Determining the Feedback Multipliers in a p-ary Linear Feedback Shift Registers

ANTONIYA TASHEVA, ZHANETA SAVOVA-TASHEVA, BOYAN PETROV, KAMEN STOYKOV
Faculty of Computer Systems and Technologies
Technical University of Sofia
8 Kliment Ohridski blvd. Sofia
BULGARIA
atasheva@tu-sofia.bg

Abstract: - This paper focuses on a method for construction both Galois and Fibonacci p-ary *LFSRs*. Theorems for the transformations of the primitive polynomial generating the extended Galois field $GF(p^L)$ that need to be done in order to receive the values of the multiplier coefficients of the register's feedback polynomial are proven. An algorithm for the transformation is proposed.

Key-Words: - pLFSR, primitive polynomial, feedback polynomial, feedback multipliers, Galois LFSR, Fibonacci LFSR

1 Introduction

Nowadays, stream ciphers are often used for fast encryption over communication channels such as mobile and wireless telephone and Internet. Stream ciphers offer a number of advantages to the user, including high speed encryption, immunity from dictionary attacks, low error propagation and protection against active wiretapping. For synchronous stream ciphers, the keystream is generated independently of the plaintext and the cipher text using a keystream generator, commonly a Pseudo Random Number Generator (*PRNG*) which produces binary Pseudo Random Sequences (*PRSs*).

The goal of the stream cipher cryptosystems design is to design a *PRNG* with good randomness properties, which is equivalently to unpredictability of generated keystream. In order to be unpredictable *PRSs* must have long period, balance and run property, n-tuple distribution, two-level autocorrelation, low-level cross correlation and large linear complexity. Most of those sequences can be generated by means of Linear Feedback Shift Registers (*LFSRs*) and Feedback with Carry Shift Registers (*FCSRs*) [3].

In this paper we will focus on the task of constructing such *LFSRs*. They provide a fast and efficient method for generating a wide variety of pseudo-random sequences both with their hardware and software implementations. Binary *LFSRs* are well studied and discussed but a major application of p-ary *LFSRs* (*pLFSR*) can be found as their long period and good statistical properties of their output sequences are proven.

This paper is organized as follows. First a recall of the *LFSR* architectures is made, their recurrence equations are stated. Next, a theorem for transforming a primitive polynomial into a feedback one used for building a *pLFSR* with Galois architecture is proven. Then, it is proven that the feedback polynomial for a *pLFSR* with Fibonacci architecture has the same order. Finally, a proposition of an algorithm for transforming a primitive polynomial into feedback polynomial is made.

2 pLFSR architectures

A p-ary linear-feedback shift register (pLFSR) is a circuit consisting of L storage units a_i, $0 \le i \le L$-1, regulated by a single clock. Each unit can store an element of the field GF(p). At each clock pulse a linear feedback function defined by the feedback multiplier coefficients q_1, q_2, …, q_L, transforms the current state into a new one.

It is proven that when p = 2 and the feedback multiplier coefficients are defined by a primitive polynomial $q(x)$ generating the field GF(2^L) the output sequence is with maximal period [1], [2], [3], [4], [5].

In terms where p is an odd prime that direct mapping between primitive polynomial coefficients and multipliers of the feedbacks is not applicable. We will prove that when $p \ne 2$, the coefficients of the primitive polynomial $q(x)$ generating the field

GF(p^L) needs additional conversion to ensure that the register generates a maximum length sequence.

First the two underlying *LFSR* architectures will be recalled. Depending on the position of the addition operators modulo p in the scheme *LFSRs* can be characterized as Galois *LFSR* (Internal Feedback *LFSR* or one-to-many) or Fibonacci *LFSR* (External Feedback *LFSR* or many-to-one). [2][3][7][8]

2.1 Galois Architecture

The Galois architecture is shown on figure 1. As one can see the new state of each cell a_i depends on the value in the cell on their left a_{i+1} and the rightmost value a_0 multiplied by the corresponding multiplier q_i. The multiplication is performed also modulo p. Thus the recurrence equation of the register is:

$$\begin{aligned} a_i^{'} &= a_{i+1} + q_{i+1} a_0 \ mod \ p, \\ &\text{for } 0 \le i \le L-2 \\ a_{L-1}^{'} &= q_L a_0 \ mod \ p. \end{aligned} \tag{1}$$

One of the advantages of this architecture relays on the independence of the operations when calculating the new value of each cell. Each clock cycle all multiplication and sum operations can be performed in parallel and thus increasing the speed of execution can be easily achieved.

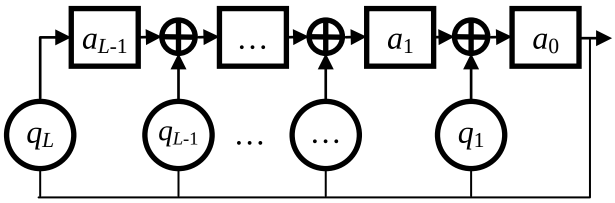

Figure 1 - Galois LFSR

2.2 Fibonacci Architecture

The *pLFSR* Fibonacci architecture is based on the well-known for more than 2000 years Fibonacci number sequence that is a linear recurrent sequence.

The Fibonacci *LFSR* architecture is given in figure 2. The register cells are loaded with initial values a_0, a_1, …, $a_{L\text{-}1}$. Each clock cycle a new value for the leftmost cell is calculated by the formula:

$$\begin{aligned} a_{L-1}^{'} &= \textstyle\sum_{i=1}^{L} q_i a_{L-i} \ mod \ p, \text{ for } t \ge L \\ a_j^{'} &= a_{j+1}, \text{ for } 0 \le j \le L-2 \end{aligned} \tag{2}$$

Here only the multiplications modulo p can be performed in parallel. There will be a second step of summing all results modulo p. In order to achieve speed-up, it is a good practice to choose construction primitive polynomial with fewer elements in order to reduce number of calculations.

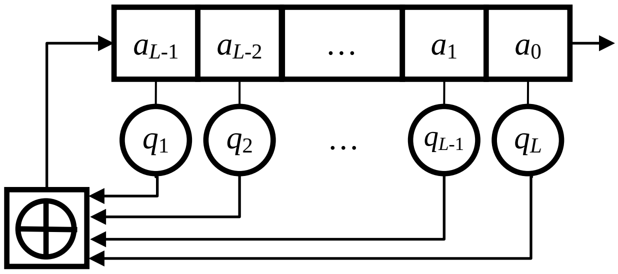

Figure 2 - Fibonacci LFSR

For binary LFSR it is known that it is maximal-length if and only if the corresponding feedback polynomial is primitive. The same can be stated for a *pLFSR* with the correction that feedback coefficients are obtained from the primitive polynomial by some mathematical transformations.

3 Polynomial transformations

Both the Galois and Fibonacci architectures of *pLFSR* will produce maximum length sequence when a primitive polynomial generating field GF(p^L) is used for choosing the feedbacks.

In this section two theorems for the feedback polynomial of a *pLFSR* register with Galois and Fibonacci architecture will be proven.

3.1 Feedback polynomial in Galois architecture

In order to build a *pLFSR* with Galois architecture for a chosen extended Galois field GF(p^L) we need first to choose a primitive polynomial that generates the field. The next step is to find the corresponding to its coefficients multipliers in the register's architecture. The following theorem will set the relation between them.

Theorem 1. The feedback polynomial $q^*(x)$ of a *pLFSR* register with Galois architecture is defined by the formula

$$q^*(x) = \sum_{i=1}^{L} \left(q_i \frac{(p-1)}{q_0} \ mod \ p \right) . x^{i-1} - 1, \tag{3}$$

where $q_i, i = 0, 1, \dots, L$, are the coefficients of the primitive polynomial $q(x)$ generating the field GF(p^L)

$$q(x) = \sum_{i=0}^{L} q_i \, x^i. \tag{4}$$

In this case the generating function of the *pLFSR* output sequence is

$$O(x) = -\frac{h^0(x)}{q(x)}, \tag{5}$$

where $h^0(x)$ is the polynomial defined by the initial state $(a_{L-1},\ \dots,\ a_1,\ a_0)$ of the *pLFSR* register.

Proof of Theorem 1

The first operation that the *pLFSR* register with Galois architecture is performing is addition in GF(p) of $a_0.q_i$ and a_i for $1 \le i \le L-1$. Then a shift operation is performed as all elements are moved one position to the right and the leftmost position a_{L-1} is replaced with $a_0 q_L$. The new *pLFSR* content can be formulated as following:

$$h^1(x) = \sum_{i=1}^{L-1} a_i\, x^{i-1} + a_0 \sum_{i=1}^{L} q_i\, x^{i-1}. \tag{6}$$

Multiplying both sides of the equation by x and adding and subtracting a_0 it is obtained

$$h^1(x)x = a_0 + \sum_{i=1}^{L-1} a_i\, x^{i} + a_0 \sum_{i=1}^{L} q_i\, x^{i} - a_0 \tag{7}$$

The upper equation (7) can be represented like

$$h^1(x)x = h(x) + a_0 q(x), \tag{8}$$

where

$$q(x) = \sum_{i=1}^{L} q_i\, x^{i} - 1 \tag{9}$$

is the feedback polynomial.

Let $q(x)$ is a primitive polynomial in GF(p) and it generates the extended Galois field GF(p^L). Because the primitive element α of the field is a root of $q(x)$ transforming (8) we receive

$$h^1(\alpha)\alpha = h(\alpha). \tag{10}$$

Therefore if $h(\alpha) = \alpha^j$ then $h^1(\alpha) = \alpha^{j-1}$. From this, it can be concluded that the *pLFSR* register with Galois architecture generates the powers of the primitive element α in reverse order. Respectively the output of the register is a sequence of the zero coefficients of those powers. The sequence will have a period $T = p^L - 1$ because the number of non-zero elements in GF(p^L) is $p^L - 1$.

Equation (8) can be generalized for the moment $t + 1$ as:

$$h^{t+1}(x)x^{t+1} = h^t(x)x^t + O_t x^t q(x), \tag{11}$$

where $h^t(x)$ is the *pLFSR* state in the moment t, and O_t – its input at the same moment t, $t = 1, 2, \dots$.

When summing (11) for all moments $t = 0, 1, \dots \infty$ we get

$$\sum_{t=0}^{\infty} h^t(x)x^t = h^0(x) + \sum_{t=0}^{\infty} h^t(x)x^t + O(x)q(x), \tag{12}$$

For the output generation function $O(x) = \sum_{t=0}^{\infty} O_t x^t$ we get

$$O(x) = -\frac{h^0(x)}{q(x)}, \tag{13}$$

where $h^0(x)$ is the initial *pLFSR* state.

As one can see in (9) the free coefficient of the feedback polynomial is -1. When working with field with base $p = 2$ we can use the fact that GF(2) -1 ≡ 1 mod 2 and thus the feedback polynomial can be written as

$$q^*(x) = q(x) = \sum_{i=0}^{L} q_i\, x^i. \tag{14}$$

Generally, in fields GF(p) with any base p the following equation is true

$$-1 \equiv p - 1 \ mod\ p. \tag{15}$$

Therefore, the general representation of the primitive polynomial (4) in GF(p^L) is needed to be transformed so that its free coefficient is equal to (p - 1).

Equation (4) can be rewritten as

$$q(x) = \sum_{i=1}^{L} q_i\, x^{i-1} + q_0. \tag{16}$$

Multiplying both sides of (16) with the coefficient $\frac{(p-1)}{q_0}\ mod\ p$, we get

$$q(x)\frac{(p-1)}{q_0}\ mod\ p = \sum_{i=1}^{L}\left(q_i \frac{(p-1)}{q_0}\ mod\ p\right).x^{i-1} + (p-1). \tag{17}$$

When a primitive polynomial $q(x)$ is multiplied by a constant the result is also primitive [6], therefore the polynomial (17) is also primitive.

We can generalize the feedback polynomial $q^*(x)$ for every p as (3)

$$q^*(x) = \sum_{i=1}^{L}\left(q_i \frac{(p-1)}{q_0}\ mod\ p\right).x^{i-1} - 1, \tag{18}$$

and with that the theorem is proven.

3.2 Feedback polynomial in Fibonacci architecture

In this section it will be proved that the theorem 1 is valid also when the feedback polynomial of a *pLFSR* register with Fibonacci architecture is determined.

Theorem 2. The feedback polynomial $q^*(x)$ of a *pLFSR* register with Fibonacci architecture is defined by formula (3), where $q_i, i = 0, 1, \dots, L$, are the coefficients of the primitive polynomial $q(x)$ generating the field GF(p^L), represented as (4).

In this case the generating function of the *pLFSR* output sequence after the subtraction of the initial register state is (5).

Proof of Theorem **2**

An approach derived from the essence of Fibonacci sequence will be applied. When *pLFSR* with Fibonacci architecture is in operation (2) is calculated as the register's input:

$$a_n = (q_1 a_{n-1} + q_2 a_{n-2} + \cdots + q_L a_{n-L}) \bmod p, \quad (19)$$

for $n \geq L$.

Both sides of (19) are multiplied by x^n and summed for $n \geq L$, then the result is

$$\sum_{n\geq L} a_n x^n = q_1 \sum_{n\geq L} a_{n-1} x^n + q_2 \sum_{n\geq L} a_{n-2} x^n + \cdots + q_L \sum_{n\geq L} a_{n-L} x^n. \quad (20)$$

By denoting the generation function $O(x)$, the polynomial of the initial state $h_0(x)$ and representing the right part of the equation as shifted versions of the output sequence minus a polynomial for every shift, respectively $h_1(x), h_2(x), h_3(x)$ ... the equation is transformed into

$$h_L(x) + \cdots + h_1(x) - h_0(x) = = O(x)(q_L x^L + \cdots + q_2 x^2 + q_1 x - 1). \quad (21)$$

From (21) we can retrieve the value of the output generation function, that is

$$O(x) = \frac{-(h_0(x) - h_1(x) - \cdots - h_L(x))}{q_L x^L + \cdots + q_2 x^2 + q_1 x - 1} = -\frac{h^0(x)}{q(x)}. \quad (22)$$

Where $q(x) = q_L x^L + \cdots + q_2 x^2 + q_1 x - 1$ is the feedback polynomial of the *pLFSR* with Fibonacci architecture, and the polynomial $h^0(x) = h_0(x) - h_1(x) - \cdots - h_L(x)$ depends only on the initial state of the register and has power lower than L.

As one can see from (22) the feedback polynomial has its free coefficient equal to -1. Therefore, a transformation of the primitive polynomial is needed in order to have free coefficient equal to (p - 1) = -1 mod p. That is done by multiplying all coefficients of the primitive polynomial with the constant $\frac{(p-1)}{q_0}\ mod\ p$ and by this the result will be equation (3) and with that the theorem is proven.

4 Algorithm proposition

Based on theorem 1 and 2 an algorithm for finding the feedback multipliers for constructing a *p*-ary LFSR with both Galois and Fibonacci architecture can be constructed as follows.

Algorithm 1. Determining the feedback multipliers of a p-ary LFSR

Input: Primitive polynomial $q(x)$ of degree L, generating the extended Galois field GF(p^L).

Output: Coefficients of a primitive polynomial $q^*(x)$ of degree L, that define the feedback multipliers in a *p*-ary LFSR.

Steps:

1. Calculating the constant $c = \frac{(p-1)}{q_0}\ mod\ p$, where q_0 is the free factor of $q(x)$.
2. For every i = 1, 2, …, L the following is calculated

$$q^*_i = q_i c\ mod\ p.$$

It is important to note that when constructing a *pLFSR* with Galois architecture of the coefficient q^*_1 is positioned rightmost, and q^*_L – leftmost in the scheme and with Fibonacci architecture it is reverse (q^*_L is positioned rightmost, and q^*_1 – leftmost).

5 Conclusion

In this paper we have shown how to construct both Galois and Fibonacci *p*-ary *LFSRs*. When the register is binary, the coefficients of its feedback polynomial can be directly substituted by the coefficients of a primitive polynomial in GF(2^L) and the output sequence is proven to be with maximum length. In controversy, the same is not true when p is an odd prime. Further transformation of the chosen primitive polynomial is needed. We have proven two theorems for both Galois and Fibonacci architectures, that define the transformations of the primitive polynomial generating the extended Galois field *GF*(p^L) in order to receive the values of the multiplier coefficients of the register's feedback polynomial. Finally, a unified algorithm for the transformation in both architectures is proposed.

Acknowledgements

This paper is a result of a project supported by the National Science Fund, Ministry of Education and Science, Bulgaria via FINANCIAL SUPPORT FOR PROJECT OF JUNIOR RESEARCHERS – 2016 [Grant Number DM07/5 – 15.12.2016]

References:

[1] Arnault, François, Thierry Berger, Marine Minier, and Benjamin Pousse. "Revisiting LFSRs for cryptographic

applications." *Information Theory, IEEE Transactions on* Volume 57, Number 12, 2011, pp. 8095-8113.

[2] Gong, Guang. "Sequence analysis." *Lecture Notes for CO739x*, 1999, ps. 137.

[3] M. Goresky, A. Klapper, Fibonacci and Galois Representations of Feedback-With-Carry Shift Registers, *IEEE Trans. on Inform. Theory*, vol. 48, pp. 2826−2836, November 2002.

[4] Goresky, Mark, and Andrew Klapper. *Algebraic Shift Register Sequences*. Cambridge University Press. 2012, ps. 514.

[5] Klein, Andreas. *Stream Ciphers*. Springer-Verlag London. 2013, ps. 399.

[6] Lidl, Rudolf. *Introduction to finite fields and their applications*. Cambridge university press, 1994, ps. 415.

[7] W. Li and X. Yang, "A Parallel and Reconfigurable United Architecture for Fibonacci and Galois LFSR," *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, 2015, pp. 203-206.

[8] G. Mrugalski, J. Rajski and J. Tyszer, Ring generators - new devices for embedded test applications, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 9, pp. 1306-1320, Sept. 2004.