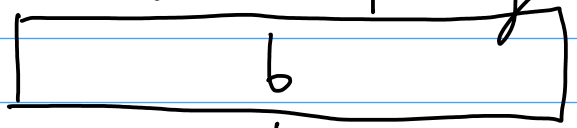
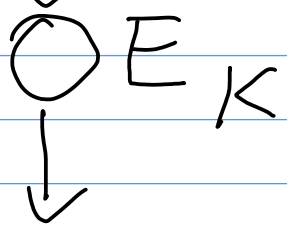
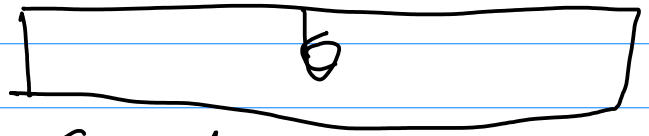
Szyfry blokowe

tekst jawny



64 bity (lub 128)

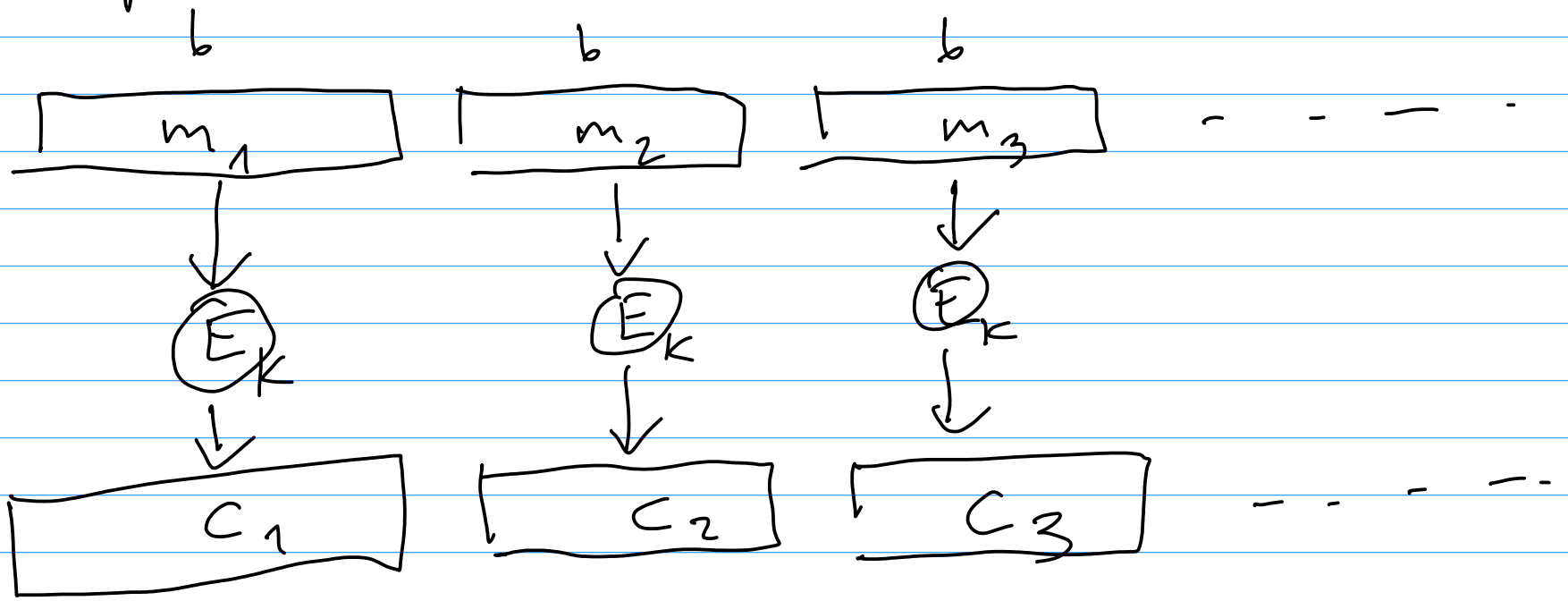




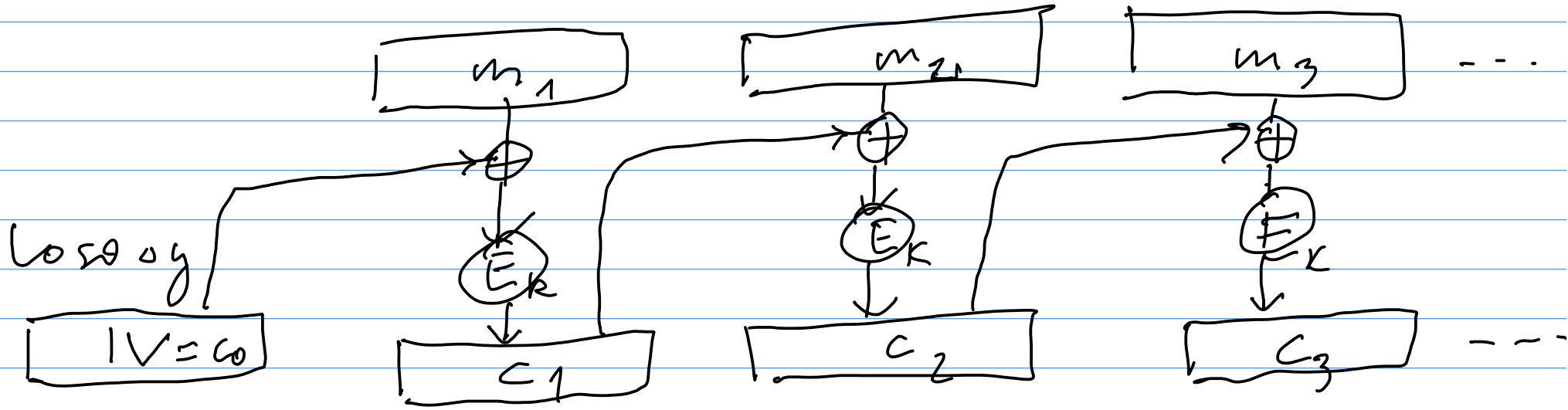
64 bity

szyfrogram

Τρόπος ECB (Electronic Codebook)



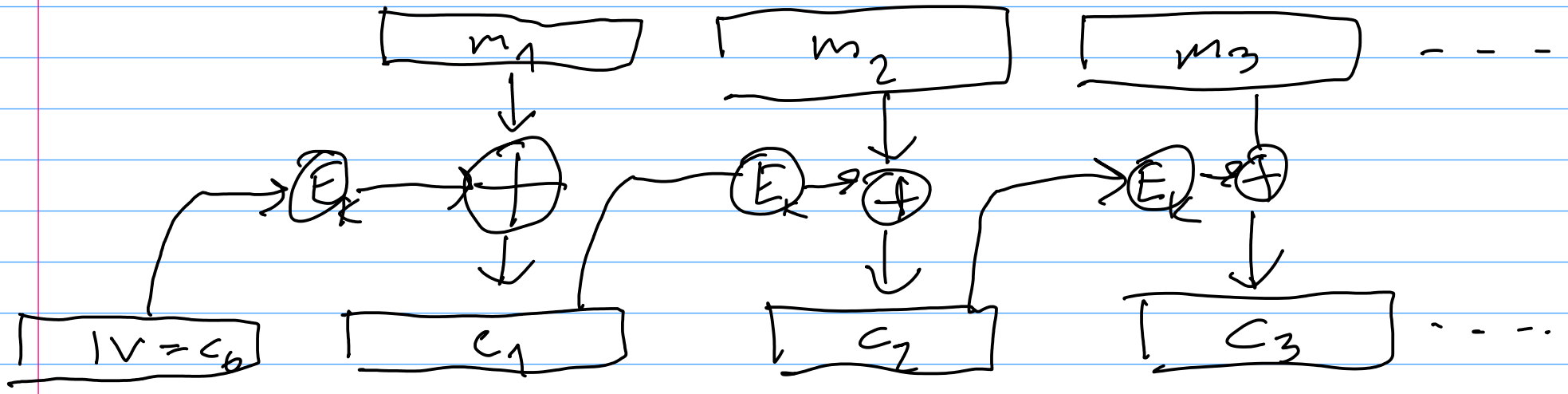
Tryb CBC (Cipher Block Chaining)



Initial vector

$$c_i = E_K(m_i \oplus c_{i-1}) \qquad m_i = D_K(c_i) \oplus c_{i-1}$$

Tryp CFB (Cipher Feedback)



$$c_i = E_k(c_{i-1}) \oplus m_i \qquad m_i = E_k(c_{i-1}) \oplus c_i$$

Tryb OFB (Output Feedback)

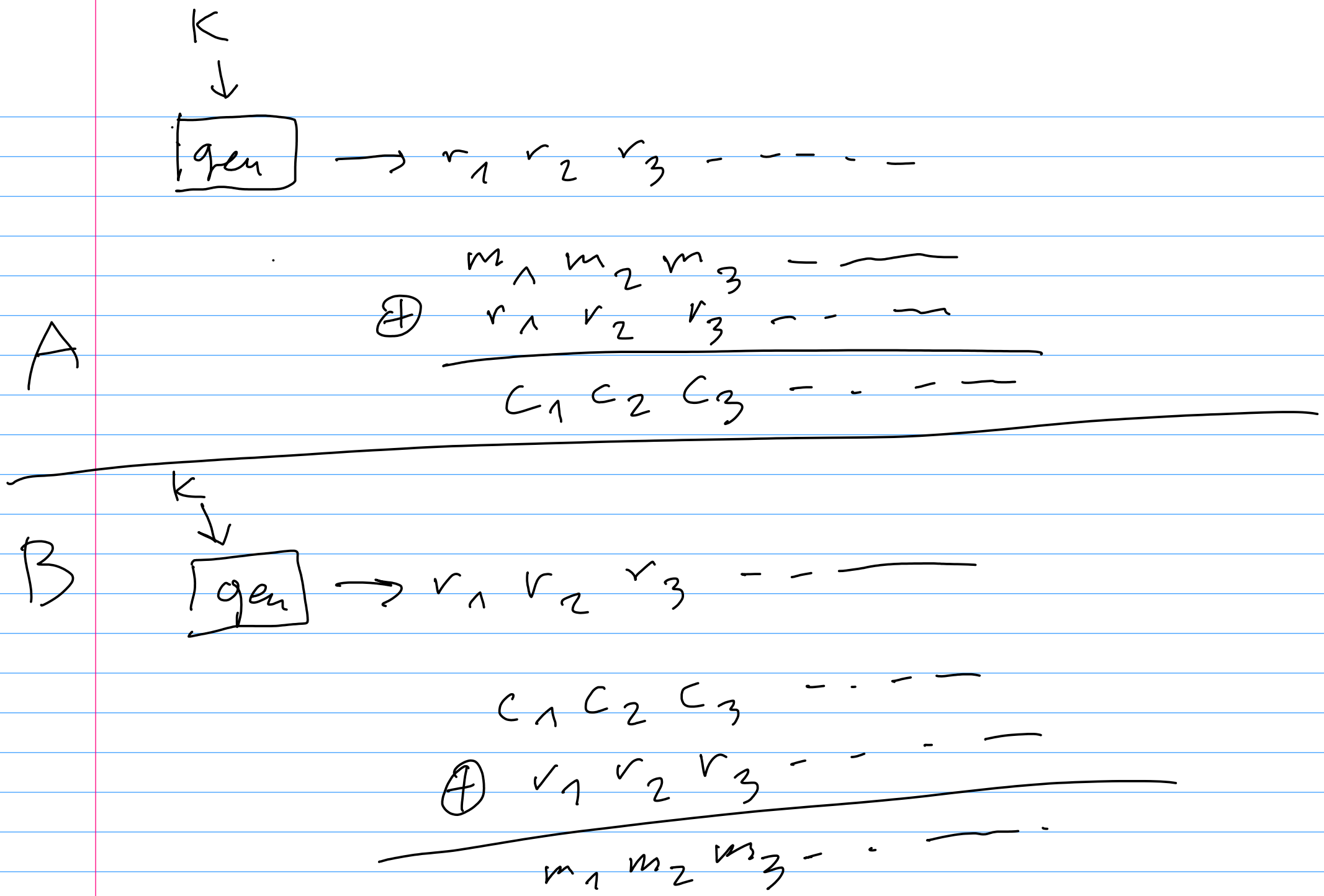
Szyfrowanie strumieniowe

One time pad $\oplus$ $\underline{m_1\ m_2\ m_3\ \ldots}$

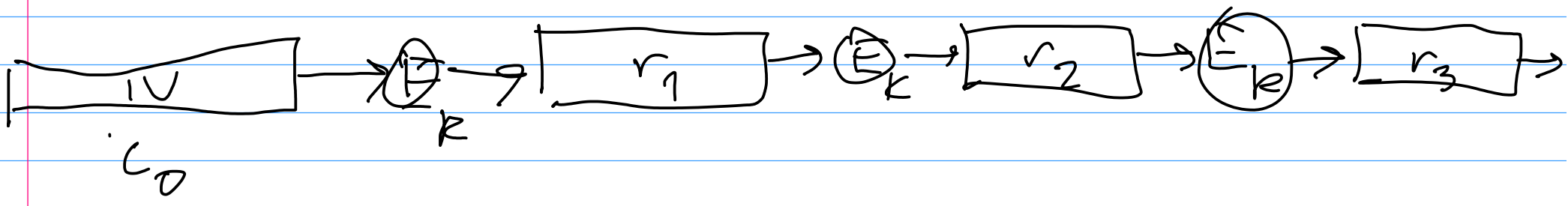
$r_1\ r_2\ r_3\ \ldots$

$c_1\ c_2\ c_3\ \ldots$

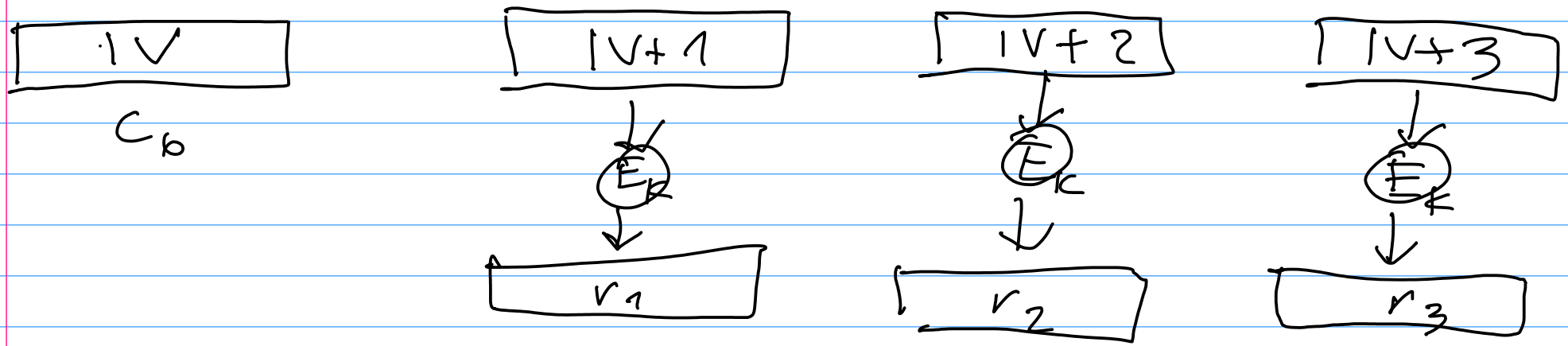
W szyfrowaniu strumieniowym $r_1\ r_2\ r_3\ \ldots$ nie jest kluczem tylko jest produkowane przez generator pseudolosowy przy użyciu tajnego klucza k


K
gen
$r_1\ r_2\ r_3$ - - - -
A
$m_1\ m_2\ m_3$ - - -
$r_1\ r_2\ r_3$ - - -
$c_1\ c_2\ c_3$ - - - -
K
B
gen
$r_1\ r_2\ r_3$ - - -
$c_1\ c_2\ c_3$ - - - -
$r_1\ r_2\ r_3$ - - - -
$m_1\ m_2\ m_3$ - - - -

OFB

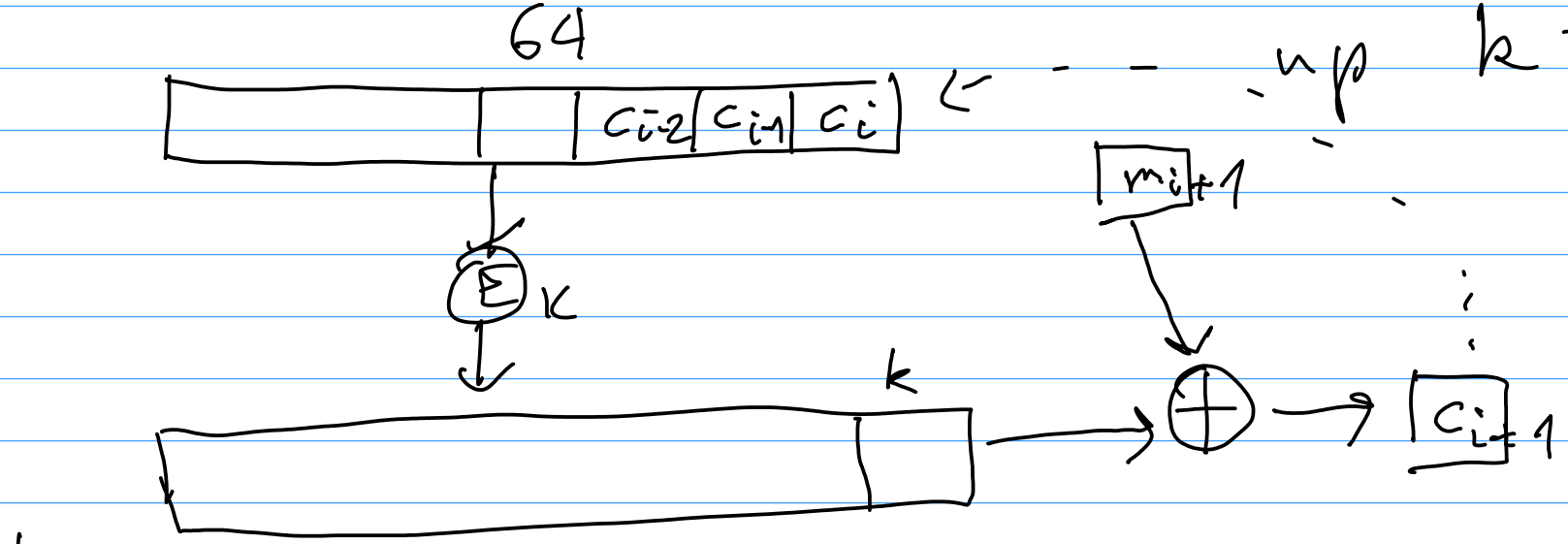


CTR (Counter mode)

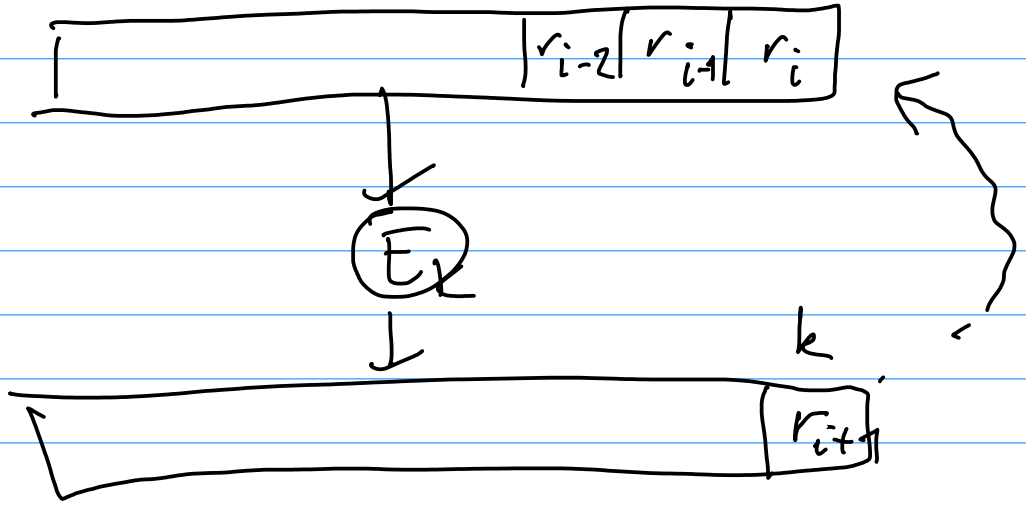


k - CFB

mesyfanie bloki, k-bitowe



k - OFB



Kryptoanaliza DES

Obecnie najprostszym sposobem jest przeszukanie przestrzeni wszystkich kluczy.

Kryptoanaliza różnicowa

Kryptoanaliza liniowa

3-DES

$C = E_{K_1} D_{K_2} E_{K_3}(m)$

Dla czego nie 2-DES ? $c = E_{K_1} E_{K_2}(m)$

Attak Meet in the Middle

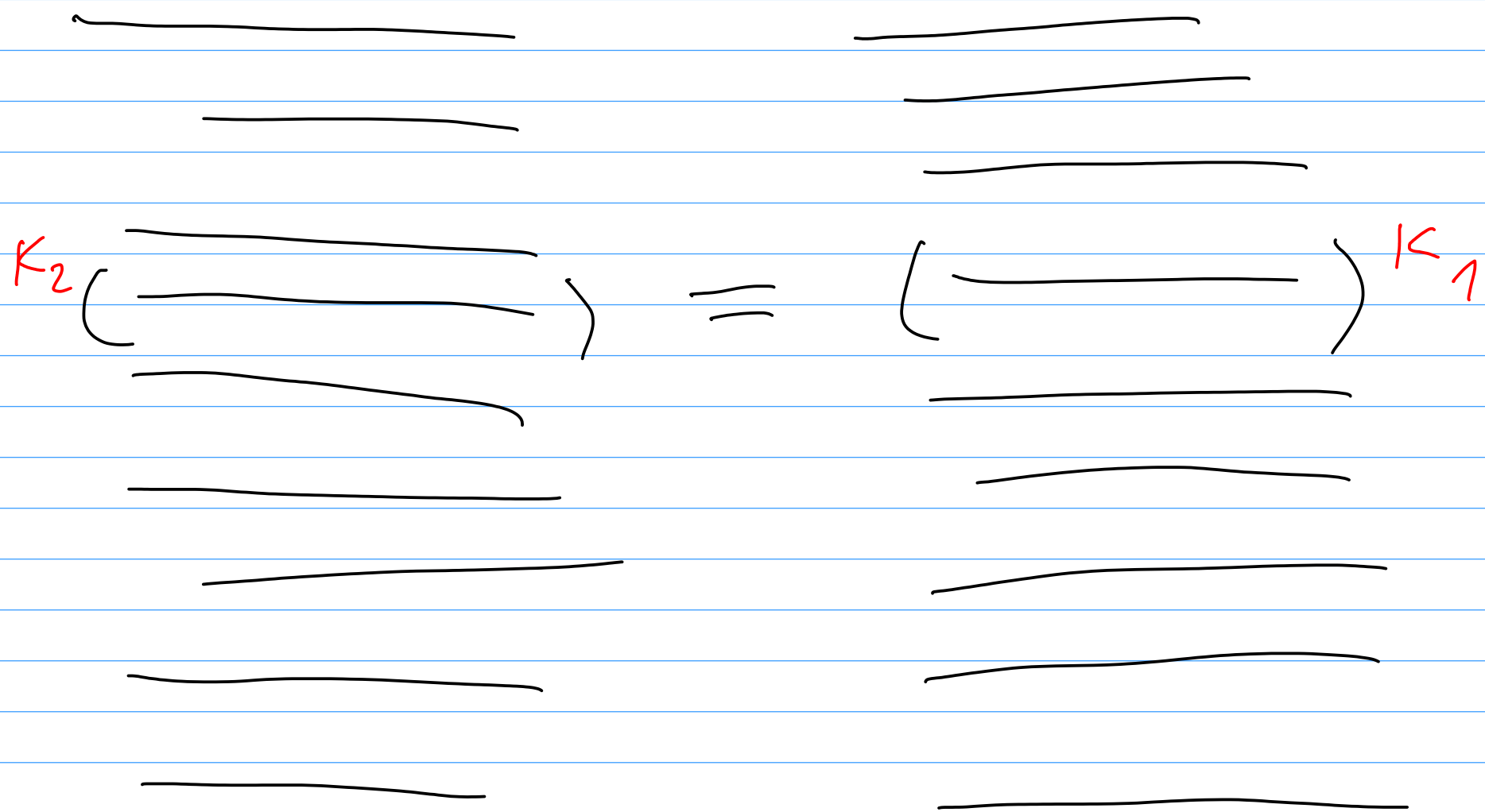
$m_1 m_2 m_3$ $\quad$ $c_1 c_2 c_3$

$E_{K_2}(m_1)\ E_{K_2}(m_2) E_{K_2}(m_3)$ $\quad$ $D_{K_1}(c_1)\ D_{K_1}(c_2) D_{K_1}(c_3)$

$\forall K_2$ obl porzesq trojke $\quad$ $\forall K_1$ obl tabe trojke

wyniki sortujemy $\quad$ wyniki sortujemy

odp (k_1, k_2)



Bezpieczeństwo alg szyfrującego w sensie LOR (Left or right)

Dany jest pewien alg szyfrujący $E_k(\cdot)$

Algorytm ten jest bezpieczny w sensie LOR gdy nie umiemy odróżnić 2 maszyn szyfrujących:

$$O_0(M_0, M_1) = E_k(M_0)$$

$$O_1(M_0, M_1) = E_k(M_1)$$

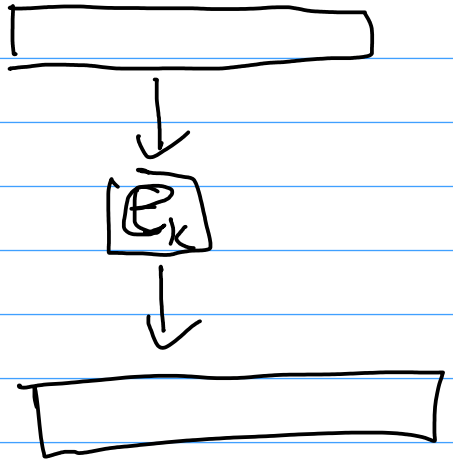
Przykład: ECB nie jest bezpieczny w sensie LOR

$$O_i(M_0, M_0) = E_K(M_0)$$
$$O_i(M_1, M_1) = E_K(M_1)$$
$$O_i(M_0, M_1) = E_K(M_i)$$

Tu tryb CBC jest bezpieczny w sensie LOR (jeśli szyfr blokowy na którym bazuje jest bezpieczny)

Szyfr blokowy

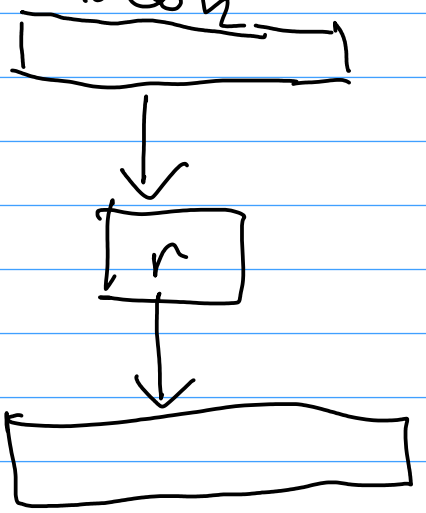
blok



szyfrogram bloku

maszyna losująca

blok



losowy blok wyjściowy

Szyfr blokowy jest bezpieczny gdy nie można go odróżnić od maszyny losującej

Def

e_k jest bezpieczny jeśli dla dowolnego algorytmu A który działa w max czasie t i odwołuje się do e_k (lub v) max l razy (A zwraca 0 v 1)

$$\left| Pr\left(A(e_k) = 1\right) - Pr\left(A(v) = 1\right) \right| < \varepsilon$$

E_k jest bezpieczny w sensie ROR gdy nie można odróżnić 2 wyroczni

$$E_k(M) \quad i \quad E_k(R(M))$$

gdzie $R(M)$ to losowy ciąg o tej samej dł co M

ROR to skrót od Random or Real

Tu E_k jest bezpieczny w sensie LOR $\Leftrightarrow$ E_k jest bezpieczny w sensie ROR

LOR

$\forall$ A działający w czasie T i wykonujący szyfrowanie (blokowe)

(A zwraca 0 v 1)

$$\left| Pr\left(A(O_0) = 1\right) - Pr\left(A(O_1) = 1\right) \right| < \varepsilon$$

ROR

$$\left| Pr\left(A(E_k) = 1\right) - Pr\left(A(E_k(R)) = 1\right) \right| < \varepsilon$$

$$O_0(M_0, M_1) = E_{1c}(M_0)$$

$$O_1(M_0, M_1) = E_k(M_1)$$

$$O_2(M_0, M_1) = E_k(R(M_0))$$

Zał. że szyfr jest bezpieczny w sensie ROR i pokazmy że jest bezpieczny w sensie LOR

$$|Pr(A(O_1)=1) - Pr(A(O_0)=1)| =$$
$$|Pr(A(O_1)=1) - Pr(A(O_2)=1) +$$
$$Pr(A(O_2)=1) - Pr(A(O_0)=1)| \leq$$
$$|Pr(A(O_1)=1) - Pr(A(O_2)=1)| +$$
$$|Pr(A(O_0)=1) - Pr(A(O_2)=1)| \leq$$
$$2\,|Pr(A(E_k)=1) - Pr(A(E_k(R)=1)| < 2\varepsilon$$

Załóżmy, że szyfr jest bezpieczny w sensie LOR i pokażmy że jest bezpieczny w sensie ROR

Gdyby szyfr nie był bezpieczny w sensie ROR, to można byłoby by odróżniać $E_k(M)$ i $E_k(R(M))$

$\exists A$

$$\left| Pr(A(E_k) = 1) - Pr(A(E_k(R) = 1) \right| \geq \varepsilon$$

$$O_0(M, R(M)) = E_k(M)$$

$$O_1(M, R(M)) = E_k(R(M))$$