# Operační režimy (módy) blokových šifer

Andrew Kozlík

KA MFF UK

# Operační režimy (módy) blokových šifer

- Říkáme, že šifra $(\mathcal{P}, \mathcal{C}, \mathcal{K}, E, D)$ je *bloková*, jestliže $\mathcal{P} = \mathcal{C} = \{0,1\}^b$ pro nějaké $b$.
  - Například DES ($b = 64$ bitů) nebo AES ($b = 128$ bitů).

- **Problém:** Jak šifrovat zprávu, která je delší než blok šifry?

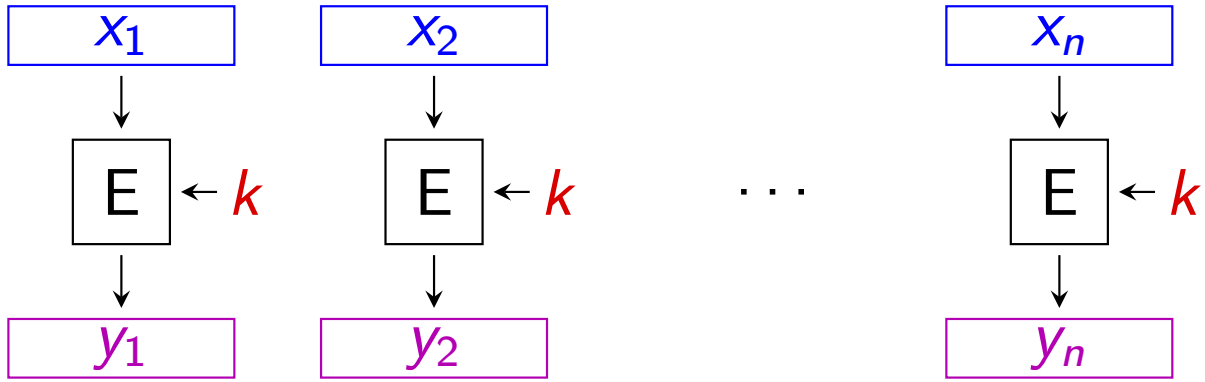
- Otevřený text rozdělíme na bloky délky $b$ a ty zpracujeme v některém operačním režimu.

- $x_i$ značíme $i$-tý blok otevřeného textu (OT).

- $y_i$ značíme $i$-tý blok šifrového textu (ŠT).

# Režim elektronické kódové knihy (ECB)

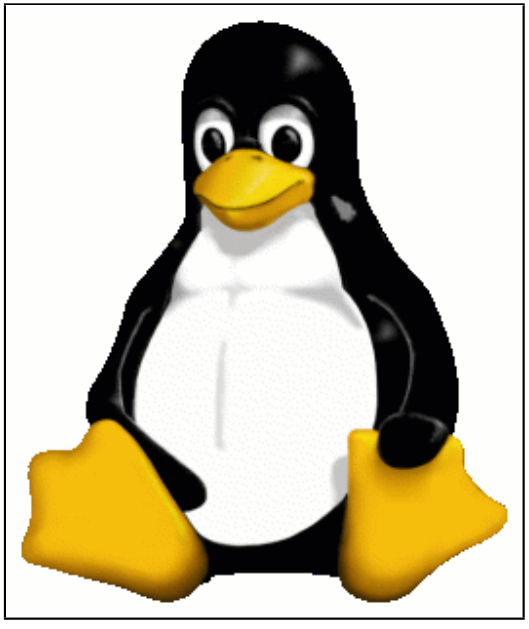
- **Anglicky:** electronic codebook mode (ECB)
- $y_i = E(k, x_i)$



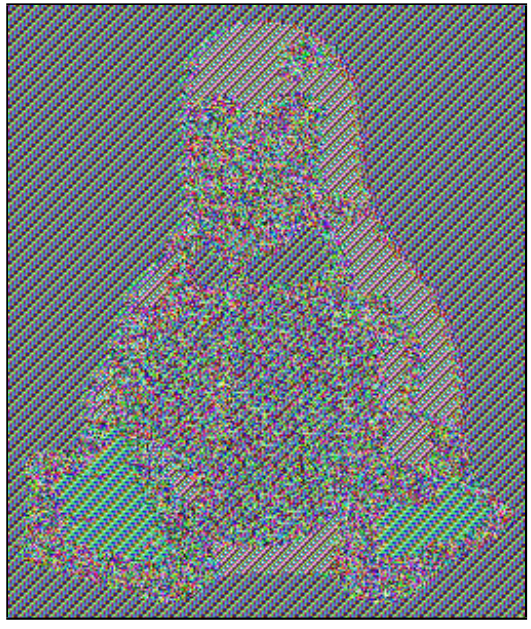

- **Problém:** Jsou-li dva bloky OT totožné, pak i příslušné bloky ŠT jsou totožné.
  - Uniká informace o shodách v OT.
    $\Rightarrow$ Útočník může získat informace o struktuře OT.
  - Snadná tvárnost: Útočník může bloky ŠT permutovat, duplikovat nebo mazat a docílí tím téže změny v OT.

# Únik informace o struktuře OT v ECB režimu

- **Příklad:**



Originál



ECB

CBC

# Tvárnost ECB režimu

- **Anglicky:** *malleability*
- **Příklad:**
  - Alice posílá platební příkaz „. . . převeďte 1 000 Kč. . . “:

    OT: | . . . př | eveď | te␣1 | ␣000 | ␣Kč. . . |

    ŠT: | $y_{15}$ | $y_{16}$ | $y_{17}$ | $y_{18}$ | $y_{19}$ |
  - Útočník zadrží ŠT, duplikuje blok $y_{18}$ a pošle vše dál.
  - Banka přijme pozměněný ŠT a dešifruje ho:

    ŠT: | $y_{15}$ | $y_{16}$ | $y_{17}$ | $y_{18}$ | $y_{18}$ | $y_{19}$ |

    OT: | . . . př | eveď | te␣1 | ␣000 | ␣000 | ␣Kč. . . |

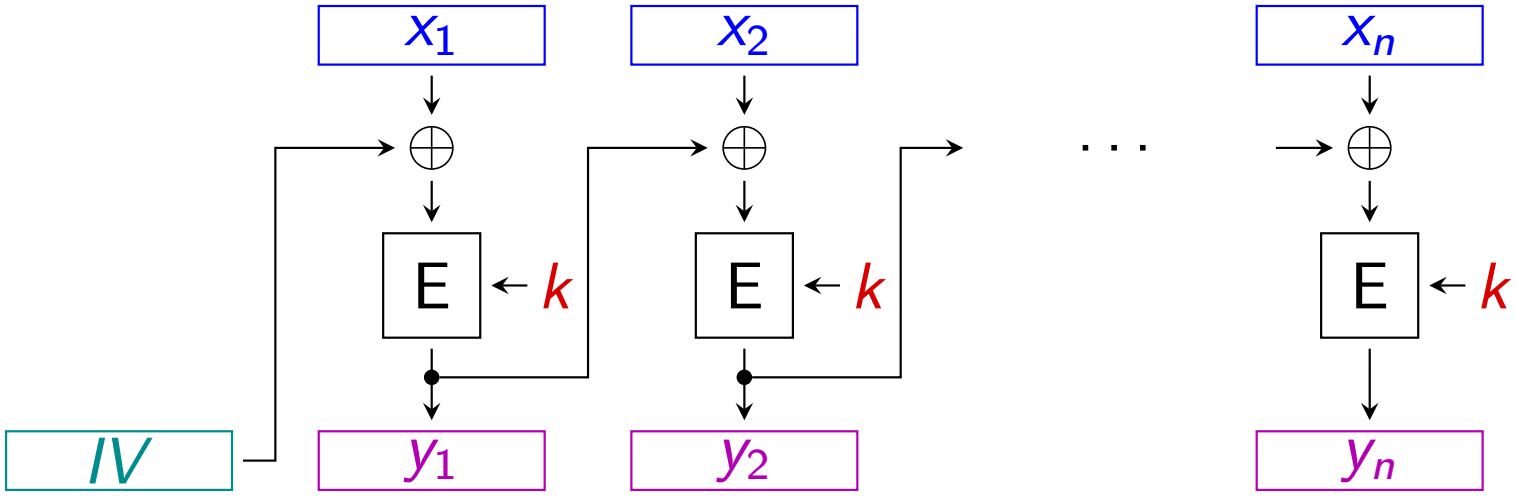
    „. . . převeďte 1 000 000 Kč. . . “
  - Banka nemůže poznat, že zpráva byla pozměněna.
- **Řešení:** Zpráva musí být doprovázena tzv. *autentizačním kódem zprávy*, který ochrání její integritu.

# Inicializační vektor ($IV$)

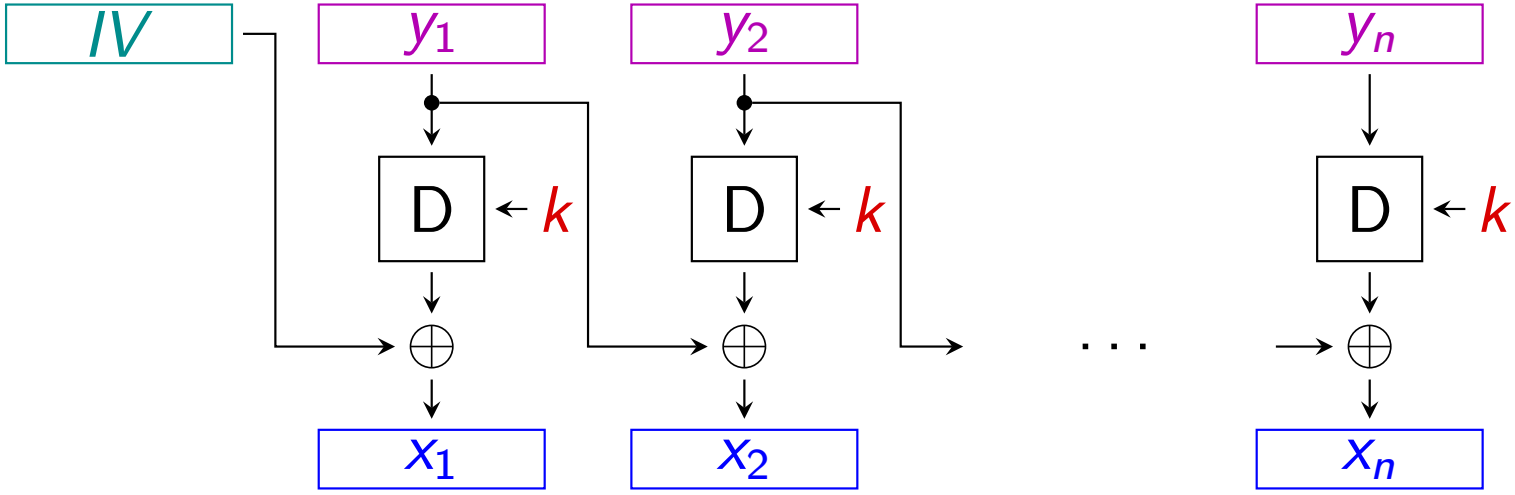
- Operační režimy zpravidla využívají tzv. *inicializační vektor*.
- $IV$ je blok délky $b$, který se generuje pro každý OT.
- $IV$ nemusí být tajný.
- Obvykle se generuje náhodně.
- Obvykle se posílá společně s ŠT, např. $(IV, y_1, y_2, \ldots, y_n)$.

# Režim cipher-block chaining (CBC)

- **Šifrování:** $y_i = E(k, x_i \oplus y_{i-1})$, kde $y_0 := IV$.




- **Dešifrování:** $x_i = D(k, y_i) \oplus y_{i-1}$, kde $y_0 := IV$.



# Režim cipher-block chaining (CBC)

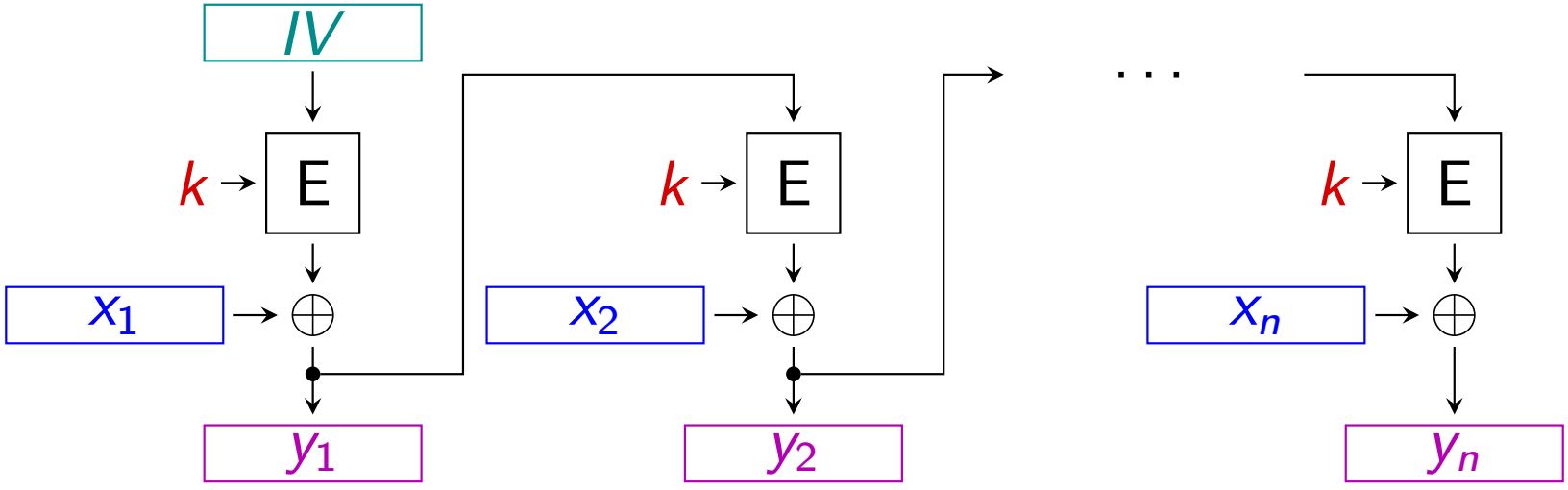
- CBC řeší problém ECB:
  Stejné bloky se v různých kontextech zašifrují jinak.
- $IV$ se generuje nově pro každý OT.
  - Kdyby se $IV$ neměnil, tak by útočník poznal, když dva OT začínají stejnými bloky.
- $IV$ musí být nepředvídatelný.
  - V opačném případě lze provést chosen-plaintext útok k ověření hypotézy o otevřeném textu.
  - Útočník zachytí ŠT $(IV, y_1, y_2, \ldots, y_n)$.
  - Útočník chce ověřit hypotézu, že pátý blok OT je $x'_5$.
  - Útočník předvídá, že další inicializační vektor bude $IV'$.
  - Útočník nechá zašifrovat zprávu $IV' \oplus x'_5 \oplus y_4$.
    Pokud je výsledkem $y_5$, pak hypotéza platí.
  - V praxi: BEAST attack na SSL/TLS.

# Vlastnosti CBC režimu

- Dojde-li při přenosu k **chybě** v jednom bitu bloku $y_i$, pak
  - blok $x_i$ se dešifruje chybně,
  - v bloku $x_{i+1}$ bude chyba v jednom bitu a
  - ostatní bloky se dešifrují správně.
- Dojde-li při přenosu k **výpadku** bloku $y_i$, pak
  - ztratíme $x_i$,
  - $x_{i+1}$ se dešifruje chybně a
  - ostatní bloky se dešifrují správně.
- Dojde-li při přenosu k **výpadku** neceločíselného násobku bloku, pak všechny další bloky se dešifrují chybně.
- Šifrování nelze paralelizovat.
- Dešifrování lze paralelizovat.

# Režim cipher feedback (CFB)

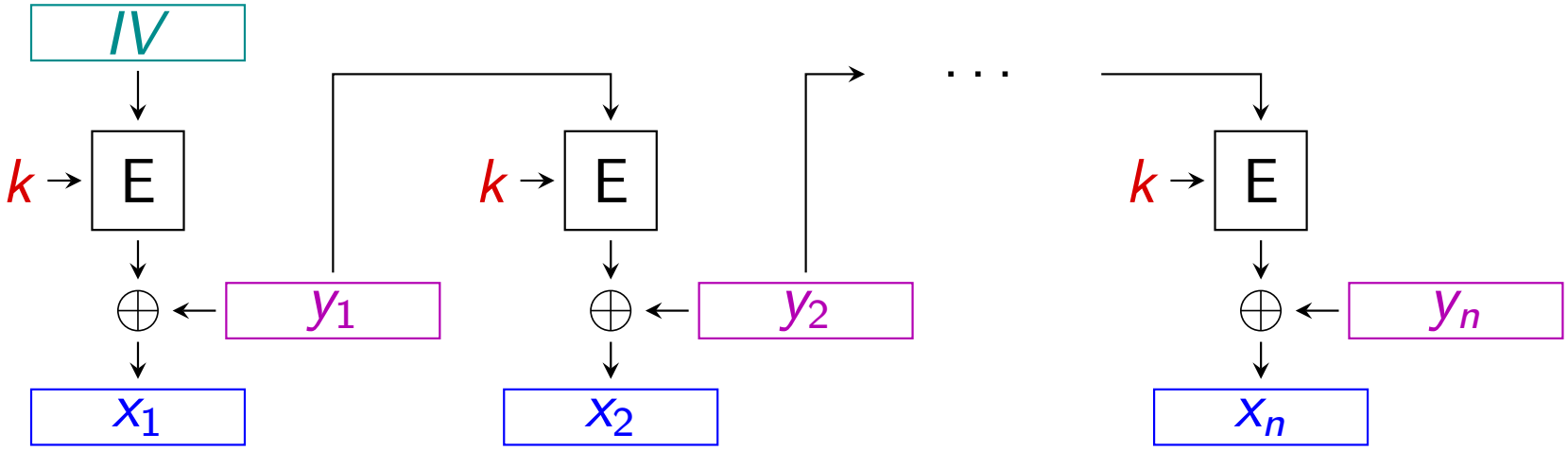
- **Šifrování:** $y_i = E(k, y_{i-1}) \oplus x_i$, kde $y_0 = IV$.




- $IV$ musí být jedinečný pro každý OT zašifrovaný stejným $k$.
  - Takový $IV$ se nazývá *nonce* (number used only once).
  - Kdyby se dvakrát šifrovalo se stejným $IV$ a se stejným $k$, pak by se k prvnímu bloku OT přičítal stejný výstup
- Šifrování nelze paralelizovat.

# Režim cipher feedback (CFB)

- **Dešifrování:** $x_i = E(k, y_{i-1}) \oplus y_i$, kde $y_0 = IV$.




- Dojde-li při přenosu k **chybě** v jednom bitu bloku $y_i$, pak
  - v bloku $x_i$ bude chyba v jednom bitu a
  - blok $x_{i+1}$ se dešifruje chybně.
- Dojde-li při přenosu k **výpadku** bloku $y_i$, pak
  - ztratíme $x_i$ a
  - $x_{i+1}$ se dešifruje chybně.
- Dešifrování lze paralelizovat.

# Modifikace CFB režimu

- Dojde-li při přenosu k výpadku neceločíselného násobku bloku, pak všechny další bloky se dešifrují chybně.
- CFB režim lze upravit tak, aby k samosynchronizaci došlo po výpadku $m$-násobku bitů, kde $1 \leq m \leq b$.

1. OT rozdělíme na bloky velikosti $m$ bitů.
2. *IV* zapíšeme do posuvného registru.
3. Obsah registru použijeme jako vstup do šifry.
4. Prvních $m$ bitů výstupu šifry přičteme k bloku OT a zbylé bity výstupu zahodíme.
5. Blok ŠT vsuneme na konec posuvného registru.
6. Pokračujeme krokem 3.

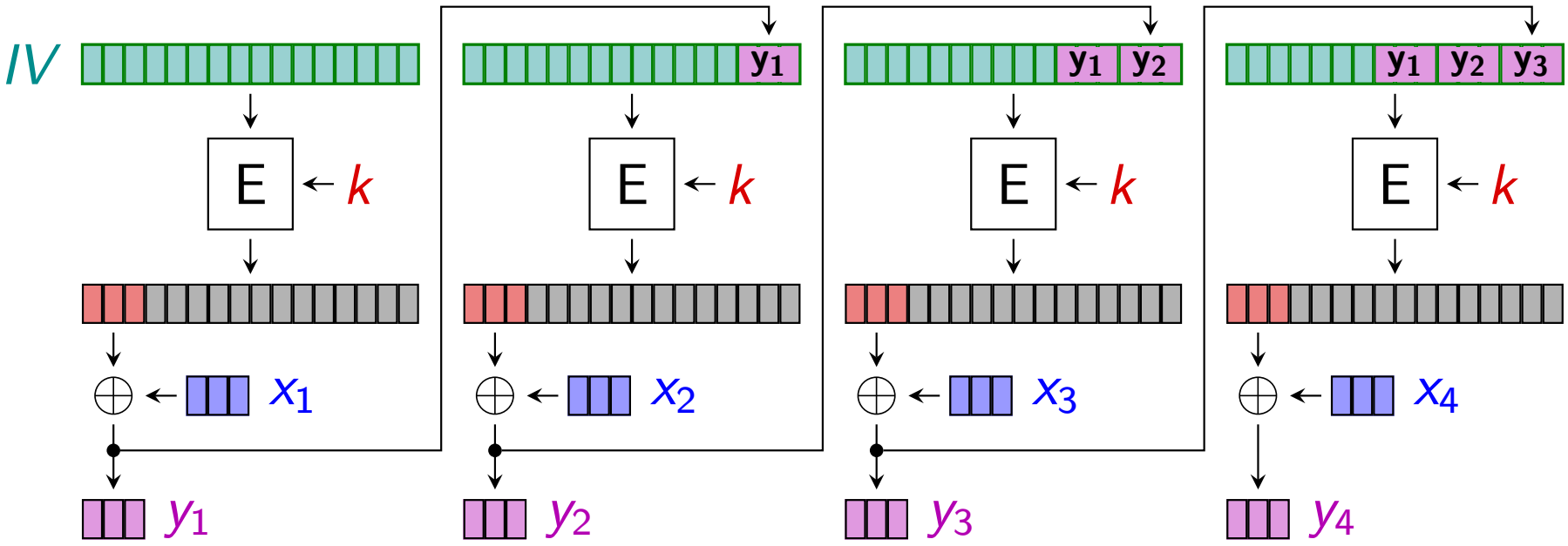
# Režim CFB-$m$

**Šifrování:**

- $s_1 = IV$
- $s_i = \text{tail}_{b-m}(s_{i-1}) \,\|\, y_{i-1}$
- $y_i = \text{head}_m(E(k, s_i)) \oplus x_i$

Posledních $b - m$ bitů $s_{i-1}$.

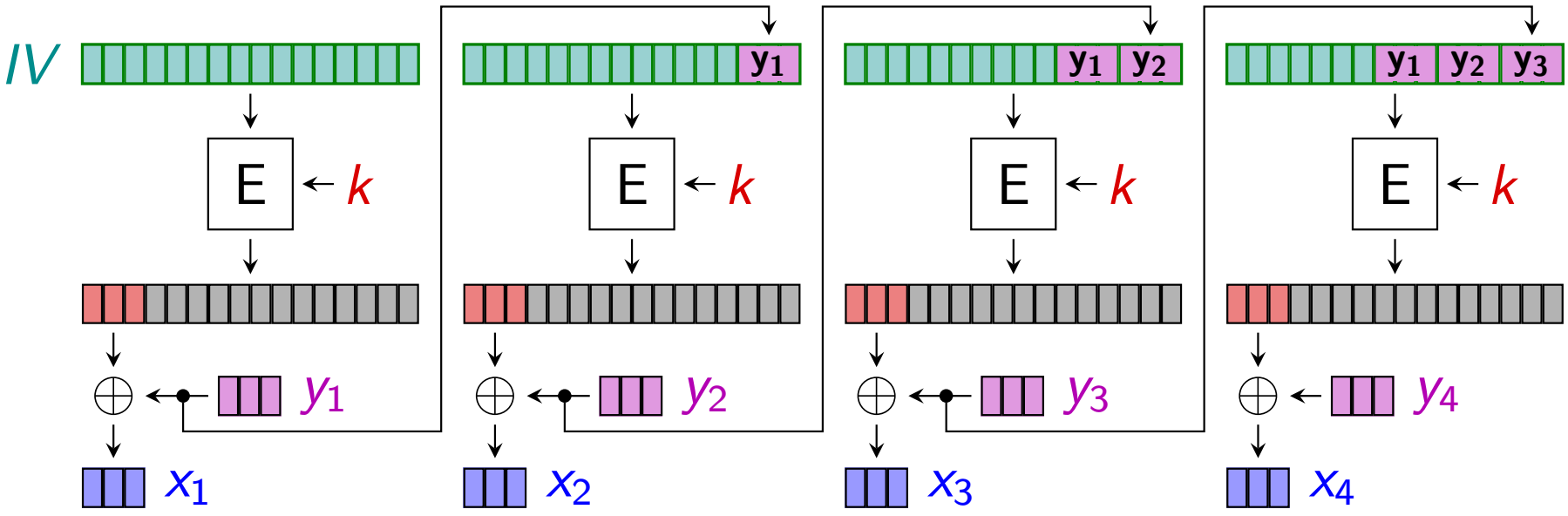
Prvních $m$ bitů $E(k, s_i)$.



# Režim CFB-$m$

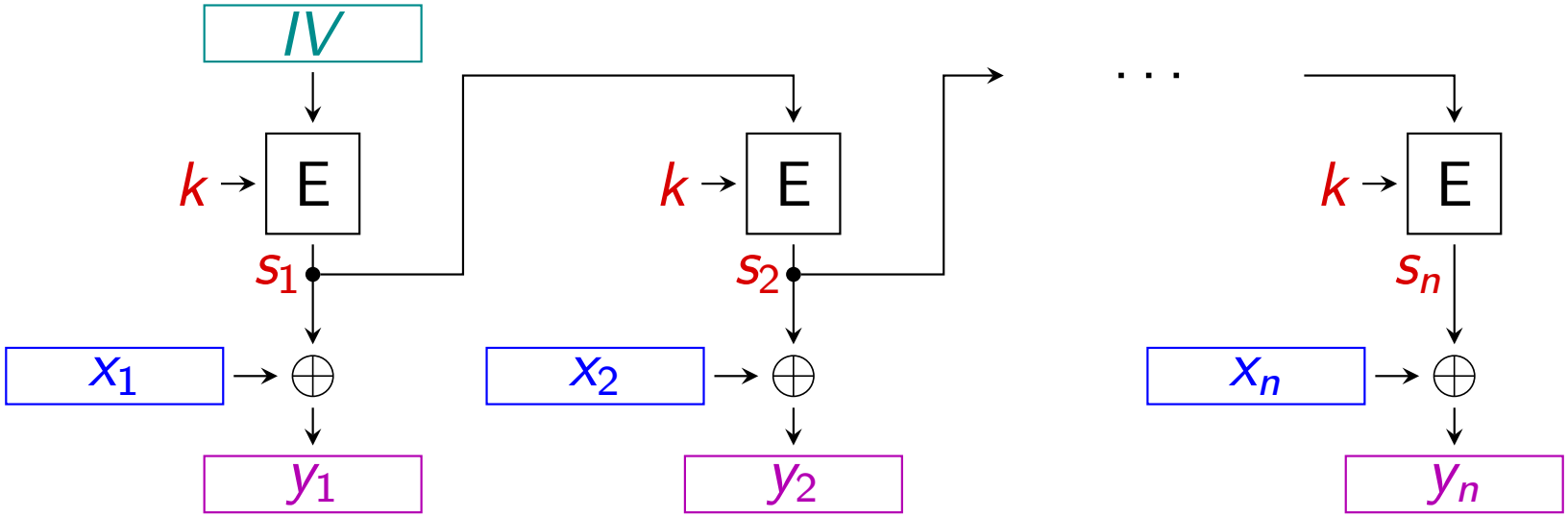
**Dešifrování:**

- $s_1 = IV$
- $s_i = \mathrm{tail}_{b-m}(s_{i-1}) \parallel y_{i-1}$
- $x_i = \mathrm{head}_m(E(k, s_i)) \oplus y_i$



# Režim output feedback (OFB)

- **Šifrování:** $y_i = s_i \oplus x_i$, kde $s_i = E(k, s_{i-1})$ a $s_0 = IV$.




- $IV$ musí být nonce.
- $IV$ může být předvídatelný.
- Dojde-li při přenosu k **chybě** v jednom bitu ŠT, pak bude chyba v jednom bitu OT.
- Dojde-li při přenosu k **výpadku** jakékoliv části ŠT, pak všechny další bloky se dešifrují chybně.

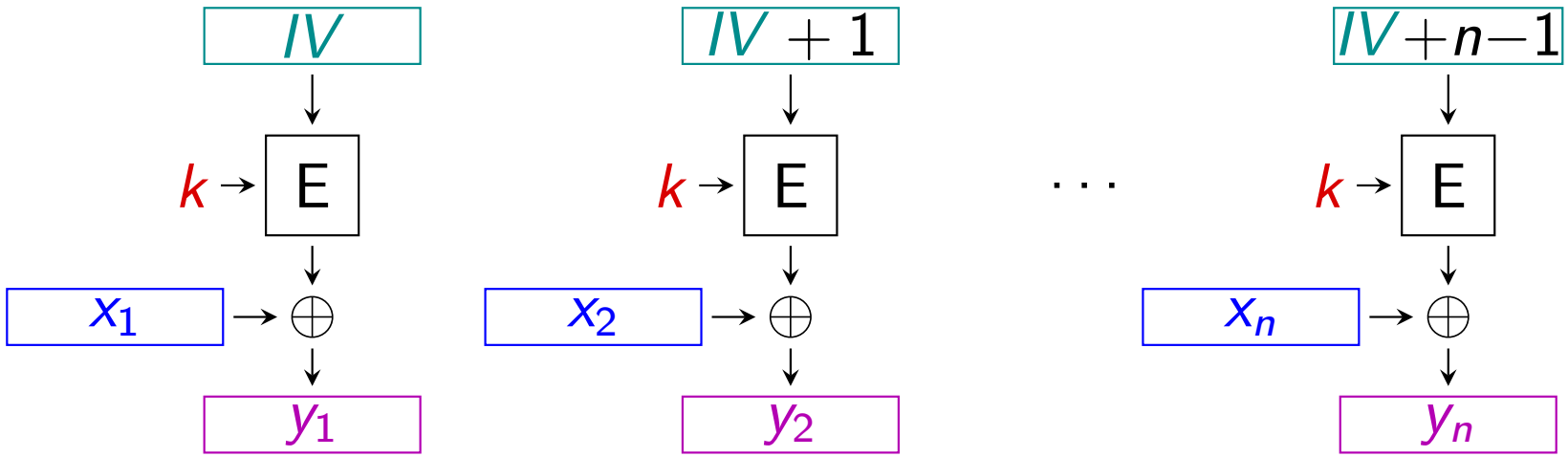
# Vlastnosti OFB režimu

- Známe-li *IV* předem, pak můžeme provést předvýpočet všech operací E.
- Předvýpočet nelze paralelizovat.
- OFB převádí blokovou šifru na tzv. *synchronní proudovou šifru*. Posloupnost $\{s_i\}_{i=1}^{n}$ nazýváme proud hesla.
- Proud hesla je periodický s periodou nejvýše $2^b$ bloků, ale perioda může být kratší.
- Střední hodnota periody je přibližně $2^{b-1}$ bloků.
  - Pravděpodobnost, že $i$-tý blok proudu hesla se shodne s *IV* zatímco předchozích $i-1$ nikoliv, je $2^{-b}$.
  - Střední hodnota periody je tedy

$$\sum_{i=1}^{2^b} \frac{i}{2^b} = \frac{2^b+1}{2} \text{ bloků.}$$

# Čítačový režim (CTR)

- **Anglicky:** counter mode (CTR)
- **Šifrování:** $y_i = x_i \oplus E(k, IV + i - 1)$.




- $IV$ musí být volen tak, aby se vstup do E nikdy neopakoval pro dané $k$. Dvě obvyklá řešení:

1. Zvolíme $IV$ jako poslední hodnotu čítače z minulého šifrování zvýšenou o 1.
2. Zvolíme $IV = nonce \,\|\, 0^m$, kde $nonce \in \{0,1\}^{b-m}$.
   - Délku zprávy omezíme na nejvýše $2^m$ bloků.
   - Typicky $m = 32$, což pro AES omezuje zprávy na 64 GB.

# Vlastnosti CTR režimu

- Šíření chyb a výpadků je v CTR shodné jako v OFB.
- Oproti OFB má CTR maximální periodu hesla.
- Oproti OFB lze v CTR rychle spočítat libovolný blok proudu hesla.
  - V CTR spočteme $i$-tý blok hesla jedním voláním funkce E.
  - V OFB spočteme $i$-tý blok hesla $i$ voláními funkce E.
- Známe-li $IV$ předem, pak můžeme provést předvýpočet všech operací E.
- Předvýpočet lze paralelizovat.

# Porovnání operačních režimů

| | ECB | CBC | CFB | OFB | CTR |
|---|---|---|---|---|---|
| Paralelizovatelné šifrování | ANO | NE | NE | NE | ANO |
| Paralelizovatelné dešifrování | ANO | ANO | ANO | NE | ANO |
| Lze předpočítat | NE | NE | NE | ANO | ANO |
| Šíření chyby v ŠT | 1 blok | 1 blok | 1 blok | NE | NE |
| Samosynchronizující | ANO | ANO | ANO | NE | NE |
| Obejde se bez implementace D | NE | NE | ANO | ANO | ANO |
| Znovupoužití IV | — |  |  |  |  |

# Padding

- **Problém:** Jak postupovat v ECB a CBC, když OT má délku, která není dělitelná velikostí bloku?
- **Řešení** (*zero padding*): Vyplníme nulovými bity na nejbližší násobek $b$.
- **Problém:** Jak pozná příjemce po dešifrování, které nuly na konci jsou součástí zprávy a které jsou výplň?
- **Řešení** (*bit padding*): Vyplníme posloupností bitů tvaru $100\ldots0$ na nejbližší násobek $b$.
- Příjemce ví, že poslední jednička v OT značí začátek výplně.
- **Nevýhoda:** Když je délka OT dělitelná $b$, pak musíme přidat na konec OT celý blok obsahující jen výplň. Minimální délka výplně je totiž 1 bit.

# PKCS #7 padding

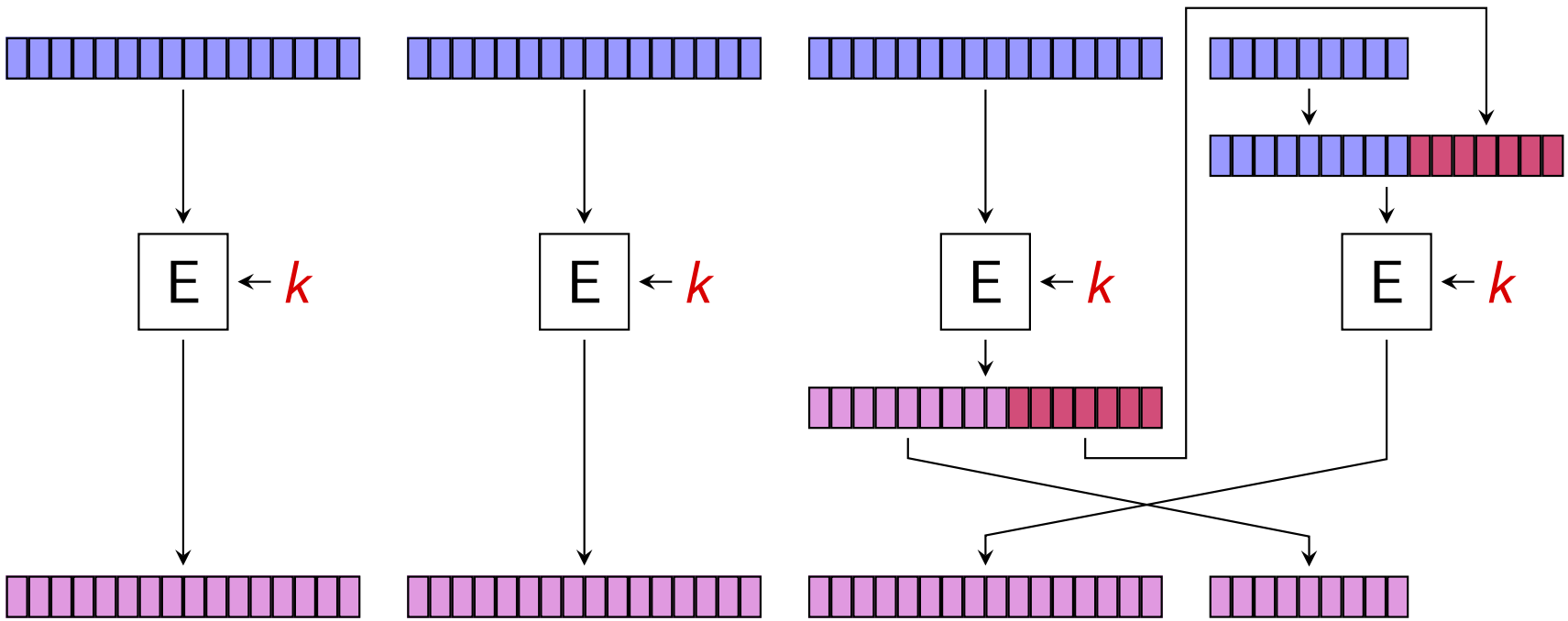
- Označme
  - $B$ velikost bloku v bajtech, $B = \frac{1}{8}b$, a
  - $\ell$ délku zprávy v bajtech.
- Zprávu na konci doplníme $B - (\ell \bmod B)$ bajty s hodnotou $B - (\ell \bmod B)$. Jinými slovy:

| Výplň | Podmínka |
|---|---|
| 0x01 | $\ell \equiv B - 1 \pmod{B}$ |
| 0x02 0x02 | $\ell \equiv B - 2 \pmod{B}$ |
| 0x03 0x03 0x03 | $\ell \equiv B - 3 \pmod{B}$ |
| $\vdots$ | $\vdots$ |

- Tato metoda funguje jen pro šifry s délkou bloku $B < 256$.
- **Nevýhoda:** Když je délka OT dělitelná velikostí bloku, pak musíme přidat na konec OT celý blok obsahující jen výplň.

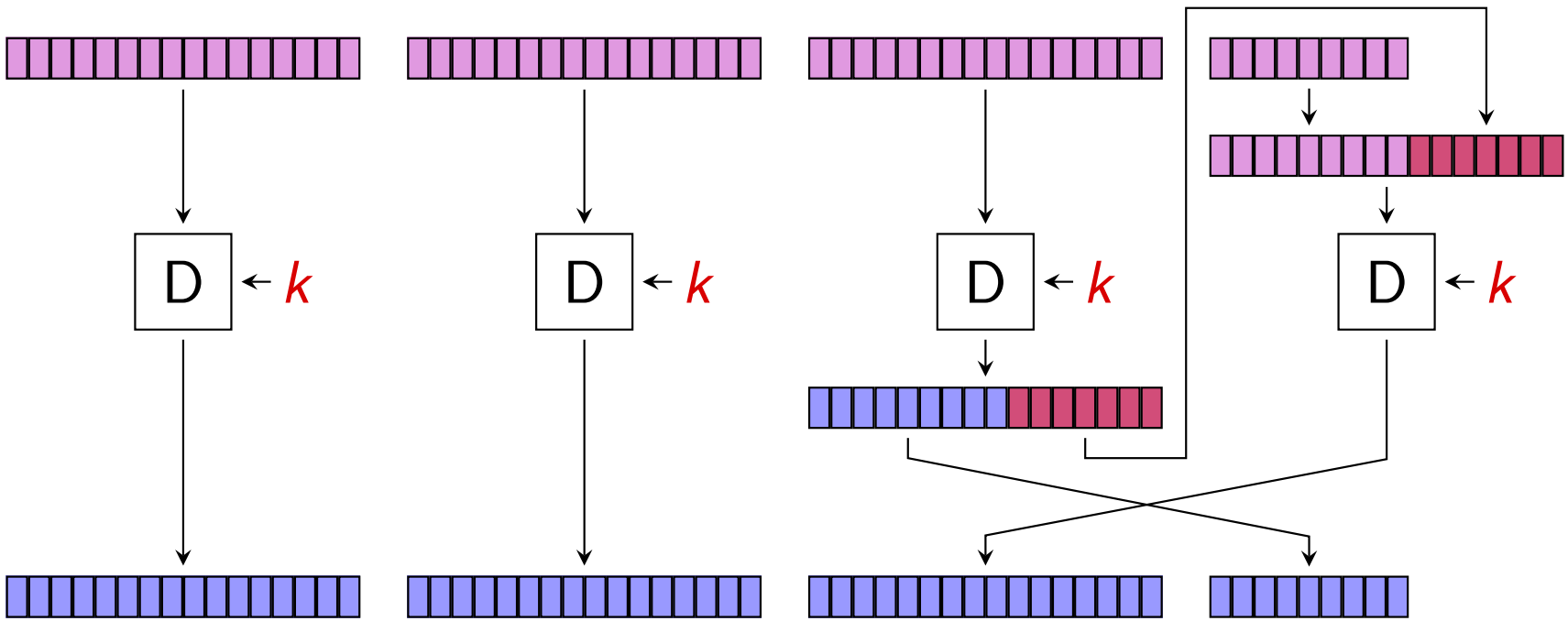
# Ciphertext stealing v ECB: Šifrování

- Bloky $x_1, \ldots, x_{n-1}$ zašifrujeme v ECB režimu.
- OT vyplníme posledními $b - \text{len}(x_{n-1})$ bity bloku $y_{n-1}$.
- Zašifrujeme poslední blok OT.
- Poslední dva bloky ŠT $y_{n-1}$ a $y_n$ vyměníme.
- ŠT na konci ořízneme na délku původního OT.



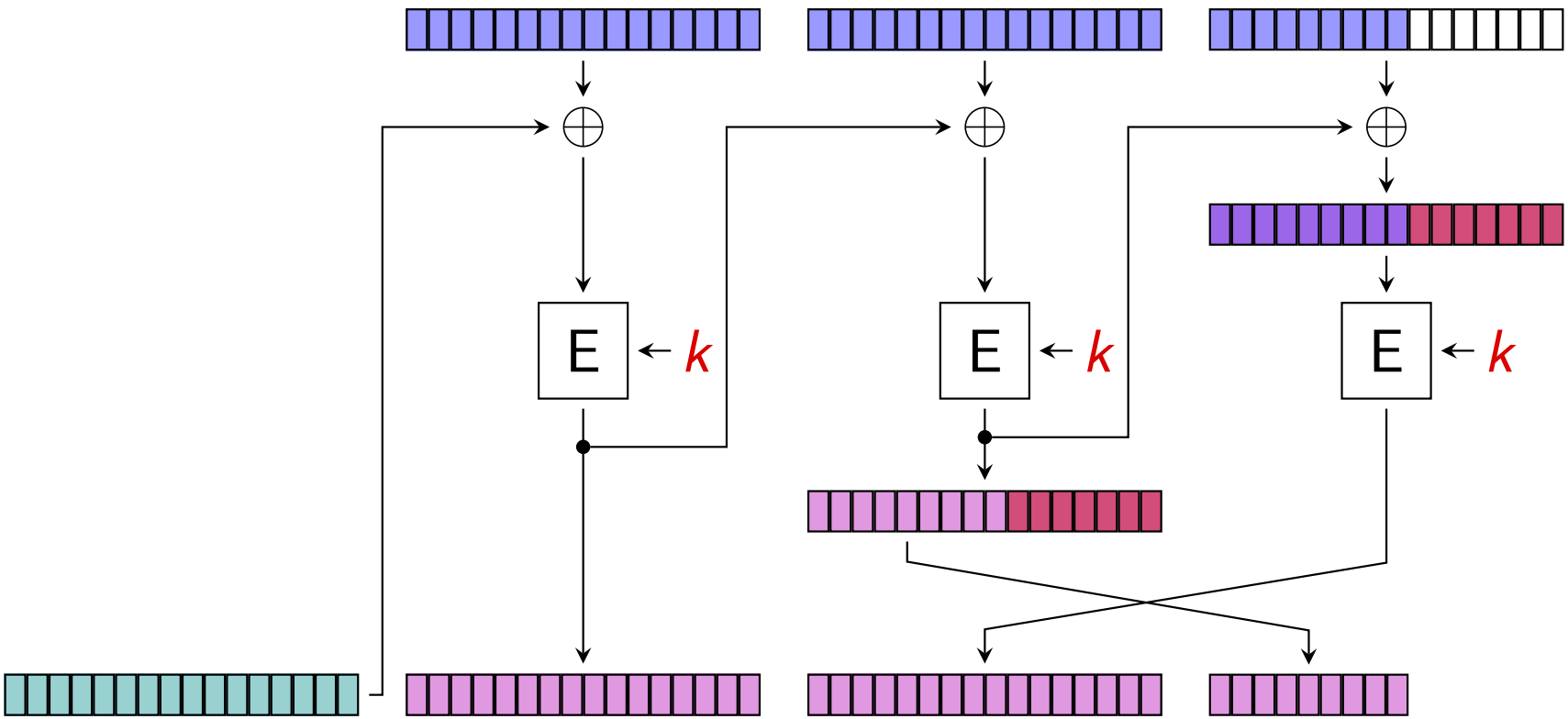
# Ciphertext stealing v ECB: Dešifrování

- Bloky $y_1, \ldots, y_{n-1}$ dešifrujeme v ECB režimu.
- ŠT vyplníme posledními $b - \text{len}(y_{n-1})$ bity bloku $x_{n-1}$.
- Dešifrujeme poslední blok ŠT.
- Poslední dva bloky OT $x_{n-1}$ a $x_n$ vyměníme.
- OT na konci ořízneme na délku původního ŠT.



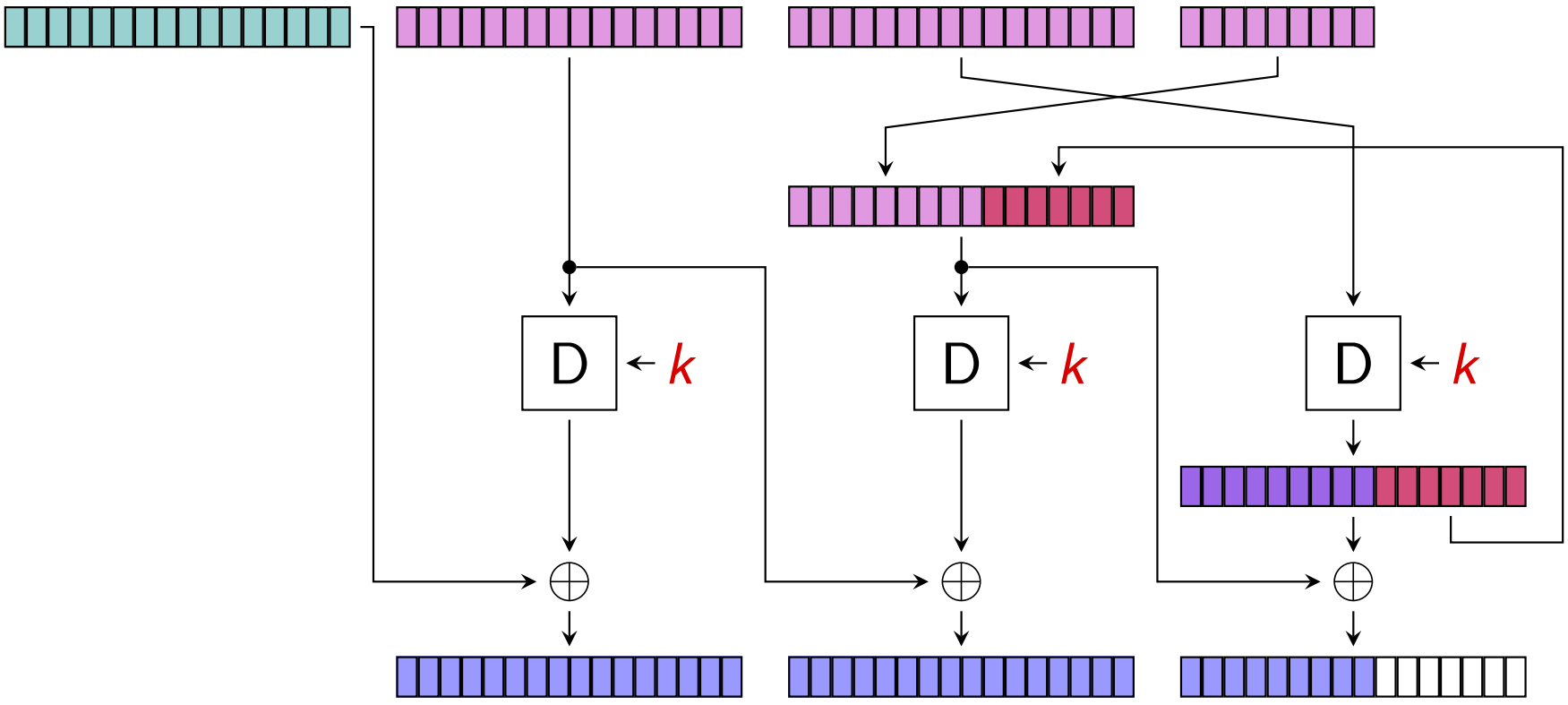
# Ciphertext stealing v CBC: Šifrování

- OT vyplníme nulovými bity na nejbližší násobek $b$.
- OT zašifrujeme v CBC režimu.
- Poslední dva bloky ŠT $y_{n-1}$ a $y_n$ vyměníme.
- ŠT na konci ořízneme na délku původního OT.



# Ciphertext stealing v CBC: Dešifrování

Proces dešifrování nejlépe ilustruje obrázek:



# Další operační režimy

- Všechny uvedené režimy trpí jistou mírou tvárnosti. Zajišťují pouze důvěrnost dat, ale nezajišťují integritu. Řešením je doprovodit zprávu autentizačním kódem.
- Existují režimy, které zajišťují důvěrnost, integritu a autenticitu dat:
  - GCM: Galois counter mode
  - CCM: Counter with CBC-MAC
  - EAX: Čítačový režim s OMAC
  - OCB: Offset codebook mode
- Existuje speciální režim pro šifrování pevných disků:
  - XEX-based tweaked-codebook mode with ciphertext stealing (XTS)
  - Úlohu IV nahrazuje číslo sektoru na disku.
  - Délka ŠT je shodná s délkou OT.