Filtering Modified de Bruijn Sequences with Designated Linear Complexity

Guang Gong
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada
ggong@uwaterloo.ca

Kalikinkar Mandal
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3
kmandal@unb.ca

Abstract

In this preliminary work, we study the construction of filtering modified de Bruijn sequence generators and the linear complexity/span of filtering modified de Bruijn sequences. We study two filtering generator constructions consisting of an NLFSR that generates a modified de Bruijn sequence and a filtering function. In our first construction, the filtering function is based on an LFSR that can be selected so that the filtering sequence can have a chosen attainable linear complexity. Moreover, for a proper choice of an LFSR feedback, the filtering sequence can be another modified de Bruijn sequence with a minimum linear complexity. For our second filtering generator, we present some properties of the filtering functions and experimentally study the filtering functions and the linear span of corresponding filtering modified de Bruijn sequences. Our results show that the filtering function does not always guarantee the high linear complexity of a filtering sequence even when the modified de Bruijn sequence generated by the NLFSR has an optimal linear complexity.

1 Introduction

Pseudorandom sequence or number generators (PRSGs/PRNGs) are at the heart of modern stream cipher constructions (e.g., Grain [16], Trivium [6], SNOW-3/V [8], and ZUC [33]). Feedback shift registers (FSRs) provide an efficient mechanism to generate pseudorandom sequences. There are two types of FSRs, namely linear feedback shift register (LFSR) and nonlinear feedback shift register (NLFSR). The preferred randomness properties of a pseudorandom sequence are long period, balance, equal distribution of tuples, 2-level autocorrelation, low crosscorrelation and high linear span [4, 13, 30]. Well-known binary sequences are maximum length sequences (in short, m-sequences), de Bruijn sequences and modified de Bruijn sequences. De Bruijn and modified de Bruijn sequences have known randomness properties such as long period, balancedness, ideal tuple distribution and high linear complexity [25, 26, 27, 3], and can also be generated by NLFSRs with a minimal storage/length.

A binary de Bruijn sequence is a sequence of period 2^n where each n-tuple occurs exactly once in one period of the sequence. A *modified de Bruijn* sequence is a pseudorandom sequence with period $2^n - 1$ where each *nonzero* n-tuple occurs exactly once in one period of the sequence. A modified de Bruijn sequence is also called a *span* n sequence. m-sequences are a class of span n sequences that can be generated by LFSRs. We interchangeably use the terms modified de Bruijn sequence and span n sequence. The total number of binary de Bruijn sequences of period 2^n (also modified de Bruijn of period $2^n - 1$) is $2^{2^{n-1}-n}$ [5].

There is a one-to-one correspondence between a de Bruijn sequence and a modified de Bruijn sequence, which is as follows. A span n sequence can be constructed from a de Bruijn sequence by removing one zero from the run of zeros of length n, and similarly, a de Bruijn sequence can be formed from a span n sequence by adding one zero to the run of zeros of length $n-1$. From a security point of view, the linear span or linear complexity is an unpredictability property of a sequence. However, the stability of the linear span is crucial. As mentioned in [14], by adding one zero to the run of zeros of length $n-1$ to an m-sequence, the linear span of the resultant de Bruijn sequence becomes high, which varies between $2^{n-1}+n$ and 2^n-1. But, after removing any one zero from the run of zeros of length n from the resultant de Bruijn sequence, it becomes an m-sequence or span n sequence with linear span n. This example suggests to study the linear span property of a modified de Bruijn sequence, instead of de Bruijn sequences, for their use in cryptographic applications such as designing stream ciphers and PRNGs.

There is a large volume of works in the literature that broadly focus on (i) searching de Bruijn sequences and modified de Bruijn sequences by an exhaustive search, e.g., in [20, 7, 31]; (ii) constructions of de Bruijn sequences and modified de Bruijn sequences, for example, by joining two or more cycles using conjugates, e.g., in [10, 29, 17, 18, 28, 19, 23, 1]; and (iii) studying statistical and randomness properties such as linear span of de Bruijn sequences and modified de Bruijn sequences, e.g., in [3, 9, 23, 25, 26, 11]. Till today, except for the m-sequences, the linear span n distribution of NLFSR generated span n sequences is unknown. The works in [22, 21] study the constructions of filtering de Bruijn generators with proven tuple distribution properties.

A classical filtering generator consists of an LFSR and a filtering function where the filtering function is chosen to improve the security properties of the filtering sequence or keystream, including the linear span. For an LFSR-based filtering generator, an upper bound of the linear span of the filtering sequence is proved [32]. For LFSR-based filtering generators, existing literature has focused on improving the security of the filtering sequences with a suitable cryptographic filtering function with properties such as high algebraic degree, algebraic immunity and nonlinearity.

An intuitive methodology to design a secure PRNG or stream cipher is to choose all components with best cryptographic properties. When constructing a PRNG or stream cipher using an NLFSR, although a modified de Bruijn sequence could have a large linear span, it cannot be directly used as a keystream generator, otherwise, the internal state will be output. So a filter function or a finite state machine function needs to be employed to the internal state and the keystream will be the output of this function. In this way, the internal state is masked.

In this work, we consider a generalized filtering generator in which we replace the LFSR by an NLFSR generating a modified de Bruijn sequence and require the filtering sequence to be a modified de Bruijn sequence. We call such generator a *modified de Bruijn or span n generator*. As various randomness properties such as long period, balanceness, and equal distribution of tuples in a modified de Bruijn sequence are inherently offered, our focus is to study the linear span of a filtering modified de Bruijn generator. We ask the following question:

When the linear span of the underlying modified de Bruijn sequence generated by an NLFSR is high, can the filtering function always preserve the high linear span of the filtering sequence?

Unfortunately, our study shows that it is possible that the linear span of filtered modified de Bruijn (MDB) sequences could be dropped, even dropped to the minimum linear span n, i.e., the output filtered modified de Bruijn sequence could be an m-sequence with linear span n. As mentioned above, the main motivation behind studying this question that the filtering functions used an NLFSR-based filtering generator are well-chosen to meet cryptographic requirements.

We present two constructions of filtering modified de Bruijn sequence generators consisting of an NLFSR that generates a modified de Bruijn sequence and a filtering function. For our first construction, we consider an LFSR as a filter function that is chosen in such a way that the filtering sequence can have a chosen attainable linear span. We show another construction of an LFSR for which the output filtering sequence is also a modified de Bruijn sequence with the minimum linear span (i.e., an m-sequence). Our second filtering modified de Bruijn generator consists of an NLFSR and a nonlinear filtering function. We experimentally study the filtering functions and the linear span of filtering modified de Bruijn sequences. Our results show that there exist nonlinear filtering functions that could drop the linear span of the filtering modified de Bruijn sequences.

2 Basic concepts and properties of binary modified de Bruijn sequences

In this section, we first define some notations, basic definitions and properties of de Bruijn or modified de Bruijn n sequences.

Notations. We will use the following notations throughout the paper.

- $\mathbb{F}_2 = \{0, 1\}$ is the Galois field with two elements.
- $\mathbb{F}_{2^n}$ is a finite field of size 2^n defined by a primitive element α that is the root of the primitive polynomial $p(x) = \sum_{i=0}^{n-1} c_i x^{n-1} + x^n, c_i \in \mathbb{F}_2$.
- $\mathrm{tr}_1^m(x) = x + x^2 + x^{2^2} + \cdots x^{2^{m-1}}$ is the trace function from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$.
- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ denotes the residue ring modulo N
- $\mathbb{F}_2^n = \{(x_0, \cdots, x_{n-1}) \,|\, x_i \in \mathbb{F}_2\}$

2.1 Nonlinear feedback shift register sequences

Let $f(x_0, \cdots, x_{n-1})$ be a feedback function defined from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $\mathbf{s} = \{s_i\}$, $s_i \in \mathbb{F}_2$ be a binary sequence generated by f as

$$s_{n+i} = f(s_i, s_{i+1}, \cdots, s_{i+n-1}), i = 0, 1, 2, \cdots .$$

Figure 1 shows a block diagram of the feedback shift register sequence generation. If f is a linear function, then $\mathbf{s}$ is referred to as a linear feedback shift register (LFSR) sequence, otherwise, it is called a nonlinear feedback shift register (NLFSR) sequence. The sequence $\mathbf{s}$ is periodic if and only if the feedback function has the following form $f(x_0, \cdots, x_{n-1}) = x_0 + g(x_1, \cdots, x_{n-1})$ where g is defined from $\mathbb{F}_2^{n-1}$ to $\mathbb{F}_2$ [12]. In this work, we consider only periodic sequences of period $2^n - 1$ or 2^n. The randomness properties of LFSR sequences are well-understood, see Golomb and Gong's book [13].

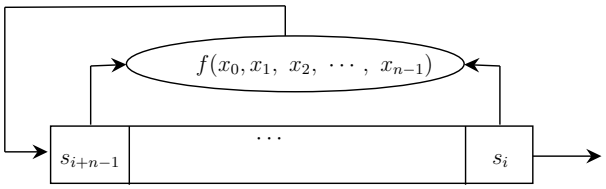

Figure 1: A diagram of an n-stage feeback shift register sequence generation.

Definition 1 (De Bruijn sequence)**.** An NLFSR sequence $\mathbf{s} = \{s_i\}$ over $\mathbb{F}_2$ is called a *de Bruijn* (DB) sequence of order or stage n if the period of $\mathbf{s}$ is 2^n where every n-tuple occurs exactly once in one period.

Definition 2 (Modified de Bruijn or span $\boldsymbol{n}$ sequence)**.** An NLFSR sequence $\mathbf{s} = \{s_i\}$ over $\mathbb{F}_2$ is called a *modified de Bruijn* or *span* n sequence of order n if the period of $\mathbf{s}$ is $2^n - 1$ where every nonzero n-tuple occurs exactly once in one period.

The *linear span or linear complexity* of a sequence is the length of the shortest LFSR that produces the sequence. We also use these two terms interchangeably. The Berlekamp-Massey algorithm can be used to compute the shortest LFSR given a sequence [24].

Property 3. The linear span of a de Bruijn sequence, denoted as LS_{db}, is bounded by $2^{n-1} + n \leq \mathsf{LS}_{db} \leq 2^n - 1$ [3]. On the other hand, the linear span of a span n sequence, denoted as LS_s, is bounded by $n \leq \mathsf{LS}_s \leq 2^n - 2$ [25].

From this property, we call that a span n sequence has the optimal linear span if its linear span is equal to $2^n - 2$.

2.2 Basic properties of binary modified de Bruijn sequences

Sequence and function representations. As our focus is on span n sequences, we recall some properties, more specifically, the trace representation of sequences with period $2^n - 1$ in the following lemma.

Lemma 4. *[13] For a binary sequence* $\mathbf{u} = (u_0, \cdots, u_{2^n-2})$ *of period* $2^n - 1$*, it has the following trace representation:*

$$f(x) = \sum_{r \in I} Tr_1^{n_r}(\gamma_r x^r), \gamma_r \in \mathbb{F}_{2^{n_r}}, \text{ with } u_i = f(\alpha^i), 0 \le i < 2^n - 2 \tag{1}$$

where α *is a primitive element in* $\mathbb{F}_{2^n}$*,* I *is the set consisting of all coset leaders modulo* $2^n - 1$ *and* $Tr_1^{n_r}(x) = x + x^2 + \cdots + x^{2^{n_r-1}}$ *where* $n_r \mid n$*.*

Theorem 5. *(Sequence version) For any binary sequence of period* $2^n - 1$ *generated by a nonlinear feedback shift register of* n *stages, say* $\{a_i\}$*, with linear span* $> n$*, there exists a filtering function* $h(x_0, \cdots, x_{n-1})$ *such that*

$$z_i = h(a_i, a_{i+1}, \cdots, a_{i+n-1}), i = 0, 1, \cdots, 2^n - 2$$

which is a sequence generated by an n*-stage LFSR. In other words, the filtering sequence* $\{z_i\}$ *has linear span* n*, which is the minimum of the linear span of a binary sequence with period* $2^n - 1$*.*

This result is rather surprising at the first glance, since intuitively, we may consider that linear span of filtering sequences should not be decreased. However, the result is contrary. It can decrease to the minimum value of any binary sequence with period $2^n - 1$.

We can have the function version of Theorem 5.

Theorem 6. *(Function version) Let* α *be a primitive element of* $\mathbb{F}_{2^n}$*,* I *be the set consisting of the coset leaders modulo* $2^n - 1$*, and* f *is a mapping from* $\mathbb{F}_{2^n}$ *to* $\mathbb{F}_2$ *with the following univariate polynomial representation*

$$f(x) = \sum_{r \in I} Tr_1^{n_r}(\gamma_r x^r), \gamma_r \in \mathbb{F}_{2^{n_r}}, \tag{2}$$

where at least one of r *satisfies* $n_r = n$ *and* $\gamma_r \neq 0$ *and* $a_i = f(\alpha^i), 0 \le i \le 2^n - 2$*. Then, there exists a function* h *from* $\mathbb{F}_2^n$ *to* $\mathbb{F}_2$ *such that*

$$h((f(x), f(\alpha x), \cdots, f(\alpha^{n-1}x)) = Tr(\beta x^t), \beta \in \mathbb{F}_{2^n}.$$

Discrete Fourier transformation. Let $\mathbf{a} = \{a_i\}$ be a binary sequence of period $N = 2^n - 1$. The Discrete Fourier Transform (DFT) of $\mathbf{s}$ is defined as

$$A_i = \sum_{j=0}^{N} a_i \alpha^{-ij}, i = 0, 1, \cdots, N-1$$

where α is the primitive element of $\mathbb{F}_{2^n}$. The sequence $\mathbf{A} = \{A_i\}$ is called a spectral sequence of $\mathbf{a}$ which is over $\mathbb{F}_{2^n}$. The Inverse Discrete Fourier Transform (IDFT) of $\mathbf{A} = \{A_i\}$ is defined by

$$a_i = \sum_{j=0}^{N} A_i \alpha^{ij}, i = 0, 1, \cdots, N-1$$

where α is the primitive element of $\mathbb{F}_{2^n}$. If we write the spectral sequence $\mathbf{A}$ as a polynomial $A(x) = \sum_{j=0}^{N-1} A_j x^j, A_j \in \mathbb{F}_{2^n}$, then the original sequence can be written as $a_i = A(\alpha^i), i = 0, 1, 2, \cdots, N-1$. According to Blahut's theorem [2], the linear span of $\mathbf{a}$ is the number of non-zero terms in the spectral sequence $\mathbf{A} = \{A_i\}$. The polynomial $A(x)$ can also be written in terms of the trace function, as shown in Fact 7.

Fact 7 (Trace representation from DFT)**.** *[14] The polynomial $A(x)$ can be written as*

$$A(x) = \sum_{i \in I} tr_1^{n_i}(A_i x^i)$$

where i is the cyclotomic coset leaders modulo N, $n_i | n$ is the number of elements in the coset for the cost leader i, and $tr_1^{n_i}(x)$ is the trace function from $\mathbb{F}_{2^{n_i}}$ to $\mathbb{F}_2$.

3 New Constructions

In this section, we present the constructions of new filtering generators based on a span n sequence and filtering functions. We study the linear span of the filtering sequences.

3.1 New filtering construction with designated linear complexity

Let $\mathbf{s} = \{s_i\} = (s_0, s_1, \cdots, s_{2^n-2})$ be a span n sequence of period $N = 2^n - 1$, generated by an NLFSR with a feedback function $f(x_0, x_1, \cdots, x_{n-1})$ as follows:

$$s_{n+i} = f(s_i, s_{i+1}, \cdots, s_{i+n-1}), i = 0, 1, \cdots.$$

Let $S_i = (s_i, s_{i+1}, s_{i+2}, \cdots, s_{i+n-1})$ be the i-th state of the NLFSR. Suppose $q(x) = \sum_{i=0}^{L} q_i x^i, q_i \in \mathbb{F}_2$ be a polynomial of degree L over $\mathbb{F}_2$. We construct a sequence $\mathbf{v} = \{v_i\}$ as

$$v_i = \sum_{j=0}^{L} q_i s_{i+j}, i = 0, 1, \cdots, N-1.$$

When the index $(i+j)$ exceeds N, a modulo N operation is applied. The construction of the sequence $\mathbf{v}$ can be viewed as applying an LFSR filter with the connection polynomial $q(x)$ to the span n sequence $\mathbf{s}$. Figure 2 shows a high-level overview of this generator.

In Figure 2, computing v_i can be viewed as an *expansion-then-compression* operation where the NLFSR expands the state S_i to a sequence $(s_i, s_{i+1}, s_{i+2}, \cdots, s_{i+L})$ of length $L+1$ using the nonlinear feedback function f, and then the LFSR takes the sequence as an input to its state and compresses it to a single bit v_i, which is the LFSR feedback computation. One can also view the process of computing the filtering sequence using an LFSR of length L as applying a filtering function on the internal state of the NLFSR S_i. Mathematically, this process can be written as

$$v_i = h(S_i) = h(s_i, s_{i+1}, s_{i+2}, \cdots, s_{i+n-1}), i = 0, 1, 2, \cdots, 2^n - 2,$$

for some Boolean function h in n variables.

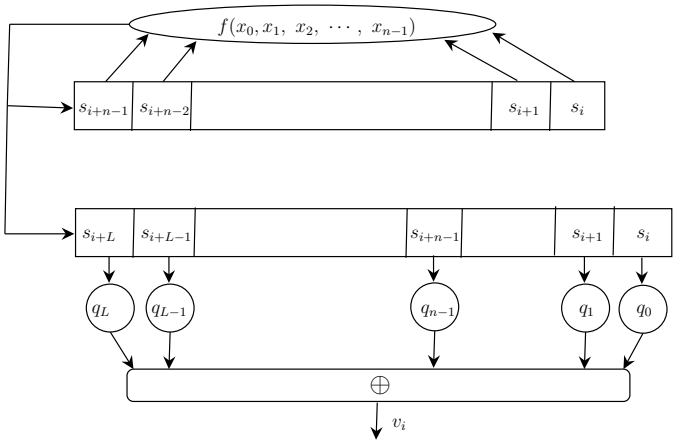

Figure 2: Construction of a filtering generator with designated linear span

Construction 1 (*Filtering sequence with known linear span).* A filtering sequence $\mathbf{v} = \{v_i\}$ with a known linear span is constructed as follows:

1. Let the sequence $\mathbf{s} = \{s_i\}$ be a span n sequence with the linear span $T \leq 2^n - 2$, the maximal value (i.e., the optimal case) where $s_i = A(\alpha^i) = \sum_{i \in I} \mathrm{tr}_1^{n_i}(\gamma_i \alpha^i)$ for some $\gamma_i \in \mathbb{F}_{2^n}$ and $I \subset I_0$ where I_0 is the set of *all* coset leaders modulo $2^n - 1$ and I consisting of those with $\gamma_i \neq 0$.

2. Let α be the primitive root of a primitive polynomial $t(x)$ defining $\mathbb{F}_{2^n}$. Let $t_{\alpha^i}(x)$ be the minimal polynomial of $\alpha^i, i \in I$. We denote

$$p(x) = \prod_{i \in I} t_{\alpha^i}(x).$$

   When the sequence $\mathbf{s}$ has the optimal linear span, we have

$$p(x) = p_0(x) := \frac{x^{2^n - 1} + 1}{x + 1}.$$

3. Select two subsets of coset leaders, denoted by J, $J \subset I$ and $J_0 \subset I_0$ disjoint with I. We compute $u(x) = u_J(x) u_{J_0}(x)$ where $u_k(x) = \prod_{i \in k} t_{\alpha^i}(x)$ where $k \in \{J, J_0\}$. Set $q(x) = p_0(x)/u(x) = \sum_{i=0}^{L} q_i x^i, q_i \in \mathbb{F}_2$ where $L = deg(q(x)) = T - deg(u(x))$ and $2^n - 2 - deg(u(x))$ when $\mathbf{s}$ has the optimal linear span, we have $u(x) = u_J(x)$ as $J_0 = \emptyset$.

4. Compute the filtering sequence $\mathbf{v} = \{v_t\}$ as

$$v_i = \sum_{j=0}^{L} q_j s_{i+j}, i = 0, 1, \cdots, 2^n - 2.$$

We now prove the linear span of the filtering sequence in Construction 1 in Theorem 8. The linear span of $\mathbf{v}$ can be proved by counting the non-zero spectral values in the spectral sequence of $\mathbf{v}$. The proof is similar to that of Theorem 1 in [15].

Theorem 8. *Let* $\mathbf{v} = \{v_i\}$ *be the filtering sequence generated in Construction 1. The linear span of* $\mathbf{v}$ *is* $\mathsf{LS}(\mathbf{v}) = deg(u_J(x))$.

Proof. As $\mathbf{s}$ is a span n sequence, the minimal polynomial of $\mathbf{s}$ for an LFSR is $p(x) = \prod_{i\in I} t_{\alpha^i}(x)$. Thus $\mathbf{v} = \{v_i\}$. According to [15], the relation between the spectral sequences of $\mathbf{v}$ and $\mathbf{a}$ is $V_i = A_i q(\alpha^i)$. Let $K = (I\backslash J) \subset I$. Then $q(\alpha^j) = 0$ and $j = k \cdot 2^i, 0 \le i \le n_k - 1$ for all coset leaders $k \in K$ and n_k is the size for the coset leader k, and $q(\alpha^j) \neq 0$ and $j = k \cdot 2^i, 0 \le i \le n_k - 1$ for all coset leaders $k \in J$. As $A_i \neq 0, 1 \le i \le 2^n - 2$, therefore $V_j = 0$ with $j = k \cdot 2^i, 0 \le i \le n_k - 1$ for all coset leaders $k \in K$ and $V_j \neq 0$ with $j = k \cdot 2^i, 0 \le i \le n_k - 1, k \in J$. Thus, the linear span of $\mathbf{v}$ is $\mathsf{LS}(v) = \sum_{k\in J} n_k = \deg(u_J(x))$. When $\mathbf{s}$ has the optimal linear span, we have $u(x) = u_J(x)$. □

Remark 9. By using this method, the linear span of the filtered sequence is not increasing. But it could decrease to the minimum linear of a span n sequence, which is an m-sequence.

When $\mathbf{s}$ is a span n sequence, in general, for any polynomial $q(x)$ of degree L, the filtering sequence $\mathbf{v}$ is not a span n sequence. We show a construction below that guarantees the filtering sequence is also a span n sequence with the worse linear span. In the following, we restrict ourselves to the case the span n sequence has the optimal linear span case in Construction 1 for simplicity.

Construction 2 (*Filtering span* $\boldsymbol{n}$ *sequence construction).* The steps to construct a filtering sequence that is also a span n sequence are as follows:

1. In Construction 1, select a coset leader $r \in I$ with $\gcd(r, 2^n-1)$. Set $q(x) = p(x)/t_{\alpha^r}(x) = \sum_{i=0}^{L} q_i x^i, q_i \in \mathbb{F}_2$ where $L = \deg(q(x)) = 2^n - 2 - n$.
2. Compute the filtering sequence $\mathbf{v} = \{v_t\}$ as

$$v_i = \sum_{j=0}^{L} q_j s_{i+j}, i = 0, 1, \cdots, 2^n - 2.$$

The filtering sequence $\mathbf{v} = \{v_i\}$ is an m-sequence that can be generated by $t_{\alpha^r}(x)$.

Proposition 10. *The number of different LFSR filtering polynomials* $q(x)$ *for which the sequence* $\mathbf{v}$ *in Construction 2 is an m-sequence, is* $\frac{\phi(2^n-1)}{n}$.

Proof. The proof follows from the fact that the number of coset leaders r with coset size n and $\gcd(r, 2^n - 1) = 1$ is $\frac{\phi(2^n-1)}{n}$. □

Construction 3 (*Computing* $\boldsymbol{h}$*).* The trace representation of h is computed as follows.

1. The mapping h from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is defined as

$$\begin{aligned} h(s_0, s_1, \cdots, s_{n-1}) &\to v_0 \\ h(s_1, s_2, \cdots, s_n) &\to v_1 \\ \vdots \qquad &\qquad \vdots \\ h(s_{2^n-2}, s_0, \cdots, s_{n-2}) &\to v_{2^n-2} \end{aligned}$$

2. Let α be the primitive root of a primitive polynomial $t(x)$ defining $\mathbb{F}_{2^n}$. Apply the DFT on the above mapping to obtain the trace representation of h (using Fact 7).

We now give an example for this constriction.

Example 1. Let $n = 4$. Let α be a root of $t(x) = x^4 + x + 1$ in $\mathbb{F}_{2^4}$. Consider the feedback function $f(x_0, x_1, x_2, x_3) = x_0 + x_1 + x_2 + x_1x_2$ over $\mathbb{F}_2$. The NLFSR corresponding to f generates a span n sequence $\mathbf{s} = 111100010110100$ of period 15. The trace representation of $\mathbf{s}$ is

$$A(x) = \mathrm{tr}(\alpha^{13}x) + \mathrm{tr}(x^3) + \mathrm{tr}(x^5) + \mathrm{tr}(\alpha^{10}x^7).$$

Let $t_{\alpha^i}(x)$ be the minimal polynomial of $\alpha^i, i \in I = \{1, 3, 5, 7\}$. Then, $p(x) = \prod_{i\in I} t_{\alpha^i}(x) = t_\alpha(x)t_{\alpha^3}(x)t_{\alpha^5}(x)t_{\alpha^7}(x)$. If $r = 7$, then $q(x) = p(x)/t_{\alpha^r}(x) = \sum_{i=0}^{L} q_i x^i, q_i \in \mathbb{F}_2, L = \deg(q(x)) = 10$ where

$$q(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1.$$

The filtering sequence $\mathbf{v} = \{v_i\}$ is computed as

$$v_i = s_i + s_{i+1} + s_{i+2} + s_{i+4} + s_{i+5} + s_{i+8} + s_{i+10}, i = 0, 1, \cdots, 14.$$

where $\mathbf{v} = 001000111101011$, which is an m-sequence of period $2^4 - 1 = 15$ for $t_{\alpha^7}(x) = 1 + x^3 + x^4$. The filtering function h for the LFSR filtering is given by

$$h(x) = \mathrm{tr}(\alpha^2 x) + \mathrm{tr}(\alpha^3 x^3) + \mathrm{tr}(\alpha^7 x^7).$$

Remark 11. According the DFT, we may explicitly obtain the trace representation of the the filtering h in Construction 3 as follows. Let $f(x_0, x_1, \cdots, x_{n-1})$ be a feedback function of a modified de Bruijn sequence $\mathbf{s} = \{s_i\}$ generated by an FSR. Let $S_i = (s_i, s_{i+1}, \cdots, s_{i+n-1})$ be the state of the FSR. Suppose σ is the mapping that uniquely maps S_i to an element of $\mathbb{F}_{2^n}$, i.e., $\sigma : S_i \to \alpha^i, i = 0, 1, 2, \cdots, 2^n - 2$. Then the following function h is given by

$$\begin{aligned} h(\beta_i) &= \mathrm{tr}(\sigma^d(S_i)) \\ &= \mathrm{tr}(\beta\alpha^{di}), \end{aligned}$$

where $\sigma^d(S_i) = \alpha^{di}$ and $\beta = q(\alpha^d)$ and the coset of d has the full size modulo $2^n - 1$.

Iteratively computing the LFSR filtering sequence. For simplicity, we assume that the NLFSR generating a span n sequence has the optimal linear span. In Construction 2, the filtering polynomial $q(x)$ is the product of a set of minimal polynomials, say $q(x) = p_1(x)p_2(x)\cdots p_m(x)$. Then, computing $\mathbf{v}$ by the LFSR filter with connection polynomial $q(x)$ is equivalent to computing the filtering sequence iteratively by an LFSR with connection polynomial $p_i(x)$ for $i = 1, 2, \cdots, m$. That is, computing $\mathbf{v} = q(\mathcal{L})\mathbf{s}$ is equivalent to computing $\mathbf{v}_i = p_i(\mathcal{L})\mathbf{v}_{i-1}, i = 1, 2, \cdots, m$ where $\mathbf{v}_0 = \mathbf{s}$, $\mathbf{v} = \mathbf{v}_m$ and $\mathcal{L}$ is the left shift operator. In other words, applying the LFSR filter with the connection polynomial $p_i(x)$ on $\mathbf{v}_{i-1}$ drops the linear span by the degree of $p_i(x)$. This way an LFSR filter can be chosen so that a filtering sequence with the required linear span is achieved.

3.2 Filtering modified de Bruijn sequence generators

Motivated by the construction of the filtering generator in Section 3.1, we consider another filtering generator consisting of an NLFSR with a feedback function $f(x_0, x_1, \cdots, x_{n-1})$ generating a modified de Bruijn sequence and a Boolean filtering function g in n variables. The binary modified de Bruijn sequence $\mathbf{s} = \{s_i\}$ is generated as

$$s_{i+n} = f(s_i, s_{i+1}, \cdots, s_{i+n-1}), i \geq 0.$$

The *filtering sequence* $\mathbf{b} = \{b_i\}$ is generated from $\mathbf{s}$ using the filtering function g as

$$b_i = g(s_i, s_{i+1}, \cdots, s_{i+n-1}), i \geq 0.$$

When $\mathbf{s}$ has the period $2^n - 1$, the sequence $\mathbf{b}$ also will have the period $2^n - 1$. We desire the filtering sequence $\mathbf{b} = \{b_i\}$ to be a modified de Bruijn sequence, other than an m-sequence. We call such generator as a filtering modified de Bruijn or span n generator. Figure 3 shows a diagram of a filtering span n generator. Example 2 shows the existence of filtering functions for which the filtering sequence is also a modified de Bruijn sequence, other than an m-sequence.

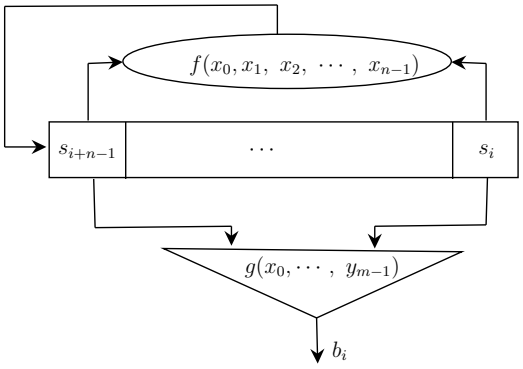

Figure 3: A block diagram of a filtering span n generator

Example 2. Let $n = 4$. Consider the feedback function $f(x_0, x_1, x_2, x_3) = x_0 + x_2 + x_1x_2 + x_1x_3$. The following recurrence relation with the initial state $(1, 1, 1, 1)$ generates the span n sequence $\mathbf{a} = 111101000101100$ with linear span 14:

$$a_{i+4} = f(a_i, a_{i+1}, a_{i+2}, a_{i+3}) = a_i + a_{i+2} + a_{i+1}a_{i+2} + a_{i+1}a_{i+3}, i \geq 0.$$

The filtering function $g(x_0, x_1, x_2, x_3) = x_0 + x_1 + x_1x_3$ on $\mathbf{a}$ produces a filtering sequence $\mathbf{b} = 101111000110100$, which is also a span n sequence and the linear span is 12.

For a suitable choice of a feedback function f and a filtering function g, both sequences $\mathbf{s}$ and $\mathbf{b}$ are span n sequences of period $2^n - 1$. In Proposition 12, we list some (trivial) filtering functions g that generates a span n sequence when the NLFSR generate a span n sequence. The proof is straightforward. So, we omit it.

Proposition 12. *Let f be a feedback function in n variables that generates a span n sequence $\mathbf{s} = \{s_i\}$ of period $2^n - 1$. Let g be a filtering function used to generate a filtering span n sequence $\mathbf{b} = \{b_i\}$. The following filtering functions g generates shift equivalent span n sequences:*

- *$g(x_0, x_1, \cdots, x_{n-1}) = x_i, 0 \leq i \leq n-1$ produces a span n sequence $\mathbf{b}$ with $\mathbf{b} = \mathcal{L}^i(\mathbf{a})$ where $\mathcal{L}$ is the left shift operator.*
- *$g(x_0, x_1, \cdots, x_{n-1}) = f(x_0, x_1, \cdots, x_{n-1})$ produces a span n sequence $\mathbf{b}$ with $\mathbf{b} = \mathcal{L}^n(\mathbf{a})$.*

4 Experimental Results

In Construction 2, we have seen that some filtering functions constructed from an LFSR minimal polynomial reduces the linear complexity to the minimum. This inspires us to understand the linear span of filtering span n sequences for different filtering functions.

Experimental parameters. We perform an experiment to exhaustively search for all feedback functions that generate span n sequences for $n = 4$ and 5, and partially for $n = 6$. For $n = 4$, we consider all filtering functions (i.e., 2^{15} Boolean functions), and for each span n sequence, we generate the filtering sequences. Our experimental results show that, for each span n sequence $\mathbf{s}$, out of 2^{15} filtering sequences, only 240 filtering sequences are span n sequences, which include all shift equivalent span n sequences of period 15. The total number of shift distinct span n sequences are 16. In Table 1, we present the distribution of the filtering functions that generate (shift distinct) span n sequences with different linear span values. Similarly, for $n = 5$, we checked for several span n sequences that, out of 2^{31} filtering sequences (as there are 2^{31} filtering functions), only 63488 ($= 2048 \times 31$) filtering sequences are span n sequences, which also includes all shift equivalent span n sequences of period 31. The total number of shift distinct span n sequences are 2048. Our experimental results in Table 1 also validates Proposition 10 for Construction 2.

In [25], Mayhew and Golomb (IEEE-IT 1990) studied the linear span distribution of span n sequences and presented experimental results on the number of span n sequences for different linear span values for $n = 4, 5$, and 6. In Table 1, we summarize their results on the number of span n sequences for an easy reference.

Table 1: Distribution of filtering functions that generate (shift distinct) span n sequences and their linear span

$n = 4$			$n = 5$		
Linear span	Number of filtering	Number of span n [25]	Linear span	Number of filtering	Number of span n [25]
4	2	2	5	6	6
12	4	4	15	10	10
14	10	10	20	4	4
			25	306	306
			30	1722	1722

We make the following conjecture on the distribution of the filtering functions.

Conjecture 1. For $n \geq 4$, in a filtering modified de Bruijn sequence generator, the distribution of filtering functions g that map a span n sequence $\mathbf{s}$ to another filtering span n sequence $\mathbf{b}$ is the same as the linear span distribution of span n sequences.

5 Conclusion and Future Work

In this work, we studied the construction of filtering modified de Bruijn sequence generators from the linear complexity point of view. We presented two filtering generator constructions that can give a chosen linear complexity. We experimentally studied the filtering functions and the linear span of corresponding filtering modified de Bruijn sequences. Our results show that the filtering function does not always preserve the high linear complexity of the filtering sequence even when the underlying modified de Bruijn sequence has a high linear complexity. This phenomenon suggests that for cryptographic applications, the filtering functions, in addition to well-known crypto properties such as high algebraic degree and algebraic/correlation/spectral immunity, should be chosen carefully to prevent from dropping the linear complexity drastically.

As a future work, we have been continuing to our work to prove Conjecture 1 and also study other crypto properties of the filtering functions that give filtering span n sequences.

References

[1] S. R. Blackburn, T. Etzion, and K. G. Paterson. Permutation polynomials, de bruijn sequences, and linear complexity. *Journal of Combinatorial Theory, Series A*, 76(1):55–82, 1996.

[2] R. E. Blahut. Theory and practice of error control codes. *Addison-Wesley*, 1983.

[3] A. H. Chan, R. A. Games, and E. L. Key. On the complexities of de bruijn sequences. *Journal of Combinatorial Theory, Series A*, 33(3):233–246, 1982.

[4] L. Chen and G. Gong. *Communication system security*. CRC press, 2012.

[5] N. G. De Bruijn. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, 49(7):758–764, 1946.

[6] C. De Cannière. Trivium: A stream cipher construction inspired by block cipher design principles. In S. K. Katsikas, J. López, M. Backes, S. Gritzalis, and B. Preneel, editors, *Information Security*, pages 171–186, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[7] E. Dubrova. A list of maximum-period nlfsrs, 2012.

[8] P. Ekdahl, T. Johansson, A. Maximov, and J. Yang. A new snow stream cipher called snow-v. *Cryptology ePrint Archive*, 2018.

[9] T. Etzion. Linear complexity of de bruijn sequences-old and new results. *IEEE Transactions on Information Theory*, 45(2):693–698, 1999.

[10] H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM review*, 24(2):195–221, 1982.

[11] R. Games and A. Chan. A fast algorithm for determining the complexity of a binary sequence with period 2^n (corresp.). *IEEE Transactions on Information Theory*, 29(1):144–146, 2006.

[12] S. W. Golomb. *Shift Register Sequences*. Aegean Park Press, USA, 1981.

[13] S. W. Golomb and G. Gong. *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

[14] G. Gong. Randomness and representation of span n sequences. In S. W. Golomb, G. Gong, T. Helleseth, and H.-Y. Song, editors, *Sequences, Subsequences, and Consequences*, pages 192–203, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[15] G. Gong, S. Ronjom, T. Helleseth, and H. Hu. Fast discrete fourier spectra attacks on stream ciphers. *IEEE Transactions on Information Theory*, 57(8):5555–5565, 2011.

[16] M. Hell, T. Johansson, A. Maximov, and W. Meier. *The Grain Family of Stream Ciphers*, pages 179–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[17] C. Li, X. Zeng, C. Li, and T. Helleseth. A class of de Bruijn sequences. *IEEE Transactions on Information Theory*, 60(12):7955–7969, 2014.

[18] C. Li, X. Zeng, C. Li, T. Helleseth, and M. Li. Construction of de Bruijn sequences from lfsrs with reducible characteristic polynomials. *IEEE Transactions on Information Theory*, 62(1):610–624, 2015.

[19] K. Mandal and G. Gong. Cryptographically strong de Bruijn sequences with large periods. In *Selected Areas in Cryptography: 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers 19*, pages 104–118. Springer, 2013.

[20] K. Mandal and G. Gong. Generating good span n sequences using orthogonal functions in nonlinear feedback shift registers. *Open Problems in Mathematics and Computational Science*, pages 127–162, 2014.

[21] K. Mandal and G. Gong. On ideal t-tuple distribution of orthogonal functions in filtering de bruijn generators. *Advances in Mathematics of Communications*, 16(3):597–619, 2022.

[22] K. Mandal, B. Yang, G. Gong, and M. Aagaard. On ideal t-tuple distribution of filtering de bruijn sequence generators. *Cryptography Commun.*, 10(4):629–641, jul 2018.

[23] K. Mandal, B. Yang, G. Gong, and M. Aagaard. Analysis and efficient implementations of a class of composited de Bruijn sequences. *IEEE Transactions on Computers*, 69(12):1835–1848, 2020.

[24] J. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[25] G. Mayhew and S. Golomb. Linear spans of modified de Bruijn sequences. *IEEE Transactions on Information Theory*, 36(5):1166–1167, 1990.

[26] G. L. Mayhew. Weight class distributions of de Bruijn sequences. *Discrete Mathematics*, 126(1):425–429, 1994.

[27] G. L. Mayhew and S. W. Golomb. Characterizations of generators for modified de Bruijn sequences. *Advances in Applied Mathematics*, 13(4):454–461, 1992.

[28] J. Mykkeltveit, M.-K. Siu, and P. Tong. On the cycle structure of some nonlinear shift register sequences. *Information and control*, 43(2):202–215, 1979.

[29] J. Mykkeltveit and J. Szmidt. On cross joining de Bruijn sequences. *Contemporary Mathematics*, 63:335–346, 2015.

[30] J.-S. No, S. W. Golomb, G. Gong, H.-K. Lee, and P. Gaal. Binary pseudorandom sequences of period $2^n - 1$ with ideal autocorrelation. *IEEE Transactions on Information Theory*, 44(2):814–817, 1998.

[31] T. Rachwalik, J. Szmidt, R. Wicik, and J. Zabłocki. Generation of nonlinear feedback shift registers with special-purpose hardware. In *2012 Military Communications and Information Systems Conference (MCC)*, pages 1–4. IEEE, 2012.

[32] R. A. Rueppel. *Analysis and design of stream ciphers*. Springer Science & Business Media, 2012.

[33] SAGE. Specification of the 3GPP confidentiality and integrity algorithms 128-eea3 & 128-eia3. document 2: ZUC specification. version 1.6, etsi/sage, 2011., 2011. https://www.gsma.com/aboutus/wp-content/uploads/2014/12/eea3eia3zucv16.pdf.