Linear Feedback Shift Registers

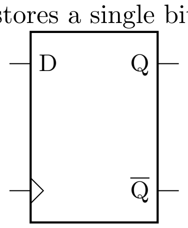
Angelo Panariti

February 15, 2024

1 Introduction to Linear Feedback Shift Registers

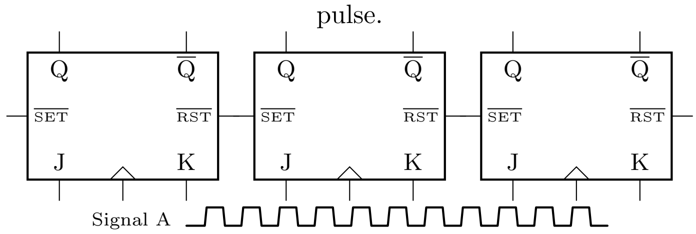
- Goal: Stream ciphers that is small and low power in hardware.
- Example: A5-1 Cipher in GSM
  - consists of 3 LFSRs

Atomic elements: flip-flop, stores a single bit



Clock input (determines when the bit is to be stored or not depending on the input signal).

Let's try to build a PRNG. Connect 3 flip-flops $(1, 0, 0)$. We generate a clock pulse.



1	0	0
0	1	0
1	0	1
1	1	0
1	1	1
0	1	1
0	0	1
1	0	0

1.1 Mathematical description

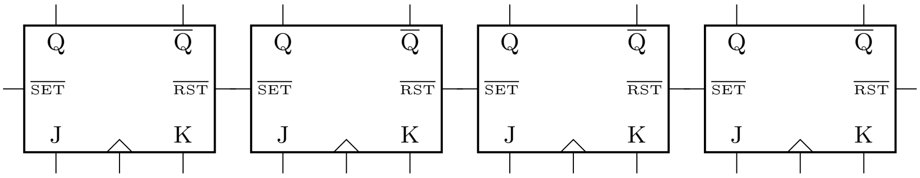
We run into a cycle after group of 7 output bits $(S_0, S_1, S_2, ..., S_7)$.

$$S_3 \equiv S_1 + S_0 \ mod\ 2$$
$$S_4 \equiv S_2 + S_1 \ mod\ 2$$
$$S_{i+3} \equiv S_{i+1} + S_i \ mod\ 2$$

Instead of a period of length 7 bits, we would need a few billion/trillion bits and then it starts to repeat.

2 General LFSRs

m numbers of flip-flops. Every flip-flop becomes a switch. We introduce a multiplier that acts like a switch



$$A \to [X] \to B$$
$$P_i = 1, B = p_i, A = A$$
$$P_i = \emptyset, B = p_i, A = \emptyset$$
$$S_m \equiv S_{m-1}P_{m-1} + S_{m-2}P_{m-2} + ... + S_1P_1 + S_0P_0 \ mod\ 2$$
$$S_{m+1} \equiv S_mP_{m-1} + S_{m-1}P_{m-2} + ... + S_2P_1 + S_1P_0 \ mod\ 2$$

2.1 Equation

$$S_{m+1} = \sum_{i=0}^{m-1} S_{i+j} \cdot P_j \ mod2$$

$$i = 0, 1, 2, 3$$

2.2 Theorem

The maximum period (or sequence length) generated by an LFSR is $2^m - 1$.

2.3 Theorem

Only certain feedback configurations $(p_{m-1}, ..., p_0)$ yield maximum length sequences m = flip-flops, 0 open, 1 close

Ex: $m = 4, \;\; p_3 = p_2 = 0, \; p_1 = p_0 = 1$

Ex: $m = 4, \;\; p_3 = p_2 = p_1 = p_0 = 1$

2.4 Notation

LFSRs are often specified by the polynomial

$$P(x) = x^m + p_{m-1}x^{m-1} + ... + p_1 x + p_0$$

Only LFSRs with primitive polynomials yield maximum length sequences.

3 Attacks against single LFSRs

Given:
- all y_i
- degree m
- $x_0, ..., x_{2m-1}$

1. First step

$$y_i \equiv x_i + s_i \; mod \; 2$$
$$s_i \equiv y_i + x_i \; mod \; 2$$

2. Second step

   Goal: recover S$_{2m}, S_{2m+1}, S_{2m+2}, ...$

Q: What is $p_0, p_1, ..., p_{m-1}$?

Using what we mentioned in Equation [2.1], we can solve for

$$S_m \equiv S_{m-1}P_{m-1} + ... + S_0 P_0 \; mod \; 2$$
$$S_{m+1} = S_m p_{m-1} + ... + S_1 p_0 mod 2$$
$$S_{2m+1} = S_{2m-2} p_{m-1} + ... + S_{m-1} p_0 \; mod 2$$

System of m linear equations with m unknowns. $\rightarrow$ can easily be solved with Gaussian elimination (or matrix inversion). If an attacker knows (at least) $2m$ output values of an LFSR, he can recover the entire LFSR configuration.

3. Third step
   - Using $(p_{m-1}, ..., p_0)$ build LFSR
   - compute $s_0, ..., s_{2m-1}, s_{2m}, s_{2m+1} \cdots$
   - decipher

$$x_i \equiv y_i + s_i \; mod \; 2$$