Implementacja wybranych trybów pracy szyfratorów blokowych oraz badanie jakości szyfrów blokowych

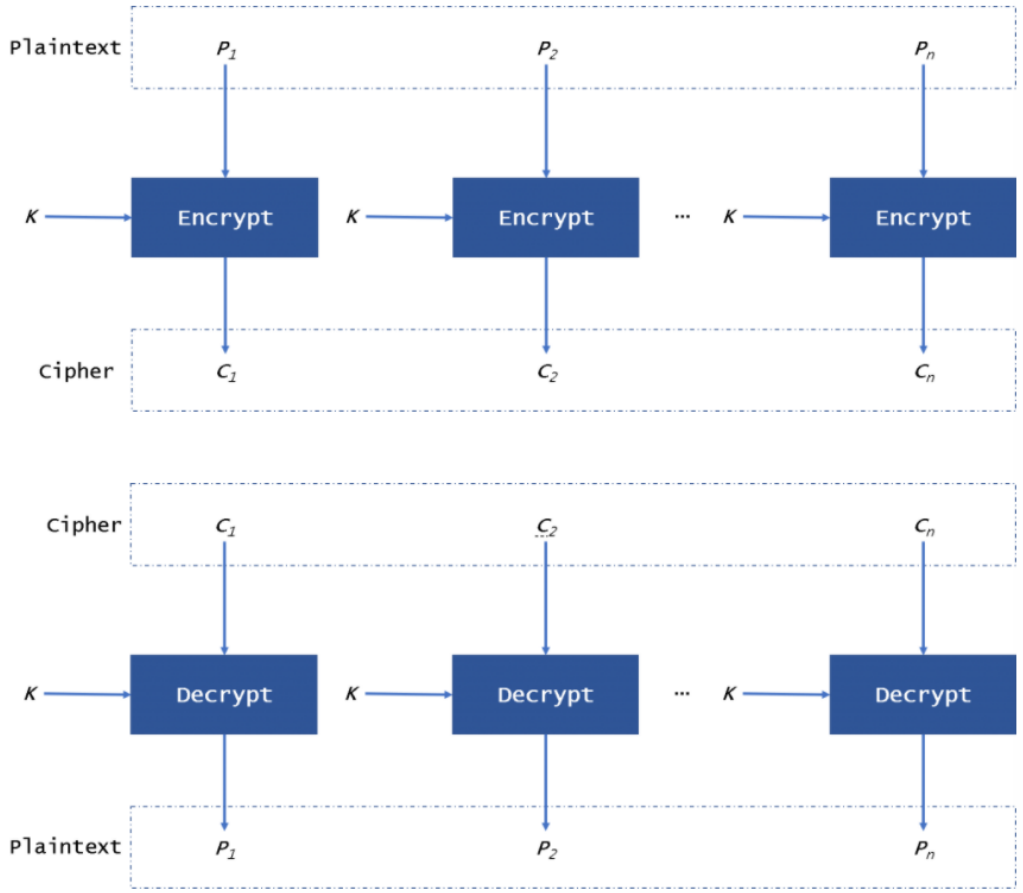
Paweł Koch 145330, 19.03.2022

1.Implementacja:

Do implementacji użyto języka Python w wersji 3.9.2 oraz 3.10 z biblotekami: pycryptodome (bibliotek kryptograficzna), timeit(biblioteka do pomiarów czasowych)

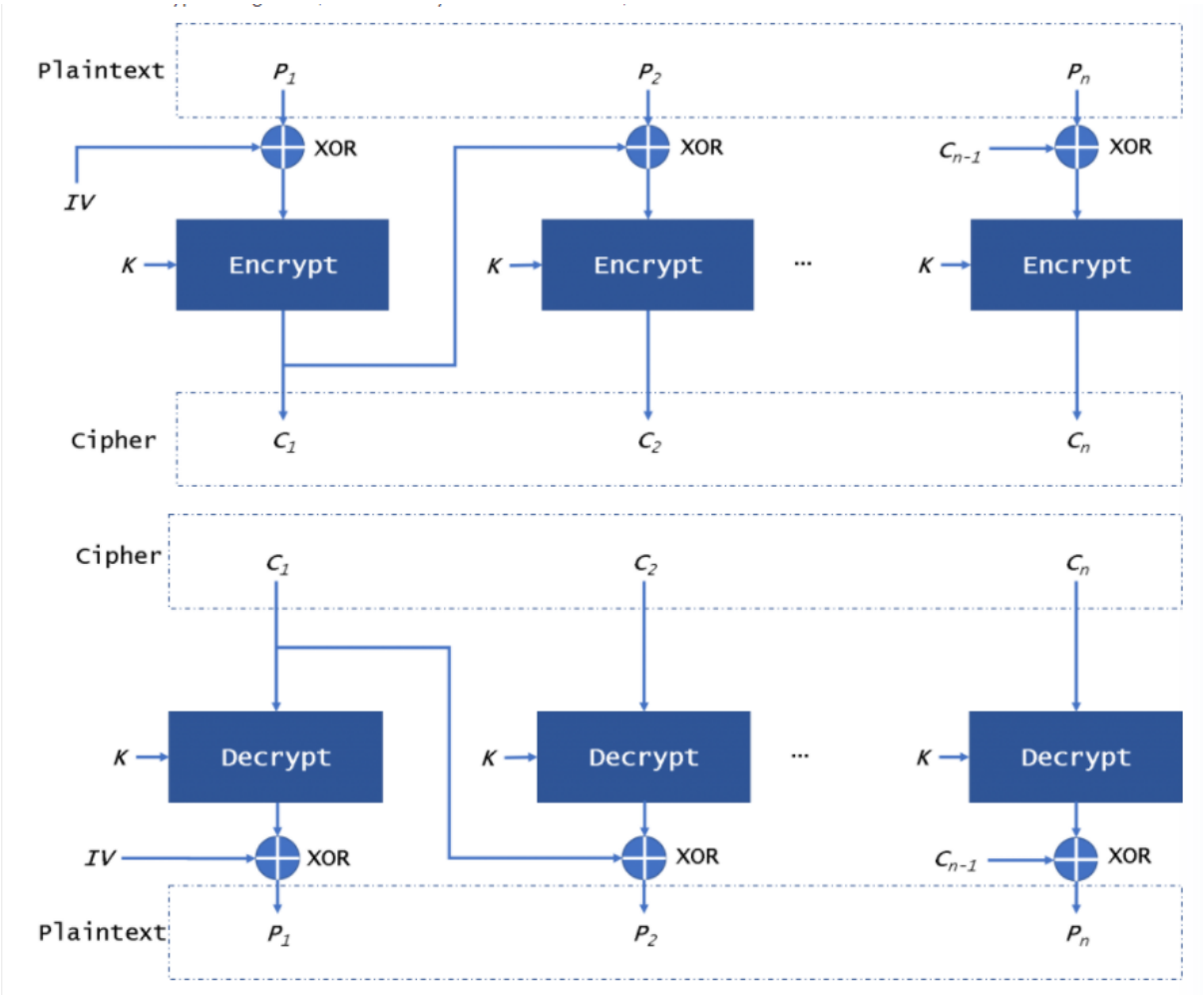
Tryb ECB:

Uzyto gotowego rozwiazania z bilbioteki kryptograficznej traktujac ten tryb jako “czarną skrzynke” do implementacji kolejnych trybów. Tryb ten wymagał także wyrównania tekstu do odowpiedniej długości.



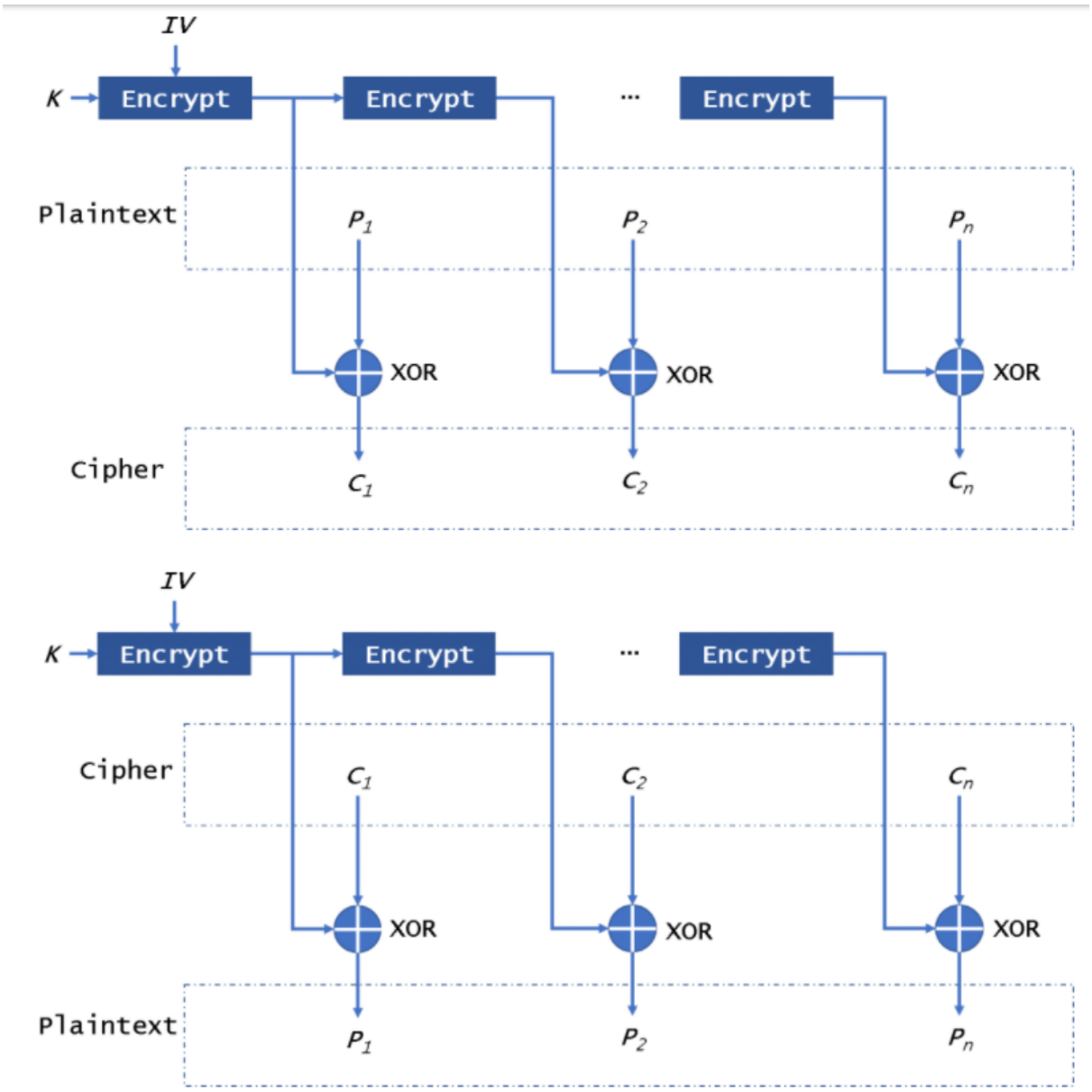
Tryb CBC:

Do implementacji użytu własnej funkcji xor oraz nadmienionej czarnej skrzynki ECB:



Tryb OFB:

Do implementacji użyto własnej funkcji xor oraz nadmienione czarnej skrzynki ECB



Wszelkie testy szyfrowania/deszyfrowania zostały przeprowadzone na inwokacji z “Pana Tadeusza” A.Mickiewicza, Alfabet to znaki mozliwe do zakodowania w UTF-8, Bloki wyswietlane są na ekranie w postaci bloków 16x4 heksadecymalnych liczb.

2.Analiza dostepnych pracy szyfrów blokowych

Tekstem do zaszyfrowania i odszyfrowania był “Pan Tadeusz” A.Mickiewicza w odpowiednich rozmiarach:

- short (pierwsze zdanie inwokacji),
- medium(4 pierwsze zdania inwokacji),
- long(cała inwokacjai)

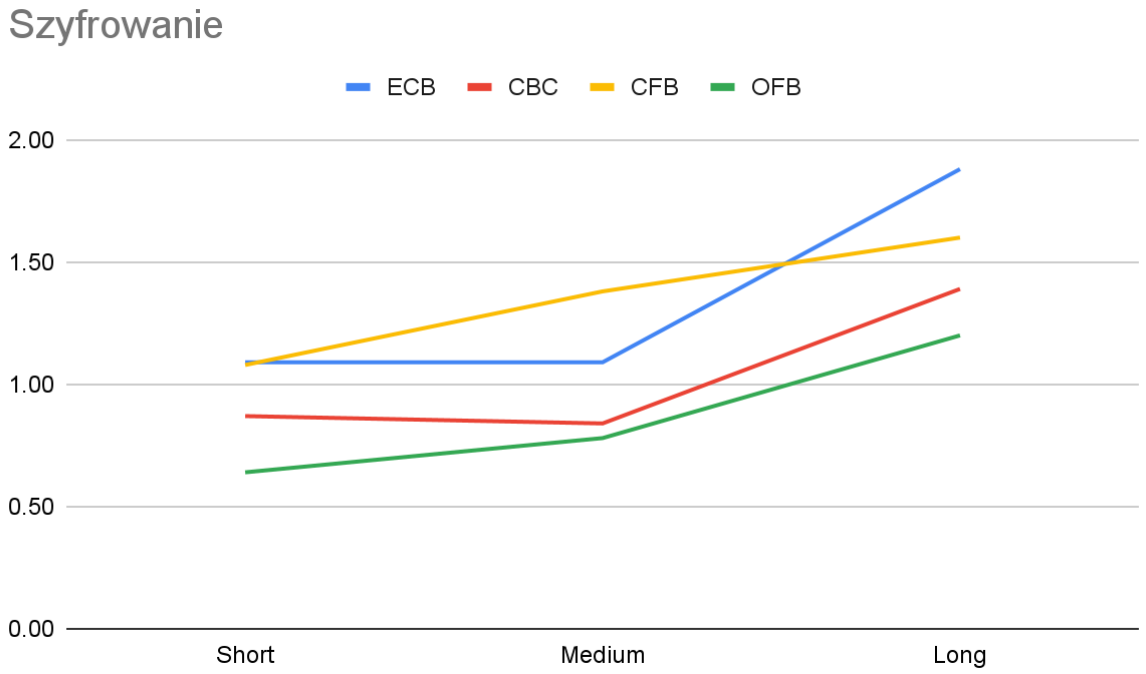
Algorytm szyfrujacy to AES (rozmiar bloku 16 bajtow) w trybach:

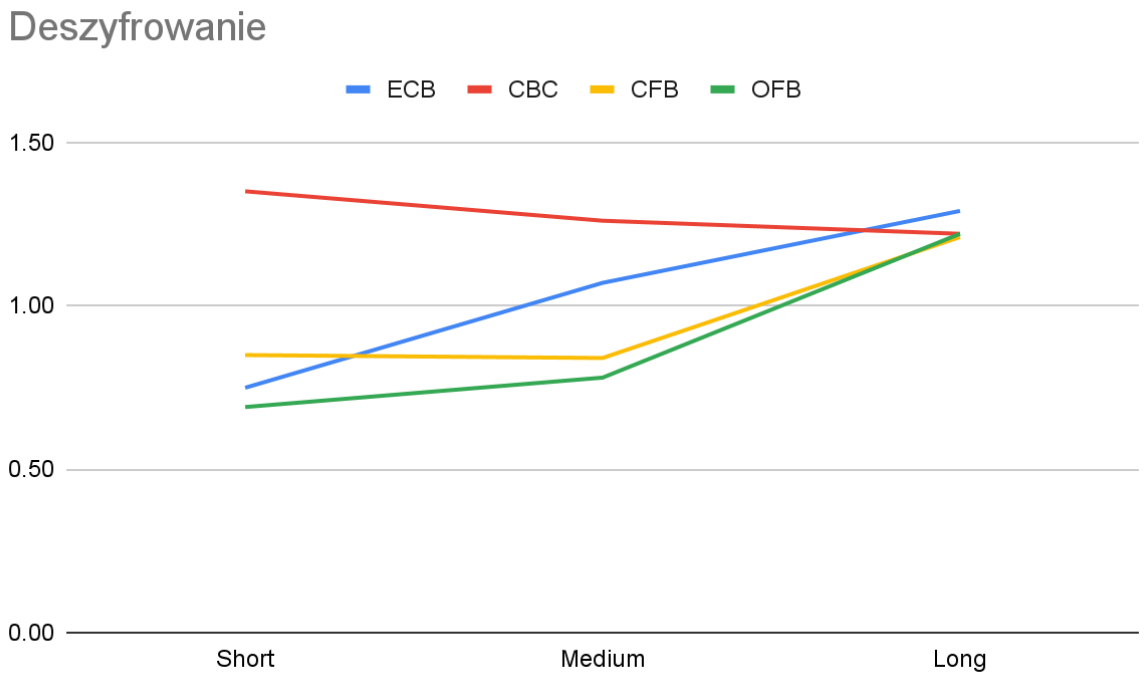
- ECB,
- CBC,
- CFB,
- OFB

Wyniki podane na wykresach zostały podane w milisekundach
Wszystkie wyniki są usrednieniem z 10 prób.
Wszystkie implementacje trybów są częścią biblioteki pycryptodome

Interpretacja:

Czas szyfrowania oraz deszyfrowania rosnie w miare liniowo z wielkoscia wiadomosci. Warto zaznaczyyć że użyty tryb ECB to wariant jednowątkowy a nie obliczany rownolegle, wynika to z tego że przy zrównoleglaniu szyfrowania ECB byłby kilku/kulkunastokrotnie szybszy niz pozostałe działające sekwencyjnie algorytmy. Im wieksza wiadomośc tym wiekszy byłby nasz zysk na zrownolegleniu obliczen trybem ECB.
Pozostałe tryby wykonuja zwykle dodatkowe operacje jak np operator xor co skutkuje lekko wolniejszym działaniem. Żaden jednak nie ma miażdzącej przewagi nad resztą w czasie szyfrowania/deszyfrowania





3. Manipulowanie zaszyfrowanymi blokami:

Zostały użyte własne implementacje trybów (ECB biblioteczne, CBC i OFB własne), Testowane były teksty zawierajace 16 takic hsamych znaków ‘a’, kolejnych znaków alfabetu łacińskiego oraz wersja long inwokacji.

Wyniki:

a) usuniecie całego bloku (usuwany jest 2 blok):
    i) ECB – w deszyfrowanym tekscie brakuje części jednego zdania, poza tym wszystko zgadza sie z orygnalną wiadomością
    ii) CBC – wiadomośc jest nierozszyfrowalna ponieważ kolejne bloki są deszyfrowane na podstawie porzednich bloków a w przypadku braku jednego dostajemy zupełnie inny tekst
    iii) OFB – usuniety blok nie pozwala odczytac poprawnie orygnalnej wiadomosci
b) powielenie bloku (powielany jest 2 blok):
    i) ECB – w deszyfrowanym tekscie znajduja sie 2 takie same zdania, poza tym wszystko zgadza sie z oryginalna wiadomością
    ii) CBC – Ponownie uszkodzenie jednego z bloków prowadzi do braku możliwości odczytania orygnalnej wiadomości
    iii) OFB – ten sam przypadek co CBC
c) zamiana bloków miejscami
    i) ECB – w deszyfrowanym tekscie 2 zdania zostały zamienione miejscami
    ii) CBC – zamiana blokow prowadzi do braku mozliwosci poprawnego deszyfrowania, nie udało sie odtworzyc orygnalnej wiadomosci
    iii) OFB – ten sam przypadek co CBC
d) dodanie nowego bloku (zaszyfrowane zdanie “Dodatkowy blok” dodane przed ostatnim blokiem zaszyfrowanej wiadomosci):
    i) ECB – w deszyfrowanym tekscie na koncu pojawiło sie dodatkowe zdanie rowne poprawnie zdeszyfrowanemu dodanemu blokowo (z paddingiem!)
    ii) CBC – dodanie nowego bloku uniemożliwiło prawidłowe deszyfrowanie wiadomosci
    iii) OFB – ten sam przypadek co CBC
e) zmiana wartosci jednego bajtu(17 bajt zaszyfrowanej wiadomosci jest incrementowany o 1):
    i) ECB – w deszyfrowanym tekscie nastepuje kilka błedów ale dalej jest czytelny
    ii) CBC – ponownie deszyfrowany tekst jest nieczytelny
    iii) OFB – w deszyfrowanym tekscie jedna literka jest zmieniona
f) zamiana bajtow wewnatrz bloku(zamieniany jest bajt 17 z bajtem 18):
    i) ECB – w deszyfrowanym tekscie nastepuje kilka bledów, ale dalej jest czytelny
    ii) CBC – ponownie deszyfrowany tekst jest nieczytelny

iii) OFB – w deszyfrowanym tekscie jedna literka jest zamieniona miejscami z inna

g) usuniecie fragmentu bloku(usuwany jest 22 bajt zaszyfrowanej wiadomosci):
   i) ECB – brak mozliwosci deszyfrowania, niewłasciwa liczba bajtow w bloku prowadzi do błedu deszyfratora
   ii) CBC – ponownie deszyfrowany tekst jest nieczytelny
   iii) OFB – w deszyfrowanym tekscie brakuje jednej litery

Wnioski:

W zależnosci od stopnia modyfikacji tryb ECB jest najmniej podatny na modyfikacje całego bloku z zaszyfrowanej wiadomosci, wynika to z faktu że w tym trybie każdy blok szyfrowany jest niezaleznie od innych więc modyfikacje całego bloku modyfikują tylko odpowiadajace tym blokom fragmenty tekstu jawnego. W przypadku modyfikacji pojedynczych bitów/bajtów jest bardziej skomplikowany przypadek, usuniecie jednego bajtu prowadzi do nierównosci w długosci bloków co nie pozwala deszyfrowac ze wzgledu na wymóg co do wielkosci bloków jakie nałada algorytm

Tryb CBC ze względu na zaleznośc kolejno szyfrowanych bloków od porzedniego zaszyfrowanego bloku cechuje sie najwieksza podatnościa na jakiekolwiek modyfikacje czy to pojedynczego bajtu czy to całego bloku, bardzo łatwo więc sprawić że zaszyfrowana wiadomośc będzie bezpowrotnie niemożliwa do odczytania

Tryb ofb cechuja sie odwrotnie do ECB podatnościa na zmiany całych bloków natomiast modyfikacje wewnątrz jednego bloku zdają sie nie wpływac znacznie na odszyfrowywanie, dzięki temu pojedyncze zamiany bajtów zmienia jedynie kolejnosc liter bądz samą litere, natomiast usunięcie całego bloku nie pozwala na zdeszyfrowac tekstu zaszyfrowanego

Tryb	Plusy	Minusy
ECB	• prosty • szybki • pozwala zrównoleglić obliczenia	• wymaga wyrównywania tekstu • niezbyt bezpieczny dla wiadmości dłuższych niż jeden blok • operacje na pojedynczych bajtach wplywają mocno na wynik deszyfrowania
CBC	• mniej podatny na ataki niż ECB • pozwala zrównoleglić deszyfrowanie	• nie pozwala zrównoleglić szyfrowania • bardzo podatny na jakiekolwiek modyfikacje zaszyfrowanej wiadomości
OFB	• nie wymaga wyrównywania tekstu do wielokrotności wielkości bloku • do deszyfrowania mozna wykorzystać algorytm szyfrujący • zmiany wewnątrz jednego bloku nie wplywają na inne bloki	• Nie pozwlaa na zrównoleglenie obliczeń • Modyfikacje całych bloków w zaszyfrowanym tekście nie pozwalają odczytać oryginalnej wiadomości