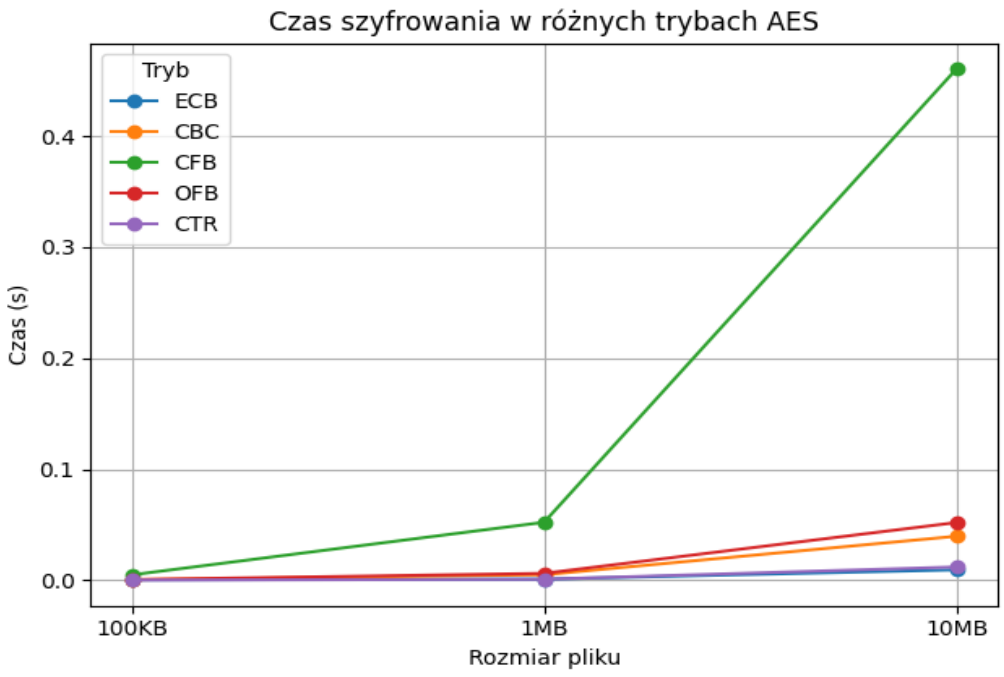
Podstawy kryptografii – szyfrowanie blokowe

Zadanie1

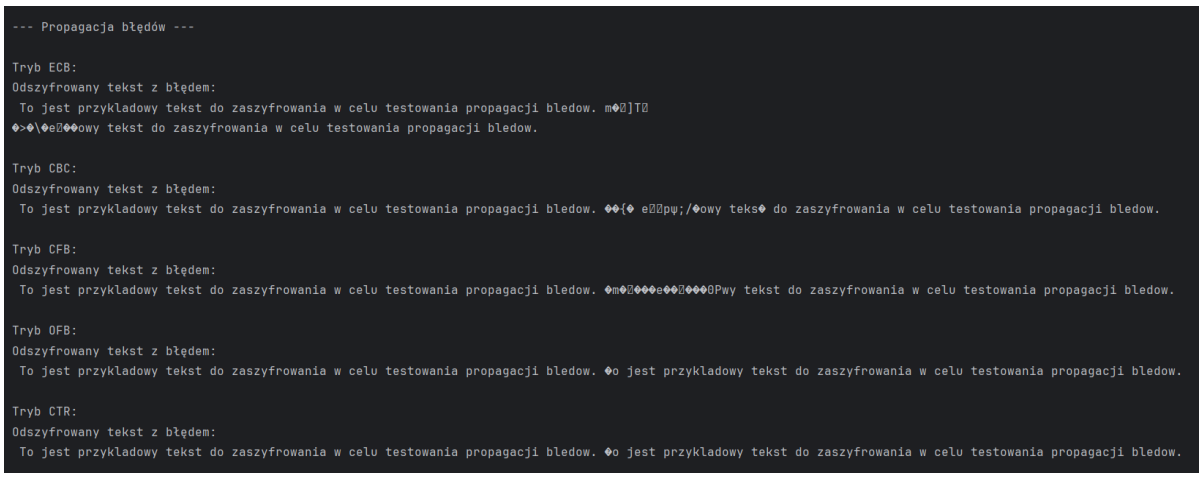
```
Plik: 100kb.txt
Tryb ECB | Szyfrowanie: 0.000337s | Deszyfrowanie: 0.000207s
Tryb CBC | Szyfrowanie: 0.000341s | Deszyfrowanie: 0.000313s
Tryb CFB | Szyfrowanie: 0.004852s | Deszyfrowanie: 0.005941s
Tryb OFB | Szyfrowanie: 0.000562s | Deszyfrowanie: 0.000887s
Tryb CTR | Szyfrowanie: 0.000140s | Deszyfrowanie: 0.000128s

Plik: 1mb.txt
Tryb ECB | Szyfrowanie: 0.000985s | Deszyfrowanie: 0.000546s
Tryb CBC | Szyfrowanie: 0.004938s | Deszyfrowanie: 0.003565s
Tryb CFB | Szyfrowanie: 0.052223s | Deszyfrowanie: 0.086085s
Tryb OFB | Szyfrowanie: 0.006197s | Deszyfrowanie: 0.003535s
Tryb CTR | Szyfrowanie: 0.000897s | Deszyfrowanie: 0.000954s

Plik: 10mb.txt
Tryb ECB | Szyfrowanie: 0.009384s | Deszyfrowanie: 0.009592s
Tryb CBC | Szyfrowanie: 0.039636s | Deszyfrowanie: 0.040145s
Tryb CFB | Szyfrowanie: 0.460581s | Deszyfrowanie: 0.484326s
Tryb OFB | Szyfrowanie: 0.051987s | Deszyfrowanie: 0.051033s
Tryb CTR | Szyfrowanie: 0.011949s | Deszyfrowanie: 0.012199s
```



Tryby CTR i ECB zapewniają najszybsze przetwarzanie danych, natomiast tryby takie jak CBC, OFB i CFB działają wolniej, szczególnie CFB, co wynika z zależności między kolejnymi blokami danych.

Podsumowując, im większy plik, tym większe znaczenie ma wybór trybu szyfrowania — należy znaleźć równowagę między prędkością a poziomem ochrony (np. ECB jest szybki, ale nie zapewnia odpowiedniego bezpieczeństwa).

Zadanie 2.



W zaszyfrowanym tekście wprowadzamy błąd – zmieniamy jeden bajt w jego środku. Następnie wykonywane jest deszyfrowanie, aby zaobserwować, jaki wpływ ma ten błąd na odszyfrowany tekst.

Wyniki obserwacji:
W trybach CTR i OFB błąd wpływa tylko na jeden bajt odszyfrowanego tekstu – reszta pozostaje niezmieniona i możliwa do odczytania.

W trybach ECB, CBC i CFB błąd wpływa na cały blok danych, w którym został wprowadzony. Reszta tekstu po deszyfrowaniu jest nadal możliwa do poprawnego odczytu.

Tryby CTR i OFB lepiej radzą sobie z pojedynczymi błędami w szyfrogramie.

Zadanie 3.

Tryb CBC został zaimplementowany przy użyciu trybu ECB w następujący sposób:

1. Generujemy losowy IV (wektor inicjalizacyjny) o długości 16 bajtów.
2. Dzielimy dane na bloki, które przed szyfrowaniem XORujemy z poprzednim blokiem (lub IV).
3. Używamy trybu ECB do szyfrowania tych bloków.
4. Podczas deszyfrowania bloków, wynik odszyfrowanego bloku XORujemy z poprzednim, aby odzyskać oryginalne dane.
5. Zastosowano dopełnianie danych do 16 bajtów przed szyfrowaniem i usuwanie go po deszyfrowaniu.