

# Finding Matching Initial States for Equivalent NLFSRs in the Fibonacci and the Galois Configurations

Elena Dubrova

Royal Institute of Technology (KTH), Electrum 229, 164 46 Kista, Sweden  
dubrova@kth.se

**Abstract**—In this paper, a mapping between initial states of the Fibonacci and the Galois configurations of NLFSRs is established. We show how to choose initial states for two configurations so that the resulting output sequences are equivalent.

**Index Terms**—Fibonacci NLFSR, Galois NLFSR, initial state, pseudo-random sequence, stream cipher.

## I. INTRODUCTION

Non-Linear Feedback Shift Registers (NLFSR) are a generalization of Linear Feedback Shift Registers (LFSRs) in which a current state is a non-linear function of the previous state [1]. While the theory behind LFSRs is well-understood, many fundamental questions related to NLFSRs remain open.

The interest in NLFSRs is motivated by their ability to generate pseudo-random sequences which are hard to break with existing cryptanalytic methods [2]. A common approach for encrypting confidential information is to use a *stream cipher* which combines plain text bits with a pseudo-random bit sequence [3]. The resulting encrypted information can be transformed back into its original form only by an authorized user possessing the cryptographic key. While LFSRs are widely used in testing and simulation [4], for cryptographic applications their pseudo-random sequences are not secure. The structure of an  $n$ -bit LFSR can be easily deduced by observing  $2n$  consecutive bit of its sequence [5]. Contrary, an adversary might need  $2^n$  bits of a sequence to determine the structure of the  $n$ -bit NLFSR which generates it [6]. A number of NLFSR-based stream ciphers for RFID and smartcards applications have been proposed, including Achterbahn [7], Grain [8], Dragon [9], Trivium [10], VEST [11], and the cipher [12].

Similarly to LFSRs, an NLFSR can be implemented either in the Fibonacci or in the Galois hardware configuration. In the former, the feedback is applied to the last bit of the register only, while in the latter the feedback can potentially be applied to every bit. The depth of circuits implementing feedback functions in a Galois configuration is usually smaller than the one in the equivalent Fibonacci configuration [13]. This makes the Galois configuration more attractive for stream ciphers where high throughput is important. For example, by re-implementing the NLFSR-based stream cipher Grain [8] from the original Fibonacci to the Galois configuration, one can double the throughput with no penalty in area or power [14].

In [13] it has been shown how to transform a Fibonacci NLFSR into an equivalent Galois NLFSR. While the resulting NLFSRs generate the same sets of output sequences, they follow different sequences of states and normally start from a different initial state. The relations between sequences of states and between initial states of two configurations are studied in this paper. One reason for studying the relation between sequences of states is that some NLFSR-based stream ciphers use not only the output of an NLFSR, but also several other bits of its state to produce a pseudo-random sequence. If a Fibonacci to Galois transformation is applied to an NLFSR-based stream cipher, it is important to know which bits of the state are affected by the transformation in order to preserve the original algorithm. Changing the algorithm is likely to influence the security of a cipher. For the same reason, we need to map the secret key and the initial value (IV) of the original cipher into the corresponding ones of the transformed cipher. Finally, knowing which initial state of the Galois configuration matches a given initial state of the Fibonacci configuration makes possible validating the equivalence of two configurations by simulation.

The paper is organized as follows. Section II gives an introduction to NLFSRs and describes the Fibonacci to Galois transformation. In Section III, we study a relation between the sequences of states generated by two equivalent NLFSRs. Section IV shows how to compute the initial state for the Galois configuration which matches a given initial state of the Fibonacci configuration. Section V concludes the paper and discusses open problems.

## II. BACKGROUND

In this section, we give an introduction to NLFSRs and briefly describe the transformation from the Fibonacci to the Galois configuration. For more details, the reader is referred to [13].

### A. Definition of NLFSRs

A *Non-Linear Feedback Shift Register (NLFSR)* consist of  $n$  binary storage elements, called *bits*. Each bit  $i \in \{0, 1, \dots, n-1\}$  has an associated *state variable*  $x_i$  which represents the current value of the bit  $i$  and a *feedback function*  $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$  which determines how the value of  $i$  is updated. For any

$i \in \{0, 1, \dots, n-1\}$ ,  $f_i$  depends on  $x_{(i+1) \bmod n}$  and a subset of variables from the set  $\{x_0, x_1, \dots, x_i\}$ .

A *state* of an NLFSR is an ordered set of values of its state variables  $(x_0, x_1, \dots, x_{n-1})$ . At every clock cycle, the next state is determined from the current state by updating the values of all bits simultaneously to the values of the corresponding  $f_i$ 's. The *output* of an NLFSR is the value of its 0th bit. The *period* of an NLFSR is the length of the longest cyclic output sequence it produces.

If for all  $i \in \{0, 1, \dots, n-2\}$  the feedback functions are of type  $f_i = x_{i+1}$ , we call an NLFSR the *Fibonacci* type. Otherwise, we call an NLFSR the *Galois* type.

Two NLFSRs are *equivalent* if their sets of output sequences are equivalent.

Feedback functions of NLFSRs are usually represented using the algebraic normal form. The *algebraic normal form* (ANF) of a Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  is a polynomial in  $GF(2)$  of type

$$f(x_0, \dots, x_{n-1}) = \sum_{i=0}^{2^n-1} c_i \cdot x_0^{i_0} \cdot x_1^{i_1} \cdot \dots \cdot x_{n-1}^{i_{n-1}},$$

where  $c_i \in \{0, 1\}$  and  $(i_0 i_1 \dots i_{n-1})$  is the binary expansion of  $i$  with  $i_0$  being the least significant bit. Throughout the paper, we call a term of the ANF a *product-term*.

### B. The transformation from the Fibonacci to the Galois configuration

Let  $f_i$  and  $f_j$  be feedback functions of bits  $i$  and  $j$  of an  $n$ -bit NLFSR, respectively. The operation *shifting*, denoted by  $f_i \xrightarrow{P} f_j$ , moves a set of product-terms  $P$  from the ANF of  $f_i$  to the ANF of  $f_j$ . The index of each variable  $x_k$  of each product-term in  $P$  is changed to  $x_{(k-i+j) \bmod n}$ .

The *terminal bit*  $\tau$  of an  $n$ -bit NLFSR is the bit with the maximal index which satisfies the following condition:

For all bits  $i$  such that  $i < \tau$ ,  $f_i$  is of type  $f_i = x_{i+1}$ .

An  $n$ -bit NLFSR is *uniform* if the following two condition hold:

- (a) all its feedback functions are *singular* functions of type

$$f_i(x_0, \dots, x_{n-1}) = x_{(i+1) \bmod n} \oplus g_i(x_0, \dots, x_{n-1}),$$

where  $g_i$  does not depend on  $x_{(i+1) \bmod n}$ .

- (b) for all its bits  $i$  such that  $i > \tau$ , the index of every variable of  $g_i$  is not larger than  $\tau$ .

**Theorem 1:** [13] Given a uniform NLFSR with the terminal bit  $\tau$ , a shifting  $g_\tau \xrightarrow{P} g_{\tau'}$ ,  $\tau' < \tau$ , results in an equivalent NLFSR if the transformed NLFSR is uniform as well.

### III. THE RELATION BETWEEN SEQUENCES OF STATES

Although a Fibonacci NLFSR and a Galois NLFSR can generate the same output sequence, they follow different sequences of states. Therefore, in order to generate the same output sequence, they normally have to be set to different initial states. In this section we study the relation between sequences of states produced by two equivalent NLFSRs and

derive a basic property which will be used to prove of the main result of the paper.

Let  $s = (s_0, s_1, \dots, s_{n-1})$  be a state of an NLFSR,  $s_i \in \{0, 1\}$ . Throughout the paper, we use  $g_i(s)$  to denote the value of the function  $g_i$  evaluated for the vector  $s$ . We also use  $g_i|_{+m}$  to denote the function obtained from the function  $g_i$  by increasing indexes of all variables of  $g_i$  by  $m$ . For example, if  $g_1 = x_1 \cdot x_2 \oplus x_3$ , then  $g_1|_{+2} = x_3 \cdot x_4 \oplus x_5$ . To simplify the exposition, we do not list variables of a function explicitly if it does not cause any ambiguity, i.e. in the previous example we wrote  $g_1$  instead of  $g_1(x_1, x_2, x_3)$ .

**Lemma 1:** Let  $N_1$  be an  $n$ -bit uniform NLFSR with the terminal bit  $\tau$ ,  $0 < \tau \leq n-1$ , which has the feedback function of type

$$f_\tau = x_{(\tau+1) \bmod n} \oplus g_\tau \oplus p_\tau$$

and let  $N_2$  be an equivalent uniform NLFSR obtained from  $N_1$  by shifting from  $\tau$  to  $\tau-1$  the set of product-terms represented by the function  $p_\tau$ .

If  $N_1$  is initialized to a state  $s = (s_0, s_1, \dots, s_{n-1})$  and  $N_2$  is initialized to the state  $(s_0, s_1, \dots, s_{\tau-1}, r_\tau, s_{\tau+1}, \dots, s_{n-1})$ , where

$$r_\tau = s_\tau \oplus p_\tau|_{-1}(s) \quad (1)$$

then they generate sequences of states which differ in the bit  $\tau$  only.

**Proof:** Suppose that  $N_1$  is initialized to a state  $s = (s_0, s_1, \dots, s_{n-1})$  and  $N_1$  is initialized to a state  $r = (r_0, r_1, \dots, r_{n-1})$ , such that  $r_i = s_i$  for all  $i$  except  $i = \tau$  and  $r_\tau$  is given by (1).

On one hand, for  $N_1$ , the next state is  $s^+ = (s_0^+, s_1^+, \dots, s_{n-1}^+)$  such that

$$\begin{aligned} s_{n-1}^+ &= s_0 \oplus g_{n-1}(s_1, s_2, \dots, s_{\tau-1}) \\ &\dots \\ s_\tau^+ &= s_{\tau+1} \oplus g_\tau(s_0, s_1, \dots, s_{\tau-1}) \oplus p_\tau(s_1, s_2, \dots, s_\tau) \\ s_{\tau-1}^+ &= s_\tau \\ &\dots \\ s_0^+ &= s_1. \end{aligned}$$

Note that, since  $N_1$  is uniform, the functions  $g_{n-1}, g_{n-2}, \dots, g_\tau$  may only depend on variables with indexes between 0 to  $\tau$ . Furthermore,  $g_{n-1}, g_{n-2}, \dots, g_\tau$  cannot depend on the variable  $x_\tau$ , since otherwise  $N_2$  would not be uniform after shifting. For the same reason, the function  $p_\tau$  cannot depend on the variable  $x_0$ .

On the other hand, for  $N_2$ , the next state is  $r^+ = (r_0^+, r_1^+, \dots, r_{n-1}^+)$ , where

$$\begin{aligned} r_{n-1}^+ &= r_0 \oplus g_{n-1}(r_1, r_2, \dots, r_{\tau-1}) \\ &\dots \\ r_\tau^+ &= r_{\tau+1} \oplus g_\tau(r_0, r_1, \dots, r_{\tau-1}) \\ r_{\tau-1}^+ &= r_\tau \oplus p_\tau|_{-1}(r_0, r_1, \dots, r_{\tau-1}) \\ r_{\tau-2}^+ &= r_{\tau-1} \\ &\dots \\ r_0^+ &= r_1. \end{aligned}$$

By substituting  $r_i = s_i$  for all  $i$  except  $i = \tau$ , we get:

$$\begin{aligned} r_{n-1}^+ &= s_0 \oplus g_{n-1}(s_1, s_2, \dots, s_{\tau-1}) \\ &\dots \\ r_{\tau}^+ &= s_{\tau+1} \oplus g_{\tau}(s_0, s_1, \dots, s_{\tau-1}) \\ r_{\tau-1}^+ &= r_{\tau} \oplus p_{\tau|-1}(s_0, s_1, \dots, s_{\tau-1}) \\ &\dots \\ r_0^+ &= s_1. \end{aligned}$$

By substituting  $r_{\tau}$  by (1), we get

$$\begin{aligned} r_{\tau-1}^+ &= s_{\tau} \oplus p_{\tau|-1}(s_0, s_1, \dots, s_{\tau-1}) \oplus p_{\tau|-1}(s_0, s_1, \dots, s_{\tau-1}) \\ &= s_{\tau}. \end{aligned}$$

So, the next state of  $N_2$  is

$$\begin{aligned} r_{n-1}^+ &= s_{n-1}^+ \\ &\dots \\ r_{\tau}^+ &= s_{\tau+1} \oplus g_{\tau}(s_0, s_1, \dots, s_{\tau-1}) \\ r_{\tau-1}^+ &= s_{\tau-1}^+ \\ &\dots \\ r_0^+ &= s_1^+ \end{aligned}$$

i.e. the next states of  $N_1$  and  $N_2$  can potentially differ only the bit position  $\tau$ .

In order to extend this conclusion to a sequence of states, it remains to show that the resulting  $r_{\tau}^+$  can be expressed according to (1). From

$$s_{\tau}^+ = s_{\tau+1} \oplus g_{\tau}(s_0, s_1, \dots, s_{\tau-1}) \oplus p_{\tau}(s_1, s_2, \dots, s_{\tau})$$

we can derive

$$s_{\tau+1} = s_{\tau}^+ \oplus g_{\tau}(s_0, s_1, \dots, s_{\tau-1}) \oplus p_{\tau}(s_1, s_2, \dots, s_{\tau}).$$

Substituting it to the expression of  $r_{\tau}^+$  above and eliminating the double occurrence of  $g_{\tau}(s_0, s_1, \dots, s_{\tau-1})$ , we get

$$r_{\tau}^+ = s_{\tau}^+ \oplus p_{\tau}(s_1, s_2, \dots, s_{\tau})$$

Since  $p_{\tau}(s_1, s_2, \dots, s_{\tau}) = p_{\tau|-1}(s_0^+, s_1^+, \dots, s_{\tau-1}^+)$ , we get

$$r_{\tau}^+ = s_{\tau}^+ \oplus p_{\tau|-1}(s^+)$$

□

As an example, consider the following 4-bit NLFSR  $N_1$ :

$$\begin{aligned} f_3 &= x_0 \oplus x_1 \\ f_2 &= x_3 \oplus x_1 \oplus x_0 x_1 \\ f_1 &= x_2 \\ f_0 &= x_1. \end{aligned}$$

which has the period 15. Suppose we shift the product term  $x_1$  from the bit 2 to the bit 1. Then we get the following equivalent NLFSR  $N_2$ :

$$\begin{aligned} f_3 &= x_0 \oplus x_1 \\ f_2 &= x_3 \oplus x_0 x_1 \\ f_1 &= x_2 \oplus x_0 \\ f_0 &= x_1. \end{aligned}$$

The sequences of states of  $N_1$  and  $N_2$  are shown in the 1st and 2nd columns of Table I. The initial states of  $N_1$  and  $N_2$  are  $(s_3 s_2 s_1 s_0) = (0001)$  and  $(r_3 r_2 r_1 r_0) = (0101)$ , respectively. According to Lemma 1, we have  $r_0 = s_0$ ,  $r_1 = s_1$ ,  $r_2 = s_2 \oplus s_0$ , and  $r_3 = s_3$ . As we can see, these sequences differ in the bit 2 only, which is the terminal bit of  $N_1$ .

TABLE I  
SEQUENCES OF STATES OF THREE EQUIVALENT 4-BIT NLFSRS.

| Galois            |                   | Fibonacci         |
|-------------------|-------------------|-------------------|
| NLFSR $N_1$       | NLFSR $N_2$       | NLFSR $N_3$       |
| $x_3 x_2 x_1 x_0$ | $x_3 x_2 x_1 x_0$ | $x_3 x_2 x_1 x_0$ |
| 0 0 0 1           | 0 1 0 1           | 0 0 0 1           |
| 1 0 0 0           | 1 0 0 0           | 1 0 0 0           |
| 0 1 0 0           | 0 1 0 0           | 0 1 0 0           |
| 0 0 1 0           | 0 0 1 0           | 1 0 1 0           |
| 1 1 0 1           | 1 0 0 1           | 1 1 0 1           |
| 1 1 1 0           | 1 1 1 0           | 0 1 1 0           |
| 1 0 1 1           | 1 1 1 1           | 1 0 1 1           |
| 0 1 0 1           | 0 0 0 1           | 0 1 0 1           |
| 1 0 1 0           | 1 0 1 0           | 0 0 1 0           |
| 1 0 0 1           | 1 1 0 1           | 1 0 0 1           |
| 1 1 0 0           | 1 1 0 0           | 1 1 0 0           |
| 0 1 1 0           | 0 1 1 0           | 1 1 1 0           |
| 1 1 1 1           | 1 0 1 1           | 1 1 1 1           |
| 0 1 1 1           | 0 0 1 1           | 0 1 1 1           |
| 0 0 1 1           | 0 1 1 1           | 0 0 1 1           |

The following property follows trivially from Lemma 1.

*Lemma 2:* Let  $N_1$  be an  $n$ -bit uniform NLFSR with the terminal bit  $\tau$ ,  $0 < \tau \leq n-1$ , which has the feedback function of type

$$f_{\tau} = x_{(\tau+1) \bmod n} \oplus g_{\tau} \oplus p_{\tau}$$

and let  $N_2$  be an equivalent uniform NLFSR obtained from  $N_1$  by shifting from  $\tau$  to  $\tau-1$  the set of product-terms represented by the function  $p_{\tau}$ .

If  $N_1$  is initialized to a state  $s = (s_0, s_1, \dots, s_{n-1})$  and  $N_2$  is initialized to the state  $(s_0, s_1, \dots, s_{\tau-1}, r_{\tau}, s_{\tau+1}, \dots, s_{n-1})$ , such that

$$r_{\tau} = s_{\tau} \oplus p_{\tau|-1}(s), \quad (2)$$

then  $N_1$  and  $N_2$  generate the same output sequence.

As an example, consider the sequences of states of NLFSRs  $N_1$  and  $N_2$  shown in the 1st and 2nd columns of Table I. Since their initial states (0001) and (0101) agree with Lemma 2,  $N_1$  and  $N_2$  generate the same output sequence 100010110100111.

#### IV. THE MAPPING BETWEEN INITIAL STATES

This section presents the main result of the paper.

*Theorem 2:* Let  $N_F$  be an  $n$ -bit Fibonacci NLFSR and  $N_G$  be an equivalent uniform Galois NLFSR with the terminal bit  $0 \leq \tau < n-1$  and the feedback functions of type

$$\begin{aligned} f_{n-1} &= x_0 \oplus g_{n-1} \\ f_{n-2} &= x_{n-1} \oplus g_{n-2} \\ &\dots \\ f_{\tau} &= x_{\tau+1} \oplus g_{\tau} \\ f_{\tau-1} &= x_{\tau} \\ &\dots \\ f_0 &= x_1. \end{aligned} \quad (3)$$

If  $N_F$  is initialized to a state  $s = (s_0, s_1, \dots, s_{n-1})$  and  $N_G$  is initialized to the state  $(s_0, s_1, \dots, s_{\tau}, r_{\tau+1}, r_{\tau+2}, \dots, r_{n-1})$  such that

$$r_i = s_i \oplus g_{i-1}(s) \oplus g_{i-2}|_{+1}(s) \oplus \dots \oplus g_{\tau}|_{+i-\tau-1}(s)$$

for all  $i \in \{n-1, n-2, \dots, \tau+1\}$ , then  $N_F$  and  $N_G$  generate the same output sequence.

**Proof:** From the definition of shifting, we can conclude that if, after the transformation, the Galois NLFSR has feedback functions of type (3), then, the feedback function of the  $n-1$ th bit of the original Fibonacci NLFSR is of type:

$$f'_{n-1} = x_0 \oplus g_{n-1} \oplus g_{n-2}|_{+1} \oplus g_{n-3}|_{+2} \oplus \dots \oplus g_{\tau}|_{+n-1-\tau}.$$

Any uniform Galois NLFSR can be obtained by first shifting all product-terms of the original Fibonacci NLFSR but the ones represented by  $g_{n-1}$  from the bit  $n-1$  to the bit  $n-2$ , then shifting all product-terms but the ones represented by  $g_{n-2}$  from the bit  $n-2$  to the bit  $n-3$ , etc., i.e. using a sequence of  $n-1-\tau$  shiftings by one bit. This means that, at each step, the set of product-terms represented by the function

$$p_{n-1-i} = g_{n-1-i-1}|_{+1} \oplus g_{n-1-i-2}|_{+2} \oplus \dots \oplus g_{\tau}|_{+n-1-i-\tau} \quad (4)$$

is shifted from the bit  $n-1-i$  to the bit  $n-1-i-1$ , for  $i \in \{0, 1, \dots, n-1-\tau-1\}$ . Furthermore, for each  $i \in \{0, 1, \dots, n-1-\tau-1\}$ , by Lemma 2, if the NLFSR before shifting is initialized to some state  $s'$  and the NLFSR after shifting is initialized to the state where the bit  $n-1-i$  has the value  $s_{n-1-i} \oplus p_{n-1-i-1}(s')$  and all other bits have the same values as the corresponding bits of  $s'$ , then two NLFSRs generate the same output sequence.

Therefore, we can conclude that if the original Fibonacci NLFSR  $N_F$  is initialized to the state  $s = (s_0, s_1, \dots, s_{n-1})$  and the NLFSR  $N_G$  obtained using the sequence of  $n-1-\tau$  shiftings by one bit described above is initialized to the state  $(s_0, s_1, \dots, s_{\tau}, r_{\tau+1}, r_{\tau+2}, \dots, r_{n-1})$  such that

$$r_j = \oplus p_j|_{-1}(s)$$

for each  $j \in \{n-1, n-2, \dots, \tau+1\}$  and  $p_j$  is defined by (4), then  $N_F$  and  $N_G$  generate the same output sequence.  $\square$

Since the functions  $g_{n-1}, g_{n-2}, \dots, g_{\tau}$  of a uniform Galois NLFSR depend on variables with indexes between 0 to  $\tau$  only, the following property follows directly from the Theorem 2.

**Lemma 3:** Let  $N_F$  be an  $n$ -bit Fibonacci NLFSR and  $N_G$  be an equivalent uniform Galois NLFSR with the terminal bit  $\tau$ . If both  $N_F$  and  $N_G$  are initialized to any state  $(s_0, s_1, \dots, s_{n-1})$  such that  $s_i = 0$  for all  $i \in \{0, 1, \dots, \tau\}$ , then they generate the same output sequence.

As an example, consider the 4-bit Fibonacci NLFSR  $N_3$  with the feedback functions:

$$\begin{aligned} f_3 &= x_0 \oplus x_1 \oplus x_2 \oplus x_1 x_2 \\ f_2 &= x_3 \\ f_1 &= x_2 \\ f_0 &= x_1 \end{aligned}$$

which is equivalent to the Galois NLFSRs  $N_1$  and  $N_2$  from the previous example. The 3rd column of Table I shows the sequence of states of  $N_3$ . The terminal bits of  $N_1$  and  $N_2$  are 2 and 1, respectively. Therefore, (1000) is used as an initial state (2nd row of Table I), all three NLFSRs generate the same output sequence 000101101001111.

## V. CONCLUSION

In this paper, we establish a relation between sequences of states generated by two equivalent NLFSRs and show how to compute the initial state for the Galois configuration which matches a given initial state of the Fibonacci configuration.

Many fundamental problems related to NLFSRs remain open. Probably the most important one is finding a systematic procedure for constructing NLFSRs with a guaranteed long period. Available algorithms either consider some special cases [15], or applicable to small NLFSRs only [16]. The general problem is hard because there seems to be no simple algebraic theory supporting it. Specifically, so far no analog of a primitive generator polynomial has been found for the nonlinear case.

## REFERENCES

- [1] S. Golomb, *Shift Register Sequences*. Aegean Park Press, 1982.
- [2] A. Canteaut, "Open problems related to algebraic attacks on stream ciphers," in *WCC*, pp. 120–134, 2005.
- [3] M. Robshaw, "Stream ciphers," Tech. Rep. TR - 701, July 1994.
- [4] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Jon Wiley and Sons, New Jersey, 1994.
- [5] J. Massey, "Shift-register synthesis and bch decoding," *IEEE Transactions on Information Theory*, vol. 15, pp. 122–127, 1969.
- [6] E. Dubrova, M. Teslenko, and H. Tenhunen, "On analysis and synthesis of  $(n, k)$ -non-linear feedback shift registers," in *Design and Test in Europe*, pp. 133–137, 2008.
- [7] B. Gammel, R. Göttert, and O. Kniffler, "Achterbahn-128/80: Design and analysis," in *SASC'2007: Workshop Record of The State of the Art of Stream Ciphers*, pp. 152–165, 2007.
- [8] M. Hell, T. Johansson, and W. Meier, "Grain - a stream cipher for constrained environments," [citeseer.ist.psu.edu/732342.html](http://citeseer.ist.psu.edu/732342.html).
- [9] K. Chen, M. Henricken, W. Millan, J. Fuller, L. Simpson, E. Dawson, H. Lee, and S. Moon, "Dragon: A fast word based stream cipher," in *eSTREAM, ECRYPT Stream Cipher Project*, 2005. Report 2005/006.
- [10] C. D. Canniere and B. Preneel, "TRIVIUM specifications," [citeseer.ist.psu.edu/734144.html](http://citeseer.ist.psu.edu/734144.html).
- [11] B. Gittins, H. A. Landman, S. O'Neil, and R. Kelson, "A presentation on VEST hardware performance, chip area measurements, power consumption estimates and benchmarking in relation to the aes, sha-256 and sha-512." Cryptology ePrint Archive, Report 2005/415, 2005. <http://eprint.iacr.org/>.
- [12] B. M. Gammel, R. Göttert, and O. Kniffler, "An NLFSR-based stream cipher," in *ISCAS*, 2006.
- [13] E. Dubrova, "An equivalence preserving transformation from the Fibonacci to the Galois NLFSRs." <http://arxiv.org/abs/0801.4079>.
- [14] S. Mansouri, *Re-Designing Grain Stream Cipher for Higher Throughput*. M. Sc. Thesis, Royal Institute of Technology (KTH), Sweden, 2009.
- [15] J. S. I. Janicka-Lipska, "Boolean feedback functions for full-length non-linear shift registers," *Telecommunications and Information Technology*, vol. 5, pp. 28–29, 2004.
- [16] H. Fredricksen, "A survey of full length nonlinear shift register cycle algorithms," *SIAM Review*, vol. 24, no. 2, pp. 195–221, 1982.