Laboratorium nr 5

Testowanie generatorów pseudolosowych

Liczba punktów: 3 +**2** + **2**
PRK: T-L-5

1. Opis laboratorium

Celem laboratorium jest przetestowanie generatorów pseudolosowych opartych o liniowe rejestry przesuwające LFSR. Wszystkie testy wymagają wygenerowania ciągu o długości 20000 bitów. W ramach tego laboratorium przetestujemy generatory z poprzedniego laboratorium.

Test pokerowy (Poker test)

1) Podzielić badany ciąg na 4-bitowe paczki obejmujące 4 kolejne bity (paczek tych jest 5000). Zliczyć częstości f (i) pojawiania się każdej z możliwych 16 sekwencji 4-bitowych ($0 \le i \le 15$).
2) Obliczyć wielkość X: $X = (16/5000)\ (\Sigma(f(i))^2)-5000$.
3) Wynik testu jest pomyślny wtedy, gdy $2.16 < X < 46.17$.

Test długich podciągów identycznych ciągów (Long runs test)

1) Jeżeli w badanym ciągu istnieje co najmniej jeden podciąg o długości > 26 bitów zawierający same bity o wartości 0 lub same bity o wartości 1, to wynik testu jest negatywny.

Test podciągów identycznych ciągów (Runs test)

1) Zliczyć wszystkie podciągi składające się tylko z bitów o wartości 0, albo tylko z bitów o wartości 1. Podzielić je na sześć grup:
   - pierwszą - zawierającą podciągi o długości 1 bita;
   - drugą - zawierającą podciągi o długości 2 bitów;
   - ….
   - szóstą - podciągi o długości 6 i więcej bitów.
2) Jeżeli liczebność którejkolwiek z sześciu grup podciągów nie mieści się w zakresie podanym w poniższej tablicy, to wynik testu jest negatywny.
3) Osobno zliczamy podciągi składające się z 0 i 1.

1	***2315-2685***
2	***1114-1386***
3	***527-723***
4	***240-384***
5	***103-209***
6+	***103-209***

2. Materiały

1. A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, Chapter 6 Stream Ciphers, http://cacr.uwaterloo.ca/hac/about/chap6.pdf
2. Shift-Register Stream Ciphers, http://www.quadibloc.com/crypto/co040801.htm
3. Ryszard Tanaś, Wykład na temat Szyfrowanie strumieniowe i generatory ciągów pseudolosowych, http://zon8.physd.amu.edu.pl/~tanas/krypt08.pdf
4. Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, http://friedo.szm.com/krypto/AC/ch17/17-06.html
5. NIST SP 800-22, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

   Dotyczące implementacji testu w zadaniu dodatkowym:
6. https://docs.microsoft.com/en-us/archive/msdn-magazine/2013/government-special-issue/test-run-implementing-the-national-institute-of-standards-and-technology-tests-of-randomness-using-csharp
7. https://github.com/DSC-SPIDAL/csharp/blob/master/SalsaTPL/Salsa.CoreTPL/SpecialFunction.cs [C#]
8. http://accord-framework.net/docs/html/T_Accord_Math_Gamma.htm [C#]
9. https://docs.scipy.org/doc/scipy/reference/special.html [Python]

3. Zadania do wykonania

Napisz aplikacje umożliwiającą testowanie generatorów pseudolosowych zgodnie z poniższymi zadaniami. Aplikacja może posiadać interfejs graficzny lub działać w trybie tekstowym.

1) **Zadanie 1** (1 + **2 pkt**) – Zaimplementuj następujące testy losowości wygenerowanych ciągów binarnych zgodnie z specyfikacją FIPS 140-2 tj.:
   i) test pokerowy (*Poker test),*
   ii) test długich podciągów identycznych ciągów (*Long runs test*),
   iii) test podciągów identycznych ciągów (*Runs test*).
   iv) (opcjonalne, **+2 pkt**) zaimplementuj test „Frequency Test within a Block" z specyfikacji NIST SP 800-22 [6] – należy użyć gotowej implementacji funkcji **igamc** wymienionej w opisie testu.
2) **Zadanie 2** (2 pkt) – Przetestuj generatory: Geffe'go, Stop-and-Go, Shrinking Generator z użyciem testów z zadania 1 z użyciem ciągu o długości 20000 bitów.
   i) Czy zaimplementowane przez ciebie generatory spełniają wymagania z testów?
   ii) Jak długość LSFR wpływa na wyniki testów?
   iii) Przygotuj tabelę z wynikami dla testowanych generatorów zbudowanych z LFSR o długości 4 bit, 16 bit i 24 bit.