

Laboratorium nr 4

Generowanie strumieni klucza przy użyciu szyfratorów strumieniowych

Liczba punktów: 3 + 2

PRK: T-L-5

1. Opis laboratorium

Celem laboratorium jest stworzenie oraz przetestowanie aplikacji służącej do badania addytywnych binarnych szyfratorów strumieniowych opartych o liniowe rejestry przesuwające LFSR.

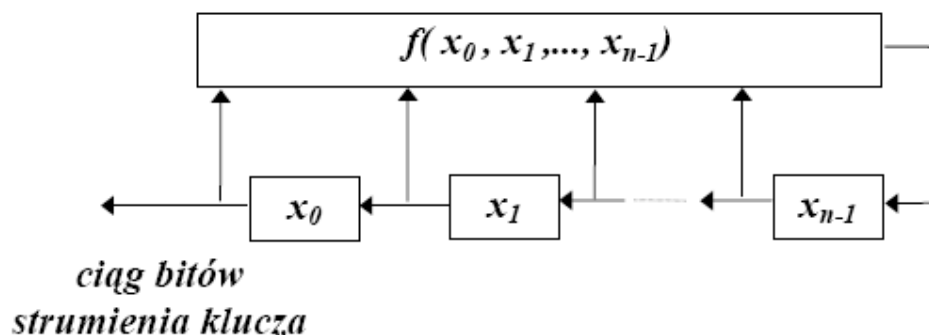
Przykładowy sposób budowy fragmentu interfejsu w zakresie LFSR:

The screenshot shows a web application for LFSR configuration. It has three main sections: 'Długość funkcji (rejestru):', 'Wyraz wolny, funkcja:', and 'Zawartość rejestru:'. Each section contains three rows for LFSR1, LFSR2, and LFSR3. Arrows point from text boxes to specific elements in the interface:

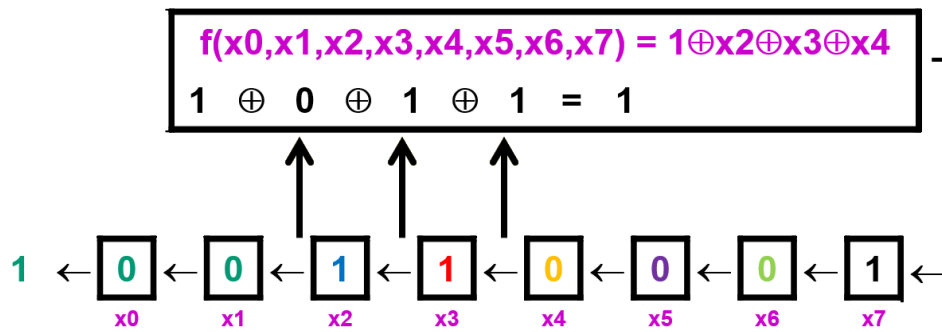
- An arrow points from 'Długość LFSR' to the length input fields.
- An arrow points from 'Wartości funkcji 1 oznacza, że bierzemy dany wyraz pod uwagę. 0 pomijamy. Dla poniższego przykładu: $0 \oplus x^6 \oplus x^7 \oplus x^8 \oplus x^{11} \oplus x^{12}$ ' to the function coefficient inputs.
- An arrow points from 'Początkowa wartość rejestru odpowiadająca długości rejestru LFSR' to the initial value input fields.
- An arrow points from 'Przycisk Inicjuj ustawia w sposób losowy początkowe wartości, aby przy długich rejestrach nie trzeba było ich wpisywać ręcznie' to the 'Inicjuj LFSR' buttons.

Długość funkcji (rejestru):	Wyraz wolny, funkcja:	Zawartość rejestru:	
LFSR1: 13	f(x1): 0 0000011100110	0000011100110	Inicjuj LFSR1
LFSR2: 17	f(x2): 0 00000001100001001	00000001100001001	Inicjuj LFSR2
LFSR3: 19	f(x3): 1 1110101110001011111	1110101110001011111	Inicjuj LFSR3

LFSR:



Przykład:



Należy pamiętać, że wyjście z funkcji f zapisujemy także w wyjściowym strumieniu klucza.

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>

3. Zadania do wykonania

Napisz aplikację umożliwiającą generowanie strumieni klucza. Aplikacja może posiadać interfejs graficzny lub działać w trybie tekstowym.

- 1) **Zadanie 1** (1 + 2 pkt do lab 4) – zaimplementuj liniowy rejestr przesuwający.
- 2) **Zadanie 2** (2 pkt) – używając implementacji liniowego rejestru przesuwającego z zadania 7.1 zaimplementuj 3 różne warianty generatora liczb losowych:
 - i) Generator Geffe'go,
 - ii) Stop-and-Go,
 - iii) Shrinking Generator.

Aplikacja musi pozwalać na ustalanie długości wykorzystywanych rejestrów LFSR oraz współczynników określających liniową funkcję w pętli sprzężenia zwrotnego każdego z rejestrów; parametrami pełniącymi rolę klucza kryptograficznego są początkowe stany użytych w generatorze strumienia klucza rejestrów LFSR (patrz przykład na rysunku na stronie 1).