1. Zgodnie z informacjami zamieszczonymi na stronie
   http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html
   NIST zatwierdził 14 trybów szyfrów blokowych:
   - 8 trybów poufności (ECB, CBC, OFB, CFB, CTR, XTS-AES, FF1 i FF3)
   - 1 tryb uwierzytelniania (CMAC)
   - 5 trybów połączonych uwierzytelniania i poufności (CCM, GCM, KW, KWP i TKW)
   Z pośród trybów poufności, 5 trybów (ECB, CBC, OFB, CFB, CTR) jest określonych do stosowania z zatwierdzonym algorytmem szyfrowania blokowego takiego jak AES.
   Inne tryby wiązania blokowego, np.:
   BC, PCBC, CBCC, PBC, PFB
2. Myślę, że polityka własna twórców i wykorzystanie danej biblioteki do tworzenia określonych produktów.
3. Pojawiający się wyjątek z faktu, że Java domyślnie ogranicza wielkość sekretnego klucza szyfrującego. Ograniczenie to jest związane z amerykańskim prawem, które ma służyć uniemożliwieniu wykorzystania zbyt silnego szyfrowania w tworzonych aplikacjach.
4. Przykład bezpiecznego przechowywania kluczy algorytmów symetrycznych jest przechowywanie ich w zaszyfrowanych plikach, gdzie bezpieczeństwo zależy od siły użytego hasła. Alternatywą do zaszyfrowanych plików jest zastosowanie komponentów sprzętowych takich jak tokenów lub kart inteligentnych, które blokują dostęp po kliku próbach wprowadzenia nieprawidłowego PINu.
5. Głównie na stronie cert.pl
6. Moim zdaniem lepsze jest zaszyfrować tekst jawny algorytmem symetrycznym, a klucz deszyfrujący algorytmem asymetrycznym. Ponieważ algorytmy asymetryczne są dużo wolniejsze od algorytmów symetrycznych, dlatego stosowanie asymetrycznych do szyfrowania dużej ilości danych jest bezcelowe.