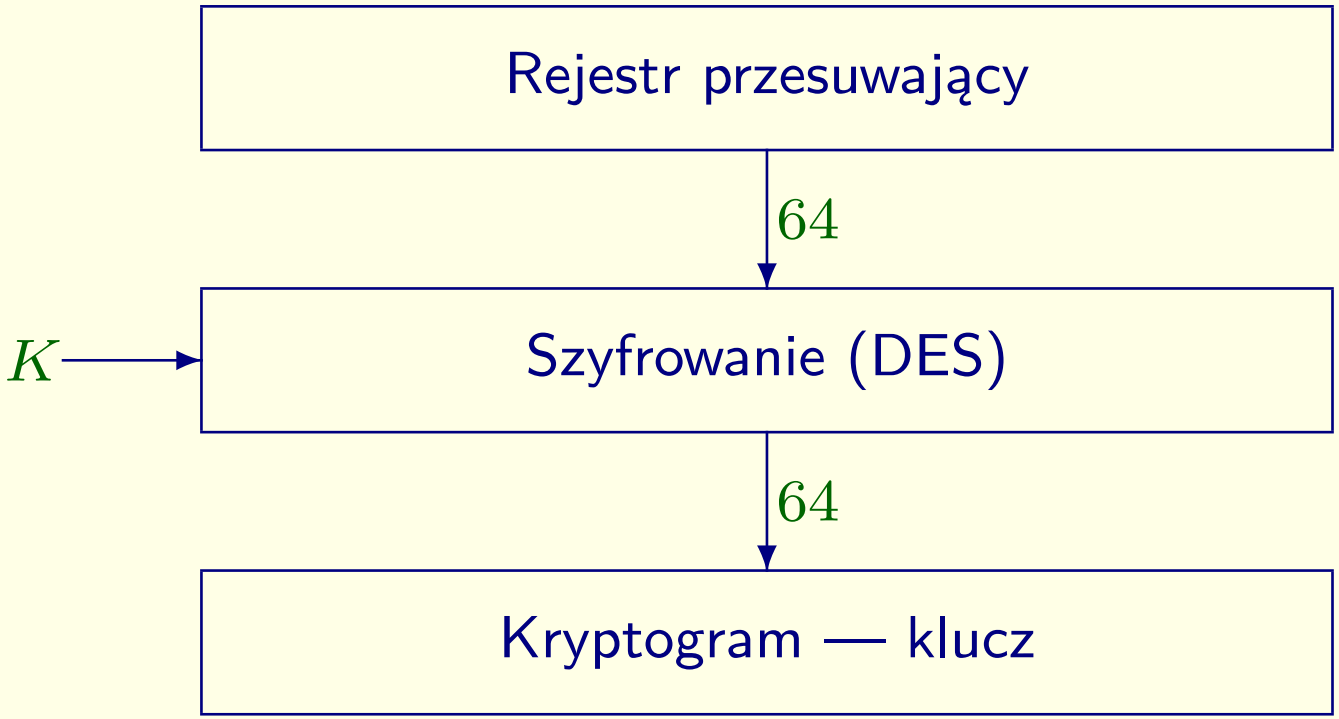
OFB — szyfrowanie

Rejestr przesuwający

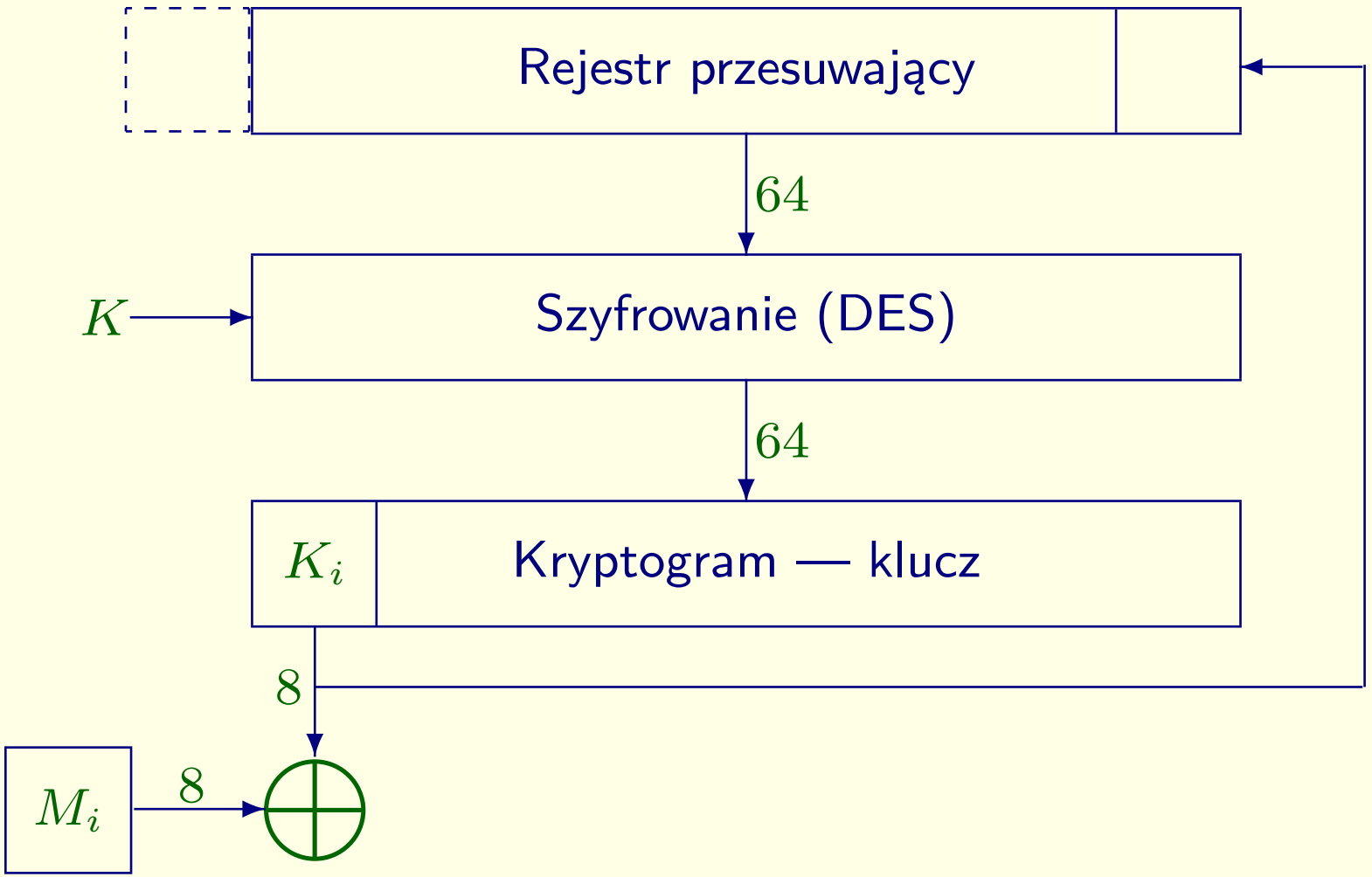
OFB — szyfrowanie



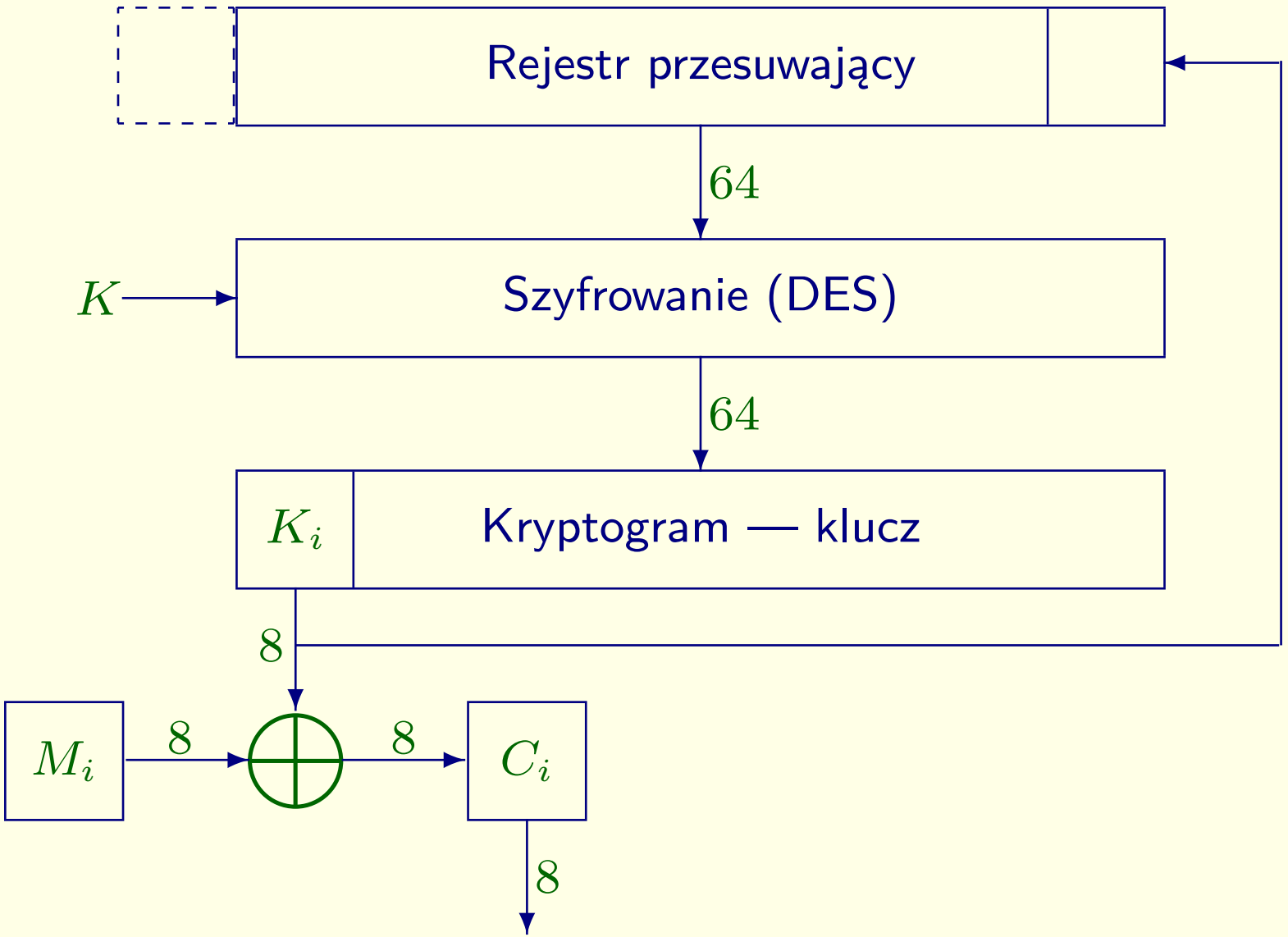
OFB — szyfrowanie



OFB — szyfrowanie



OFB — szyfrowanie

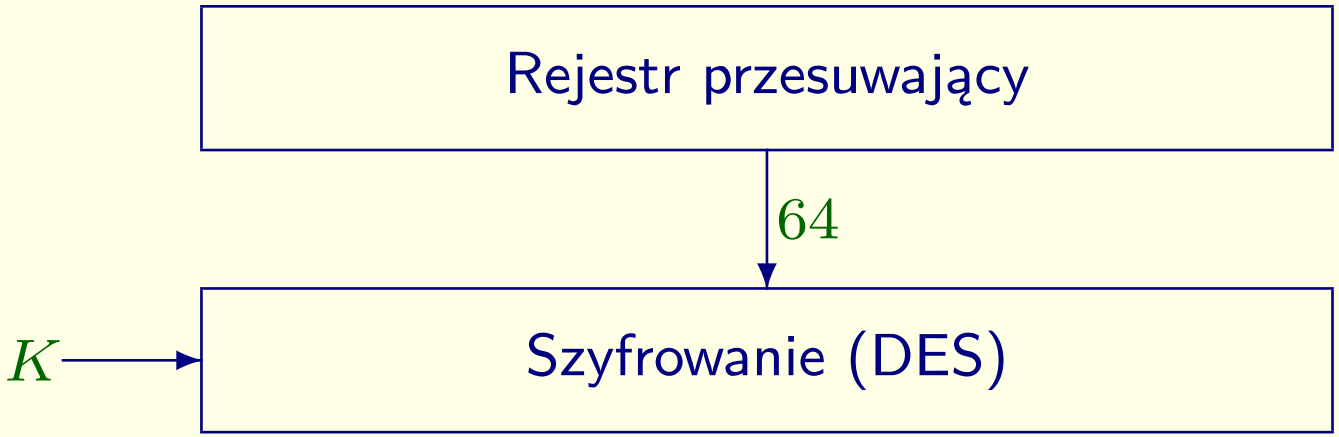


OFB — deszyfrowanie

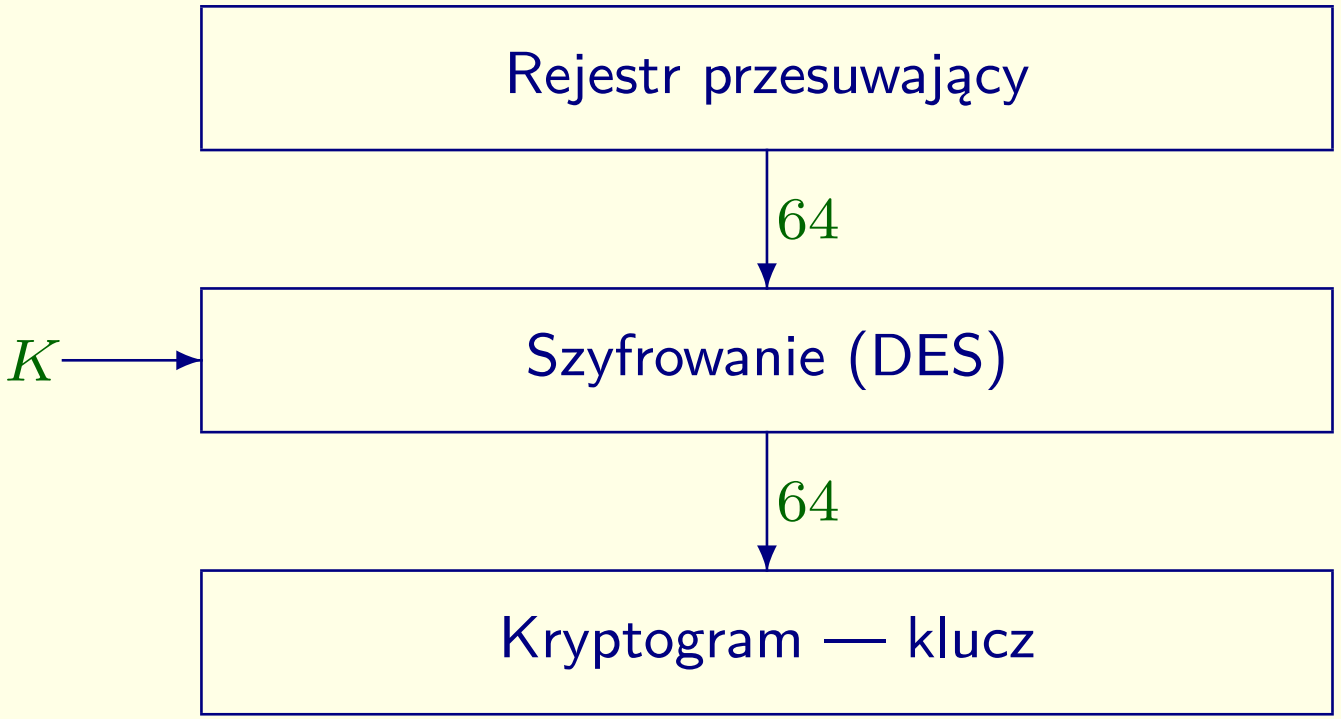
OFB — deszyfrowanie

Rejestr przesuwający

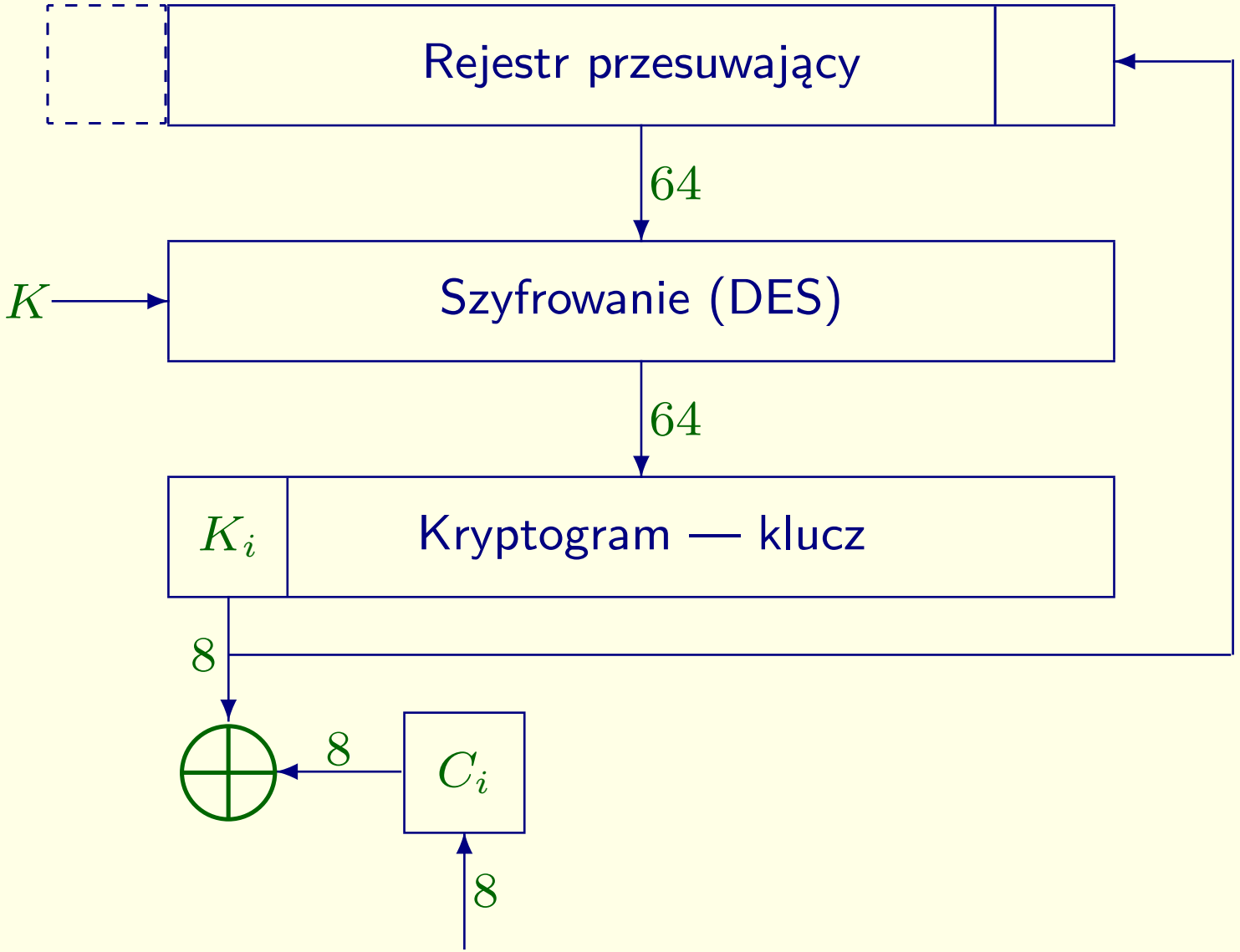
OFB — deszyfrowanie



OFB — deszyfrowanie



OFB — deszyfrowanie



OFB — deszyfrowanie