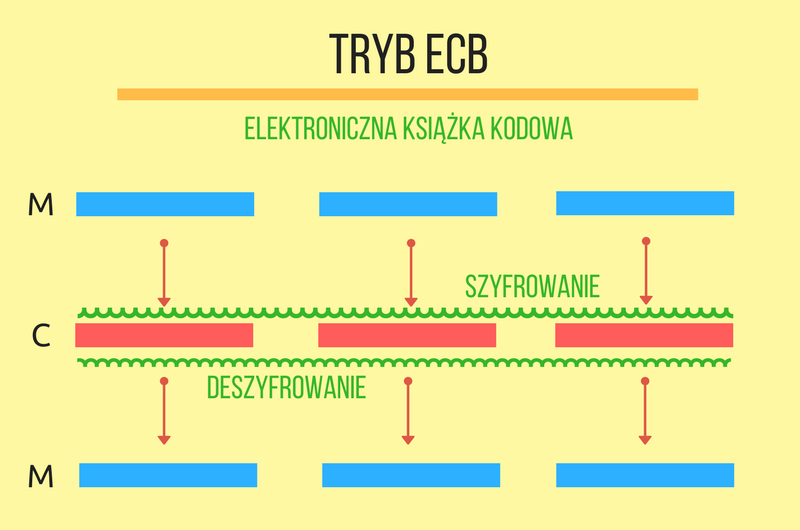
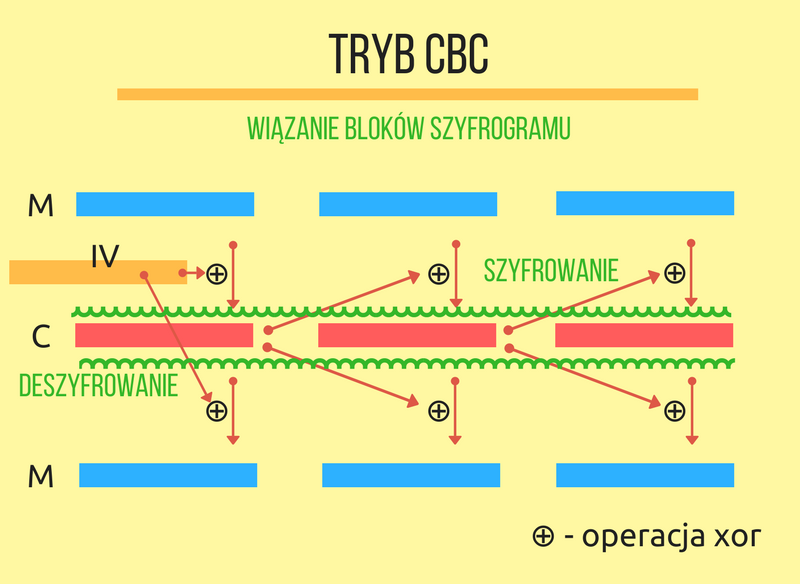
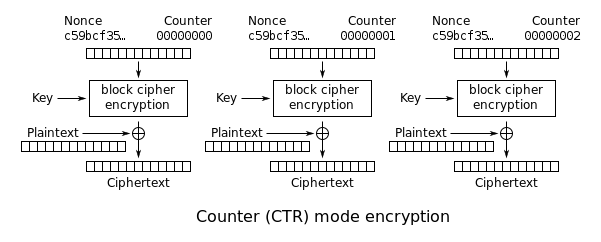
1.







2. Jedną z metod ochrony przed inżynierią wsteczną jest np. obfuskacja czyli tzw. zaciemnianie kodu. Można ją uzyskać poprzez:

* transformację wyglądu (ang. *Layout Transformation*) - zmiany nazw identyfikatorów, zmiana formatowania, usuwanie komentarzy.
* transformację danych (ang. *Data Transformation*) - rozdzielenie zmiennych, konwersja statycznych danych do procedury, zmiana kodowania, zmiana długości życia zmiennej, łączenie zmiennych skalarnych, zmiana relacji dziedziczenia, rozłam/łączenie tablic, zmiana porządku instancji zmiennych / metod / tablic.
* transformację kontroli (ang. *Control Transformation*) - zmiana przebiegu, rozszerzenie warunków pętli, zmiana kolejności komend / pętli / wyrażeń, metody *inline*, ogólnikowe wyrażenia, klonowanie metod.

Możemy również stosować szyfrowanie kluczy i haseł, nie udostępniać użytkownikowi kodu w całości, ograniczać informacje na temat działania programu w celu ochrony przed dezasemblacją kodu.

3. Nie zbadałem szybkości wykonania dla poszczególnych bibliotek.

4. Korzystałem ze stron <https://nvd.nist.gov/vuln>, https://www.cvedetails.com.

5. Wyjątek ten wynika z faktu, że środowisko JDK ma odgórnie narzucone ograniczenia dotyczące długości klucza. Klasa Cipher nie pozwala wykonać enkrypcji kluczem większym niż 128 bity. Ograniczenie to wynika z restrykcji niektórych krajów co do dopuszczalnej wielkości (siły) klucza.

6. ECB *(*electronic code book*)* - najprostszy sposób szyfrowania. Każdy z bloków wiadomości jest kodowany oddzielnie. Każdy blok szyfrogramu jest również deszyfrowany oddzielnie. Bardziej podatny na atak metodą powtórzenia (Replay Attack).

# CBC- eliminuje podstawowe wady trybu ECB tworząc zależność pomiędzy poszczególnymi blokami za pomocą operacji xor*.* Polega na dodawaniu XOR każdego kolejnego bloku tekstu jawnego do poprzednio otrzymanego bloku szyfrogramu. Wynik tego działania jest szyfrowany. Pierwszy blok tekstu jawnego jest dodawany XOR do losowego wektora inicjującego (zwyczajowo oznaczanego jako IV), o takiej samej długości jak długość każdego bloku danych. Jeśli napastnik może przewidzieć jaki wektor zostanie użyty, wtedy szyfrowanie nie jest odporne na ataki z wybranym tekstem jawnym (chosen-plaintext attack)

CTR (znany również jako SIC - Segment Integer Counter) - upodabnia działanie szyfru blokowego do szyfrów strumieniowych. Szyfruje się kolejne wartości stale zwiększającego się licznika, zsumowane z dodatkową liczbą nazywaną nonce (ang. number used once), unikalny numer pełniący taką samą rolę jak wektor inicjujący w poprzednich trybach.

Podobnie, jak w przypadku CBC, należy zmienić klucz szyfrujący po pewnej liczbie wysłanych wiadomości. Można wykazać, że CTR zapewnia większe bezpieczeństwo transmisji i klucz musi być zmieniany rzadziej. Dla szyfru [AES](http://www.crypto-it.net/pl/symetryczne/aes.html?tab=0) ilość bloków tekstu jawnego, po której należy zmienić wykorzystywany sekretny klucz, można w przybliżeniu określić na 264.