Cryptographically Strong de Bruijn Sequences with Large Periods

Kalikinkar Mandal, and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, CANADA
{kmandal, ggong}@uwaterloo.ca

Abstract. In this paper we first refine *Mykkeltveit et al.*'s technique for producing de Bruijn sequences through compositions. We then conduct an analysis on an approximation of the feedback functions that generate de Bruijn sequences. The cycle structures of the approximated feedback functions and the linear complexity of a sequence produced by an approximated feedback function are determined. Furthermore, we present a compact algebraic representation of an $(n+16)$-stage nonlinear feedback shift register (NLFSR) and a few examples of de Bruijn sequences of period 2^n, $35 \leq n \leq 40$, which are generated by the recursively constructed NLFSR together with the evaluation of their implementation.

Keywords: de Bruijn sequences, nonlinear feedback shift registers, pseudorandom sequence generators, span n sequences, compositions.

1 Introduction

Recently, *nonlinear feedback shift registers* (NLFSRs) have received a lot of attention in designing cryptographic primitives such as pseudorandom sequences generators (PRSGs) and stream ciphers to provide security and privacy in communication systems. For example, well-known stream ciphers such as Grain and Trivium used NLFSRs as the basic building blocks in their designs [4]. Due to their efficient hardware implementations, NLFSRs have a number of applications in constrained environments for instance RFID tags and sensor networks.

The theory of NLFSRs is not well explored. Most of the known results are collectively reported in Golomb's book [9]. To design a secure cryptographic primitive, such as a key stream generator in a stream cipher, an arbitrary NLFSR cannot be used to generate keystreams with unpredictability, since the randomness properties of a sequence generated by an arbitrary NLFSR are not known and hard to determine. A classical approach to use an NLFSR in a keystream generator is to combine it with a linear feedback shift register (LFSR), where the LFSR guarantees the period of an output keystream. A (binary) *de Bruijn sequence* is a sequence of period 2^n in which each n-bit pattern occurs exactly once in one period of the sequence (this is referred to as the *span n property*). A de Bruijn sequence can be generated by an n-stage NLFSR and it has known randomness properties such as long period, balance, span n property [3, 8, 9].

The linear span or linear complexity of a sequence is defined as the length of the shortest LFSR which generates the sequence. De Bruijn sequences have high linear complexity [2], i.e., the linear complexity is greater than half of its period. However, one can delete one zero bit from the run of zeros of length n of a de Bruijn sequence of period 2^n. The resulting sequence is called a *modified de Bruijn* or *span n sequence*. A span n sequence keeps the balance property and span n properly of the corresponding de Bruijn sequence except for linear span, which could be very low. A classic example of this phenomenon is m-sequences, which are a class of span n sequences that can be generated by an LFSR. By this technique, one can generate a de Bruijn sequence from an m-sequence. The linear complexity of this type of de Bruijn sequences is at least $2^{n-1}+n+1$ [2]. Likewise, from this de Bruijn sequence, one can remove a zero from the run of zeros of length n then it becomes an m-sequence with linear complexity n. Thus, the lower bound of the linear complexity of this de Bruijn sequence drops to n only after removing one zero from the run of zeros of length n [12]. This shows that the linear complexity of a de Bruijn sequence is not an adequate measurement for its randomness. Instead, it should be measured in terms of the linear complexity of its corresponding span n sequence, since they have only one bit difference.

A de Bruijn sequence and a span n sequence are in one-to-one correspondence, i.e., a span n sequence can be produced from a de Bruijn sequence by removing one zero from the run of zeros of length n. A number of publications in the literature have been discussed several techniques for generating de Bruijn sequences [1, 5–7, 16, 18, 21]. In most of the techniques, a de Bruijn sequence is produced by joining many small cycles, which enforces that either the procedure needs some extra memory for storing the state information for joining the cycles or the feedback function must contain many product terms in order to join the cycles. Most of the existing methods are not efficient for producing de Bruijn sequences of period $2^n, n \geq 30$.

The objective of this paper is to investigate how to generate a de Bruijn sequence where the corresponding span n sequence has a large linear complexity through an iterative method or a composition method. The contribution of this paper is that first we refine *Mykkeltveit et al.*'s iterative method [21] for generating a large period de Bruijn sequence recursively from a feedback function of a short stage feedback shift register which generates a span n sequence. Then we give an analysis of the recursively constructed nonlinear recurrence relation from a cryptographic point of view. In the analysis, we investigate an approximation of the feedback function by setting some product terms as constant functions, and determine the cycle structure of an approximated feedback function and the linear complexity of a sequence generated by an approximated feedback function. The analysis also shows that the de Bruijn sequences generated by the composition have strong cryptographic properties if the starting short span n sequence is strong. Thirdly, we derive an algebraic normal form representation of an $(n+16)$-stage NLFSR and present a few instances of cryptographically strong de Bruijn sequences with periods in the range of 2^{35} and 2^{40} together with the discussions of their implementation issues.

The remainder of the paper is organized as follows. In Section 2, we define some notations and recall some background results that are used in this paper. In Section 3, we present the recursive construction of an arbitrary stage nonlinear feedback shift register that can generate a de Bruijn sequence. In Section 4, we analyze the feedback function of the nonlinear recurrence relation from the cryptographic point of view. In Section 5, we present a few instances of cryptographically strong de Bruijn sequences with periods in the range of 2^{35} and 2^{40}. In section 6, we describe some methods for optimizing the number of additions while computing the feedback function of a recursively constructed NLFSR with 40 stages. Finally, in Section 7, we conclude the paper.

2 Preliminaries

In this section, we define and explain some notations, terms and mathematical functions that will be used in this paper.

- $\mathrm{F}_2 = \{0, 1\}$: the Galois field with two elements.
- F_{2^t} : a finite field with 2^t elements that is defined by a primitive element α with $p(\alpha) = 0$, where $p(x) = c_0 + c_1x + \cdots + c_{t-1}x^{t-1} + x^t$ is a primitive polynomial of degree t (≥ 2) over F_2.
- Z_o^n and Z_e^n denote two sets of odd integers and even integers between 1 and n, respectively.
- $Supp(f)$: the set of all inputs for which $f(x) = 1, x \in \mathrm{F}_{2^n}$, where f is a Boolean function in n variables.
- $H(f)$: the Hamming weight of the Boolean function f.

2.1 Basic Definitions and Properties

Let $\mathbf{a} = \{a_i\}$ be a periodic binary sequence generated by an n-stage linear or nonlinear feedback shift register, which is defined as [9]

$$a_{n+k} = f(a_k, a_{k+1}, ..., a_{k+n-1}) = a_k + g(a_{k+1}, ..., a_{k+n-1}),\ a_i \in \mathrm{F}_2,\ k \geq 0 \quad (1)$$

where $(a_0, a_1, ..., a_{n-1})$ is called the *initial state* of the feedback shift register, $f(\cdot)$ is a Boolean function in n variables and $g(\cdot)$ is a Boolean function in $(n-1)$ variables. The recurrence relation (1) is called a *nonsingular* recurrence relation. If the function f is an affine function, then the sequence $\mathbf{a}$ is called a *LFSR sequence*; otherwise it is called a *NLFSR sequence*. The *minimal polynomial* of the sequence $\mathbf{a}$ is defined by the LFSR of shortest length that can generate the sequence and the degree of the minimal polynomial determines the linear complexity of the sequence $\mathbf{a}$.

It is well known that a nonsingular feedback shift register with a feedback function f partitions the space of 2^n n-tuples into a finite number of cycles, which is known as the *cycle decomposition* or *cycle structure* of f and we denote by $\Omega(f)$ the cycle decomposition of f. Each cycle in $\Omega(f)$ can be considered as a periodic sequence. In particular, the cycle decomposition of a feedback shift register that generates a span n sequence contains only two sequences, one is the span n sequence and the other one is the zero sequence.

Property 1. The linear span of a de Bruijn sequence, denoted as LS_{db}, is bounded by [2]

$$2^{n-1} + n + 1 \leq LS_{db} \leq 2^n - 1.$$

On the other hand, the linear span of a span n sequence, denoted as LS_s, is bounded by [20]

$$2n < LS_s \leq 2^n - 2.$$

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

Proposition 1. *[9] Let f be a feedback function in n variables that generates a span n sequence, then the function $h = f + \prod_{i=1}^{n-1}(x_i + 1)$ generates a de Bruijn sequence.*

The Welch-Gong (WG) Transformation:

Let $\mathrm{Tr}(x) = x + x^2 + \cdots + x^{2^{t-1}}, x \in \mathrm{F}_{2^t}$ be the trace function mapping from F_{2^t} to F_2. Let t be a positive integer with $t \bmod 3 \not\equiv 0$ and $3k \equiv 1 \bmod t$ for some integer k. We define a function h from F_{2^t} to F_{2^t} by $h(x) = x + x^{q_1} + x^{q_2} + x^{q_3} + x^{q_4}$ and the exponents are given by

$$q_1 = 2^k + 1, q_2 = 2^{2k} + 2^k + 1, q_3 = 2^{2k} - 2^k + 1, q_4 = 2^{2k} + 2^k - 1.$$

Then the function, from F_{2^t} to F_{2^t}, defined by

$$\mathrm{WGP}(x) = h(x + 1) + 1$$

is known as the *WG permutation* and the functions, from F_{2^t} to F_2, defined by

$$f_d(x) = \mathrm{Tr}(\mathrm{WGP}(x^d)) \text{ and } g_d(x) = \mathrm{Tr}(h(x^d)),\ d \in D_t$$

are known as the *WG transformation* and *five-term (or 5-term) function*, respectively [10, 11], where D_t is the set of coset leaders which are co-prime with $2^t - 1$. The WG transformation has good cryptographic properties such as high algebraic degree, high nonlinearity. Moreover, a WG sequence has high linear span [11]. For a fixed t, the number of WG transformations including decimations is given by $\left(\frac{\phi(2^t-1)}{t}\right)^2$ [10].

2.2 Composite Recurrence Relations

Let $g(x_0, x_1, ..., x_{n-1}, x_n) = x_0 + G(x_1, x_2, ..., x_{n-1}) + x_n = 0$ and $f(x_0, x_1, ..., x_{m-1}, x_m) = x_0 + F(x_1, x_2, ..., x_{m-1}) + x_m = 0$ be two recurrence relations of n and m stages, respectively that generate periodic sequences, where G and F are Boolean functions in $(n-1)$ and $(m-1)$ variables, respectively. Then, a *composite recurrence relation*, denoted as $g \circ f$, is defined by [21]

$$g \circ f = g(f(x_0, ..., x_m), f(x_1, ..., x_{m+1}), ..., f(x_n, ..., x_{m+n-1})) = 0,$$

which is a recurrence relation of $(n + m)$ stages. The operation "$\circ$" is regarded as the composition operation of recurrence relations. Note that $g \circ f$ and $f \circ g$ are not the same in general. For any feedback function f, the cycle decomposition of g is a subset of the cycle decomposition of $g \circ f$. For more detailed treatments on the cycle decomposition of a composite recurrence relation, see [21].

Let $\psi(x_0, x_1) = x_0 + x_1$ be a Boolean function. Throughout this paper, the definition of ψ is fixed. We now restate the following results from [21] which will be used to construct an arbitrary stage NLFSR that will generate a de Bruijn sequence.

Lemma 1. *[21] Let p be a characteristic polynomial, and $q(x_0, ..., x_n) = x_0 + x_n + w(x_1, ..., x_{n-1})$ where w is a Boolean function in $(n-1)$ variables and let $a \in \Omega(q)$ and $x \in \Omega(q \circ p)$. If the minimal polynomial of a is coprime with p, then $x = b + c$ where b's minimal polynomial is the same as the minimal polynomial of a and c's minimal polynomial is p.*

Theorem 1. *[21] Let $g = x_0 + x_n + f(x_1, ..., x_{n-1})$, which generates a de Bruijn sequence with period 2^n and let $\psi(x_0, x_1) = x_0 + x_1$. Then both $h_1 = g \circ \psi + \prod_{i \in Z_o^n} x_i \prod_{i \in Z_e^n}(x_i + 1)$ and $h_2 = g \circ \psi + \prod_{i \in Z_o^n}(x_i + 1) \prod_{i \in Z_e^n} x_i$ generate de Bruijn sequences with period 2^{n+1}.*

3 Recursive Feedback Functions in Composed de Bruijn Sequences

In [21], *Mykkeltveit et al.* mentioned the idea of constructing a long stage NLFSR from a short stage NLFSR by repeatedly applying Theorem 1 when g is a linear function in two variables that generates a de Bruijn sequence. In this section, we first refine *Mykkeltveit et al.*'s method and then we show an analytic formulation of a recursive feedback function of an $(n+k)$-stage NLFSR, which is constructed from a feedback function of an n-stage NLFSR by repeatedly applying Theorem 1 and the composition operation.

3.1 The k-th Order Composition of a Boolean Function

Let $g(x_0, x_1, ..., x_n) = x_0 + x_n + G(x_1, x_2, ..., x_{n-1})$ be a Boolean function in $(n+1)$ variables where G is a Boolean function in $(n-1)$ variables. The *first order* composition of ψ and g, denoted as $g \circ \psi$, is given by [21]

$$\begin{aligned} g \circ \psi &= g(x_0 + x_1, x_1 + x_2, ..., x_n + x_{n+1}) \\ &= x_0 + x_1 + x_{n+1} + x_n + G(x_1 + x_2, ..., x_{n-1} + x_n). \end{aligned}$$

Similarly, the *k-th order* composition of g with respect to ψ is defined by

$$g \circ \psi^k = \left(g \circ \psi^{k-1}\right) \circ \psi,$$

where $g \circ \psi^{k-1}$ is $(k-1)$-th order composition of g with respect to ψ.

3.2 Repeated Compositions of a Product term

Let X_0^p be a product term in p variables which is given by

$$X_0^p = \prod_{i \in Z_o^p} x_i \prod_{i \in Z_e^p} (x_i + 1).$$

Then the first order composition of X_0^p with respect to ψ, denoted as X_1^p, is given by

$$X_1^p = \prod_{i \in Z_o^p} (x_i + x_{i+1}) \prod_{i \in Z_e^p} (x_i + x_{i+1} + 1)$$

which is a product of sum terms in $(p+1)$ variables. Similarly, the k-th order composition of X_0^p with respect to ψ, denoted by X_k^p, is defined as $X_k^p = (X_{k-1}^p) \circ \psi$, which is a product of sum terms in $(p+k)$ variables. Note that the composition operation with respect to ψ increases the number of variables in X_0^p by one when it repeats once, but the composition operation does not increase the algebraic degree of X_0^p.

We denote by $J^{n-1} = \prod_{i=1}^{n-1}(x_i+1)$. In a similar manner, the k-th order composition of J^{n-1} with respect to ψ, denoted as J_k^{n-1}, is defined by $J_k^{n-1} = \left(J_{k-1}^{n-1}\right) \circ \psi$, where J_{k-1}^{n-1} is the $(k-1)$-th order composition of J^{n-1}.

Let us now define a function I_k^n in $(n+k-1)$ variables as follows

$$I_k^n(x_1, x_2, ..., x_{n+k-1}) = J_k^{n-1} + X_{k-1}^n + X_{k-2}^{n+1} + \cdots + X_1^{n+k-2} + X_0^{n+k-1}.$$

Then, I_k^n satisfies the following recursive relation

$$I_{k+1}^n = I_k^n \circ \psi + X_0^{n+k}, \text{ for } k \geq 0 \text{ and } n \geq 2,$$

where $I_0^n = J^{n-1}$.

3.3 The Recursive Construction of the NLFSR

In this subsection, we give the construction of an $(n+k)$-stage NLFSR that is constructed from an n-stage NLFSR.

Proposition 2. *Let $g(x_0, x_1, ..., x_n) = x_n + x_0 + G(x_1, x_2, ..., x_{n-1})$, which generates a span n sequence of period $2^n - 1$, where G is a Boolean function in $(n-1)$ variables. Then, for any integer $k \geq 0$, $R_k^n(x_0, x_1, ..., x_{n+k}) = (x_n + x_0) \circ \psi^k + G(x_1, x_2, ..., x_{n-1}) \circ \psi^k + I_k^n(x_1, ..., x_{n+k-1})$ generates a de Bruijn sequence of period 2^{n+k}.*

Proof. By applying Theorem 1 to the feedback function $(g + J^{n-1})$ k times, it becomes

$$\begin{aligned} R_k^n(x_0, x_1, ..., x_{n+k}) &= (x_n + x_0) \circ \psi^k + G(x_1, x_2, ..., x_{n-1}) \circ \psi^k + \\ &\quad I_k^n(x_1, ..., x_{n+k-1}), k \geq 0 \qquad (2) \\ &= (x_n + x_0) \circ \psi^k + G(x_1 \circ \psi^k, ..., x_{n-1} \circ \psi^k) + \\ &\quad I_k^n(x_1, x_2, ..., x_{n+k-1}). \qquad (3) \end{aligned}$$

The function R_k^n is a feedback function in $(n+k)$ variables of an NLFSR and the recurrence relation, $R_k^n = 0$, generates a de Bruijn sequence with period 2^{n+k}. □

One can construct the feedback function R_{k+1}^n from R_k^n in the following recursive manner

$$R_{k+1}^n = R_k^n \circ \psi + X_0^{n+k} \text{ or } R_{k+1}^n = g \circ \psi^{k+1} + I_{k+1}^n, k \geq 0$$

where $R_0^n = (g + J^{n-1})$.

Remark 1. For $k = 1$, Proposition 2 is the same as Theorem 1 which is also found by Lempel in [18]. For $k = 1$ and g is a primitive polynomial, Proposition 2 is similar to Theorem 2 in [21].

Remark 2. According to Theorem 1, the product term X_0^p in the recurrence relation (2) can be replaced by the product term $\prod_{i \in Z_o^p}(x_i + 1)\prod_{i \in Z_e^p} x_i$.

We now present an algebraic normal form representation of I_{16}^n for a recurrence relation of $(n+16)$ stages, which is derived by putting $k = 16$ in the recurrence relation (2). Then, the nonlinear recurrence relation of $(n+16)$ stages is given by

$$R_{16}^n(x_0, ..., x_{n+16}) = x_{n+16} + x_n + x_0 + x_{16} + G(x_1 + x_{17}, ..., x_{n-1} + x_{n+15}) \\ + J_{16}^{n-1} + X_{15}^n + \cdots + X_1^{n+14} + X_0^{n+15} = 0 \qquad (4)$$

where $J_{16}^{n-1} = \prod_{i=1}^{n-1}(x_i + x_{i+16} + 1)$ and $X_j^i = T_{o,j}^i \cdot T_{e,j}^i,\ i + j = (n + 15),\ n \leq i \leq n + 15$, $T_{o,j}^i$ and $T_{e,j}^i$ are given in Table 1. In the product terms, the subscripts o and e represent the odd indices product terms and even indices product terms. Note that each product term $X_j^i,\ i+j = (n + 15),\ n \leq i \leq n + 15$, is a function of $(n + 15)$ variables.

Table 1. Product terms in I_{16}^n of the recurrence relation (4)

$T_{o,15}^n = \prod_{i \in Z_o^n}\left(\sum_{l=0}^{15} x_{i+l}\right)$	$T_{o,14}^{n+1} = \prod_{i \in Z_o^{n+1}}\left(\sum_{l=0}^{7} x_{i+2l}\right)$
$T_{o,13}^{n+2} = \prod_{i \in Z_o^{n+2}}(x_i + x_{i+1} + \sum_{l=1}^{3}(x_{i+2^l} + x_{i+2^l+1}))$	$T_{o,12}^{n+3} = \prod_{i \in Z_o^{n+3}}(\sum_{l=0}^{3} x_{i+4l})$
$T_{o,11}^{n+4} = \prod_{i \in Z_o^{n+4}}(\sum_{l=0}^{4} x_{i+l} + \sum_{l=8}^{11} x_{i+l})$	$T_{o,10}^{n+5} = \prod_{i \in Z_o^{n+5}}(x_i + x_{i+2} + x_{i+8} + x_{i+10})$
$T_{o,9}^{n+6} = \prod_{i \in Z_o^{n+6}}(x_i + x_{i+1} + x_{i+8} + x_{i+9})$	$T_{o,8}^{n+7} = \prod_{i \in Z_o^{n+7}}(x_i + x_{i+8})$
$T_{o,7}^{n+8} = \prod_{i \in Z_o^{n+8}}(\sum_{l=0}^{7} x_{i+l})$	$T_{o,6}^{n+9} = \prod_{i \in Z_o^{n+9}}(\sum_{l=0}^{3} x_{i+2l})$
$T_{o,5}^{n+10} = \prod_{i \in Z_o^{n+10}}(x_i + x_{i+1} + x_{i+4} + x_{i+5})$	$T_{o,4}^{n+11} = \prod_{i \in Z_o^{n+11}}(x_i + x_{i+4})$
$T_{o,3}^{n+12} = \prod_{i \in Z_o^{n+12}}(\sum_{l=0}^{3} x_{i+l})$	$T_{o,2}^{n+13} = \prod_{i \in Z_o^{n+13}}(x_i + x_{i+2})$
$T_{o,1}^{n+14} = \prod_{i \in Z_o^{n+14}}(x_i + x_{i+1})$	$T_{o,0}^{n+15} = \prod_{i \in Z_o^{n+16}} x_i$
$T_{e,15}^n = \prod_{i \in Z_e^n}(\sum_{l=0}^{15} x_{i+l} + 1)$	$T_{e,14}^{n+1} = \prod_{i \in Z_e^{n+1}}(\sum_{l=0}^{7} x_{i+2l} + 1)$
$T_{e,13}^{n+2} = \prod_{i \in Z_e^{n+2}}(x_i + x_{i+1} + \sum_{l=1}^{3}(x_{i+2^l} + x_{i+2^l+1}) + 1)$	$T_{e,12}^{n+3} = \prod_{i \in Z_e^{n+3}}(\sum_{l=0}^{3} x_{i+4l} + 1)$
$T_{e,11}^{n+4} = \prod_{i \in Z_e^{n+4}}(\sum_{l=0}^{4} x_{i+l} + \sum_{l=8}^{11} x_{i+l} + 1)$	$T_{e,10}^{n+5} = \prod_{i \in Z_e^{n+5}}(x_i + x_{i+2} + x_{i+8} + x_{i+10} + 1)$
$T_{e,9}^{n+6} = \prod_{i \in Z_e^{n+6}}(x_i + x_{i+1} + x_{i+8} + x_{i+9} + 1)$	$T_{e,8}^{n+7} = \prod_{i \in Z_e^{n+7}}(x_i + x_{i+8} + 1)$
$T_{e,7}^{n+8} = \prod_{i \in Z_e^{n+8}}(\sum_{l=0}^{7} x_{i+l} + 1)$	$T_{e,6}^{n+9} = \prod_{i \in Z_e^{n+9}}(\sum_{l=0}^{3} x_{i+2l} + 1)$
$T_{e,5}^{n+10} = \prod_{i \in Z_e^{n+10}}(x_i + x_{i+1} + x_{i+4} + x_{i+5} + 1)$	$T_{e,4}^{n+11} = \prod_{i \in Z_e^{n+11}}(x_i + x_{i+4} + 1)$
$T_{e,3}^{n+12} = \prod_{i \in Z_e^{n+12}}(\sum_{l=0}^{3} x_{i+l} + 1)$	$T_{e,2}^{n+13} = \prod_{i \in Z_e^{n+13}}(x_i + x_{i+2} + 1)$
$T_{e,1}^{n+14} = \prod_{i \in Z_e^{n+14}}(x_i + x_{i+1} + 1)$	$T_{e,0}^{n+15} = \prod_{i \in Z_e^{n+16}}(x_i + 1)$

4 Cryptanalysis of the Recursively Constructed NLFSR for Generating de Bruijn Sequences

Since the feedback function contains I_k^n and it includes many product terms whose algebraic degrees are high and the Hamming weights of these product terms are low, as a result, the function I_k^n can be approximated by a linear function or a constant function with high probability. In this section, we first investigate the success probability of approximating the function I_k^n by the zero function. We then study the cycle decomposition of an approximated recurrence relation after a successful approximation of the feedback function with high probability.

4.1 Hamming Weights of the Product Terms

Before calculating the success probability of approximating the function I_k^n by the zero function, we first need to derive the Hamming weight of a composed product term as I_k^n is a sum of $(k+1)$ composed product terms.

Proposition 3. *For an integer $r \geq 1$, the Hamming weight of X_r^p is equal to 2^r.*

Proof. For any product term X_0^p, the r-order composition is of the form

$$X_r^p = \prod_{i \in Z_o^p} U_i \cdot \prod_{i \in Z_e^p} V_i$$

where U_i is a sum of at most $(r+1)$ variables and V_i is also a sum of at most $(r+1)$ variables and the exact number of variables in U_i/V_i depends on the value of r. For simplicity, we assume that $r = 2^l, l \geq 0$. To find the Hamming weight of X_r^p, there are two cases arise.

Case I: When $1 \leq p \leq r+1$

If $r = 2^l$, then U_i and V_j can be written as $U_i = x_i + x_{i+r},\ i \in Z_o^p,\ V_j = (x_j + x_{j+r} + 1),\ j \in Z_e^p$, respectively. $X_r^p = 1$ if and only if $U_i = 1$ and $V_j = 1$ for all $i \in Z_o^p$ and $j \in Z_e^p$. This implies

$$\begin{aligned}
&x_1 = 1 + x_{1+r} = 1 + x_{1+2r} = \cdots = 1 + x_{l_1} = 0/1 \\
&x_2 = x_{2+r} = x_{2+2r} = \cdots = x_{l_2} = 0/1 \\
&\quad \vdots \\
&x_p = 1 + x_{p+r} = 1 + x_{p+2r} = \cdots = 1 + x_{l_n} = 0/1, \text{ if } p \text{ is odd} \\
&x_p = x_{p+r} = x_{p+2r} = \cdots = x_{l_p} = 0/1, \text{ if } p \text{ is even}
\end{aligned}$$

where $l_i \leq p + r,\ i = 1, 2, ..., p$. Note that X_r^p is a function in $(p+r)$ variables. For an $(p+r)$-tuple with $X_r^p = 1$, the values at $2p$ positions are determined by the values at p positions, which follows from the above set of equations and the remaining $(p + r - 2p)$ positions can take any binary value. Hence, the total number of $(p+r)$-tuples for which $X_r^p = 1$

is given by $2^p \cdot 2^{r-p} = 2^r$.

Case II: When $p \geq r+1$

Similarly, $X_r^p = 1$ if and only if $U_i = 1$ and $V_j = 1$ for all $i \in Z_o^p$ and $j \in Z_e^p$. This implies

$$\begin{aligned} x_1 &= 1 + x_{1+r} = 1 + x_{1+2r} = \cdots = 1 + x_{l_1} = 0/1 \\ x_2 &= x_{2+r} = x_{2+2r} = \cdots = x_{l_2} = 0/1 \\ &\vdots \\ x_{r-1} &= 1 + x_{2r-1} = \cdots = 1 + x_{l_{r-1}} = 0/1 \\ x_r &= x_{2r} = \cdots = x_{l_r} = 0/1 \end{aligned}$$

where $l_i \leq p+r,\ i = 1, 2, ..., r$. According to the above system of equations, the binary values at $(p+r)$ positions are determined by the binary values at r positions and these r positions can take any values. Hence, the total number of $(p+r)$-tuples for which $X_r^p = 1$ is given by 2^r.

By considering $U_i = 1$ and $V_j = 1$ for all $i \in Z_o^p$ and $j \in Z_e^p$ as a system of linear equations with p equations and $(p+r)$ unknown variables over F_2, it follows that the Hamming weight of X_r^p is equal to the number of solutions of the system of linear equations, which is equal to $2^{p+r-r} = 2^r$ for any positive integer $r (\neq 2^l)$. □

Proposition 4. *For any integer $r \geq 1$, the Hamming weight of J_r^{n-1} is equal to 2^r.*

Proof. The proof is similar to the proof of Proposition 3. □

Proposition 5. *For any integer $k \geq 1$ and $n \geq 2$, the Hamming weight of function I_k^n is equal to $2^k + 1$. One can approximate function I_k^n by the zero function with probability $(1 - \frac{1}{2^{n-1}} - \frac{1}{2^{n+k-1}})$.*

Proof. By Proposition 3, the Hamming weight of $X_j^{n+k-1-j}$, i.e, $H(X_j^{n+k-1-j})$ is equal to 2^j, for $0 \leq j \leq k-1$. Note that $X_j^{n+k-1-j} = 1$ is a system of linear equations with $(n+k-1-j)$ equations and $(n+k-1)$ unknown variables and $Supp(X_j^{n+k-1-j})$ contains the set of all solutions. It is not hard to show that the support of $X_i^{n+k-1-i}$ and $X_j^{n+k-1-j}$ are disjoint for $0 \leq i \neq j \leq n-1$. Again, $(\cup_{j=0}^{k-2} Supp(X_j^{n+k-1-j})) \subset Supp(J_k^{n-1})$, and $Supp(X_{k-1}^{n+k-1})$ and $Supp(J_k^{n-1})$ are disjoint. Then the cardinality of the support of I_k^n is equal to $(2^k + 2^{k-1} - \sum_{j=0}^{k-2} 2^j) = (2^k + 2^{k-1} - 2^{k-1} + 1) = 2^k + 1$. Hence, the Hamming weight of I_k^n is $2^k + 1$.

Since the Hamming weight of I_k^n is $2^k + 1$, the number of inputs for which I_k^n takes the value zero is equal to $2^{n+k-1} - 2^k - 1$. Hence, one can approximate the function I_k^n by the zero function with probability $(1 - \frac{1}{2^{n-1}} - \frac{1}{2^{n+k-1}})$. □

4.2 Cycle Structures of the Recurrence Relation after Approximation

By Proposition 5, the function I_k^n can be approximated by the *zero function* with probability about $(1 - \frac{1}{2^{n-1}})$. As a consequence, Eq. (2) can be written as follows

$$R_{k,a}^n(x_0, x_1, ..., x_{n+k}) = ((x_n + x_0) + G(x_1, x_2, ..., x_{n-1})) \circ \psi^k. \quad (5)$$

In the following proposition, we provide the cycle structure of the above recurrence relation.

Lemma 2. *For an integer $k \geq 1$, $\Omega(R_{k,a}^n) = \Omega(g) \oplus \Omega(\psi^k)$, i.e., any sequence $x \in \Omega(R_{k,a}^n)$ can be written as $x = b + c$, where b's minimal polynomial is the same as the minimal polynomial of a span n sequence that is generated by g and c's minimal polynomial is $(1 + x)^k$ and $\oplus$ denotes the direct sum operation.*

Proof. Let s be a span n sequence generated by g and let $h(x)$ the minimal polynomial of s. Then, $h(x) = h_1(x) \cdot h_2(x) \cdots h_r(x)$, where h_i's are distinct irreducible polynomials of degree less than or equal to n and the value of r depends on the sequence, see [10, 12, 20]. If $h_i(x) = (1 + x)$ for some i, then the sequence s is not a span n sequence. On the other hand, the minimal polynomial of ψ^k is $(1+x)^k$. Again, the minimal polynomial of a sequence generated by ψ^k is a factor of $(1 + x)^k$. As $h(x)$ does not contain the factor $(1 + x)$, the minimal polynomial of s and the minimal polynomial of ψ^k are relatively prime with each other. Then, by Lemma 1, any sequence $x \in \Omega(R_{k,a}^n)$ can be represented by $x = b+c$ where $b \in \Omega(g)$ and $c \in \Omega(\psi^k)$. Hence, the cycle decomposition of $R_{k,a}^n$ is a direct sum of $\Omega(g)$ and $\Omega(\psi^k)$, i.e., $\Omega(R_{k,a}^n) = \Omega(g) \oplus \Omega(\psi^k)$. □

Proposition 6. *The cycle decomposition of $R_{k,a}^n$, i.e., $\Omega(R_{k,a}^n)$ contains $2 \cdot (\Gamma_2(k) + 1)$ cycles with $(\Gamma_2(k) + 1)$ cycles of period at least $2^n - 1$ and $(\Gamma_2(k) + 1)$ cycles of period at most $2^{\lceil \log_2 k \rceil}$, where $\Gamma_2(k)$ is the number of all coset leaders modulo $2^k - 1$.*

Proof. For any positive integer $k \geq 1$, the cycle decomposition of ψ^k is the cycle decomposition of $(1 + x)^k$, which contains sequences with period $2^{\lceil \log_2 i \rceil}$, $1 \leq i \leq k$, and the number of cycles is given by $(\Gamma_2(k) + 1)$ including the zero cycle (see [9], Th. 3.4, page-42). Again, the cycle decomposition of g contains only two cycles, one is a cycle of length $2^n - 1$ and the other one is the zero cycle of length one. Therefore, by Lemma 2,

$\Omega(R_{k,a}^n)$ contains $2 \cdot (\Gamma_2(k) + 1)$ cycles where $(\Gamma_2(k) + 1)$ cycles are of length at least $2^n - 1$ and $(\Gamma_2(k)+1)$ cycles are of length at most $2^{\lceil \log_2 k \rceil}$. □

Remark 3. If the function R_k^n is approximated by the function $(R_{k,a}^n + J_k^{n-1})$ with high probability, then $|\Omega(R_{k,a}^n + J_k^{n-1})| = \Gamma_2(k) + 1$ and the period of a sequence in $\Omega(R_{k,a}^n + J_k^{n-1})$ is bounded below by 2^n.

Proposition 7. *Let $\Omega(R_{k,a}^n)$ be the cycle decomposition of $R_{k,a}^n$. For any sequence $x \in \Omega(R_{k,a}^n)$ with period at least $2^n - 1$, the linear complexity of x is bounded below by the linear complexity of the sequence generated by g.*

Proof. We already showed in Lemma 2 that any sequence $x \in \Omega(R_{k,a}^n)$ can be written as $x = b + c$ where $b \in \Omega(g)$, $c \in \Omega(\psi^k)$, and the minimal polynomial of b is coprime with the minimal polynomial of c. Since the minimal polynomial of b is coprime with the minimal polynomial of c, the linear complexity of x is equal to the sum of the linear complexities of b and c. Therefore, the linear complexity of x is greater or equal to the linear complexity of b. Hence, the assertion is established. □

Remark 4. Using the recurrence relation (2) with G as a linear function, one can generate a de Bruijn sequence with period 2^{n+k} and linear complexity at least $(2^{n+k-1} + n + k + 1)$ for an arbitrary positive integer k. Nevertheless, this de Bruijn sequence is not suitable for cryptographic applications such as to use this sequence as a building block for designing a PRSG or a stream cipher, because in the entire sequence most of the bits are linearly related to the internal state bits and only at $H(I_k^n)$ positions the bits are nonlinearly related to the internal state bits due to the nonlinear term I_k^n, which is vulnerable against a cryptanalytic attack. Whereas, if the function g is nonlinear, then the output sequence bits of the de Bruijn sequence will be nonlinearly related to the internal state bits of the NLFSR and which may create a complex cryptanalytic attack.

Remark 5. Propositions 5, 6, and 7 suggest that in order to generate a strong de Bruijn sequence by this technique, the starting span n sequence generated by g should have good randomness properties, particularly, long period and an optimal or suboptimal linear complexity. If an attacker is successful in approximating the feedback function R_k^n by the feedback function $g \circ \psi^k$, then the security of the sequence generated by R_k^n depends on the security of the sequence generated by g.

5 Designing Parameters for Strong Cryptographic de Bruijn Sequences

In this section, we present a few examples of cryptographically strong de Bruijn sequences with period 2^{n+k} that are generated by an $(n+k)$-stage NLFSR for $19 \leq n \leq 24$ and $k = 16$. In order to generate de Bruijn sequences with period 2^{40}, we choose $n = 24$ and $k = 16$.

5.1 Tradeoff Between *n* and *k*

It can be observed from the construction of the recurrence relation that one can construct an $(n+k)$-stage recurrence relation by choosing a small value of n and a large value of k since for a small value of n it is easy to find a span n sequence and the success probability of approximating the feedback function is low (see Proposition 5). However, for such a choice of the parameters, the recurrence relation contains many product terms, as a result, the function I_k^n may not be calculated efficiently. Thus, for generating a strong de Bruijn efficiently, one needs to choose the parameters in such a way that the nonlinearly generated span n sequence is large enough and the number of product terms in I_k^n is as small as possible.

5.2 Examples of de Bruijn Sequences with Large Periods

Let $\{x_j\}_{j\geq 0}$ be a binary span n sequence generated by the following n-stage recurrence relation for a suitable choice of a decimation number d, a primitive polynomial $p(x)$, and a t-tap position [19]

$$x_n = x_0 + f_d(x_{r_1}, x_{r_2}, ..., x_{r_t}) \tag{6}$$

where $(r_1, r_2, ..., r_t)$ with $0 < r_1 < r_2 < \cdots < r_t < n$ is called a t-tap position and f_d is a WG transformation. Here a decimation number is a coset leader which is coprime with $2^t - 1$. Then the recurrence relation (4) with G as the WG transformation can be written as

$$\begin{aligned} R_{16}^n = {} & x_{n+16} + x_n + x_0 + x_{16} + f_d(x_{r_1} + x_{r_1+16}, ..., x_{r_t} + x_{r_t+16}) + J_{16}^{n-1} \\ & + X_{15}^n + X_{14}^{n+1} + \cdots + X_1^{n+14} + X^{n+15} = 0 \end{aligned} \tag{7}$$

where $J_{16}^{n-1} = \prod_{i=1}^{n-1}(x_i + x_{i+16} + 1)$ and $X_j^p = T_{o,j}^p \cdot T_{e,j}^p, p + j = (n + 15), n \leq p \leq n + 15$, $T_{o,j}^p$ and $T_{e,j}^p$ are given in Table 1. The recurrence relation (7) can generate a de Bruijn sequence for a suitable choice of a decimation number d, a primitive polynomial $p(x)$, and a t-tap position. Our de Bruijn sequences are uniquely represented by the following four parameters:

1. the decimation number d,

2. the primitive polynomial $p(x)$,
3. the t-tap position $(r_1, r_2, ..., r_t)$, and
4. I_k^n.

Table 2 presents a few examples of cryptographically strong de Bruijn sequences with periods in the range of 2^{35} and 2^{40}. In Table 2, the computations for the linear complexity of the 24-stage span n sequence has not finished yet. However, currently the lower bound of the linear complexity is at least 2^{22}. For more instances of span n sequences with an optimal or suboptimal linear span, see [19].

Table 2. De Bruijn sequences with periods $\geq 2^{35}$

WG over F_{2^t}	Decimation	Basis Polynomial	t-tap positions	span n	Linear Span,	I_k,	Period
t	d	$(c_0, c_1, ..., c_{t-1})$	$(r_1, r_2, ..., r_t)$	n	span n	k	2^{n+k}
13	55	$(1,1,1,0,1,1,0,0,1,1,0,1,0)$	$(1,2,3,4,5,6,9,10,11,12,13,15,17)$	24	$--$	16	2^{40}
8	53	$(1,1,1,0,0,1,1,1)$	$(1,2,5,6,8,11,12,15)$	21	$2^{21}-5$	16	2^{37}
8	29	$(1,1,1,0,0,0,0,1)$	$(1,2,6,8,9,15,16,19)$	21	$2^{21}-26$	16	2^{37}
8	31	$(1,1,1,0,0,0,0,1)$	$(1,2,10,12,13,16,18,19)$	20	$2^{20}-6$	16	2^{36}
8	1	$(1,1,0,0,0,1,1,0)$	$(1,3,4,5,8,11,12,15)$	19	$2^{19}-2$	16	2^{35}
7	5	$(1,0,0,1,1,1,0)$	$(1,2,6,8,10,12,16)$	20	$2^{20}-7$	16	2^{36}
7	19	$(1,0,1,0,0,1,1)$	$(1,2,3,5,6,10,18)$	19	$2^{19}-2$	16	2^{35}
5	1	$(1,1,1,0,1)$	$(5,10,12,18,19)$	20	$2^{20}-2$	16	2^{36}

Remark 6. Any feedback function g that generates a span n sequence can be used in recurrence relation (4) for producing a long period de Bruijn sequence. To the best of our knowledge, Table 2 contains a set of (longest) de Bruijn sequences whose algebraic normal form representations of the recurrence relations are known. We here used WG transformations for producing long period de Bruijn sequences because a span n sequence can be found by using WG transformations and the compact representation of the recurrence relation (6) in a systematic manner. In [23], eight span n sequences with periods in the range of $(2^{22}-1)$ and $(2^{31}-1)$ are presented and that have been used in a stream cipher.

6 Implementation

In this section, we provide some techniques for optimizing the number additions in the product terms for $k = 16$, and give an estimation for the number of multiplications and the time complexity for computing the function I_k^n in terms of n and k.

6.1 Optimizing the Number of Additions

For $k = 16$, I_k^n in recurrence relation (7) contains 17 product terms. For example, for $n = 24$ and $k = 16$, one needs 2116 additions for computing all product terms in I_k^n. In Table 1, we can observe that many partial-sum terms appear in different product terms. By reusing the result of a previously computed sum term, we can optimize the number of additions. For $k = 16$, three optimization rules are described in Table 3. According to the above three rules given in Table 3, the product terms in Table 1 can be written as that are given in Table 4.

Table 3. Optimization rules for addition

Optimization Rule I (OR-I)			
$Y_{1,i}^1 = x_i + x_{i+1}$	$Y_{1,i}^2 = x_{i+2} + x_{i+3}$	$Y_{3,i}^1 = x_{i+8} + x_{i+9}$	$Y_{3,i}^2 = x_{i+10} + x_{i+11}$
$Y_{2,i}^1 = x_{i+4} + x_{i+5}$	$Y_{2,i}^2 = x_{i+6} + x_{i+7}$	$Y_{4,i}^1 = x_{i+12} + x_{i+13}$	$Y_{4,i}^2 = x_{i+14} + x_{i+15}$
$Y_{1,i} = Y_{1,i}^1 + Y_{1,i}^2$	$Y_{2,i} = Y_{2,i}^1 + Y_{2,i}^2$	$Y_{0,2,i} = x_i + x_{i+2}$	$Y_{4,6,i} = x_{i+4} + x_{i+6}$
$Y_{3,i} = Y_{3,i}^1 + Y_{3,i}^2$	$Y_{4,i} = Y_{4,i}^1 + Y_{4,i}^2$	$Y_{8,10,i} = x_{i+8} + x_{i+10}$	$Y_{12,14,i} = x_{i+12} + x_{i+14}$
$Q_{0,i} = x_i$	$Q_{4,i} = x_i + x_{i+4}$	$Q_{3,i} = Y_{1,i}$	$Q_{7,i} = Q_{3,i} + Y_{2,i}$
$Q_{8,i} = x_i + x_{i+8}$	$Q_{12,i} = Q_{4,i} + x_{i+8} + x_{i+12}$	$Q_{11,i} = Q_{3,i} + Y_{3,i}$	$Q_{15,i} = Q_{7,i} + Y_{3,i} + Y_{4,i}$
$Q_{2,i} = Y_{0,2,i}$	$Q_{6,i} = Q_{2,i} + Y_{4,6,i}$	$Q_{1,i} = Y_{1,i}^1$	$Q_{5,i} = Q_{1,i} + Y_{2,i}^1$
$Q_{10,i} = Q_{2,i} + Y_{8,10,i}$	$Q_{14,i} = Q_{6,i} + Y_{8,10,i} + Y_{12,14,i}$	$Q_{9,i} = Q_{1,i} + Y_{3,i}^1$	$Q_{13,i} = Q_{5,i} + Y_{3,i}^1 + Y_{4,i}^1$
Optimization Rule II (OR-II)			
$Y_{1,i}^1 = x_i + x_{i+1}$	$Y_{1,i}^2 = x_{i+2} + x_{i+3}$	$Y_{1,i} = Y_{1,i}^1 + Y_{1,i}^2$	$Y_{2,i} = Y_{2,i}^1 + Y_{2,i}^2$
$Y_{2,i}^1 = x_{i+4} + x_{i+5}$	$Y_{2,i}^2 = x_{i+6} + x_{i+7}$	$Y_i = Y_{1,i} + Y_{2,i}$	$Y_{0,2,i} = x_i + x_{i+2}$
$Y_{4,6,i} = x_{i+4} + x_{i+6}$		$Y_{8,10,i} = x_{i+8} + x_{i+10}$	
$W_{0,i} = x_i$	$W_{1,i} = Y_{1,i}^1$	$W_{4,i} = x_i + x_{i+4}$	$W_{5,i} = Y_{1,i}^1 + Y_{2,i}^1$
$W_{2,i} = Y_{0,2,i}$	$W_{3,i} = Y_{1,i}$	$W_{6,i} = Y_{0,2,i} + Y_{4,6,i}$	$W_{7,i} = Y_{1,i} + Y_{2,i}$
$W_{8,i} = x_i + x_{i+8}$	$W_{9,i} = Y_{1,i}^1 + x_{i+8} + x_{i+9}$	$W_{10,i} = Y_{0,2,i} + Y_{8,10,i}$	
Optimization Rule III (OR-III)			
$Y_{1,i} = x_i + x_{i+1}$	$Y_{2,i} = x_{i+2} + x_{i+3}$	$Z_{0,1} = x_i$	$Z_{1,i} = Y_{1,i}$
$Z_{2,i} = x_i + x_{i+2}$	$Z_{3,i} = Y_{1,i} + Y_{2,i}$	$Z_{4,i} = x_i + x_{i+4}$	

Applying the rules given in Tables 3, the total number of additions required for computing I_{16}^n is given by $(n - 1 + 32 \cdot \lceil \frac{n+5}{2} \rceil + 32 \cdot \lfloor \frac{n+5}{2} \rfloor + 3 \cdot 18 + 3 \cdot 19 + 2 \cdot 5 + 2 \cdot 6 + 3 + 16) = (32 \cdot (n + 5) + n + 151)$, since the numbers of additions required for OR-I, OR-II and OR-III in Table 3 are 32, 18 and 5, respectively. For $n = 24$, the number of additions after applying the above three rules is equal to 1103.

6.2 The Total Number of Multiplications and the Time Complexity for Computing I_k^n

The maximum number of multiplications required for computing I_k^n is given by $\sum_{i=n-1}^{n+k-1}(i - 1) = (n(k + 1) + \frac{(k-1)(k-2)}{2} - 3)$ as one requires $(i-1)$ multiplications to compute a product of i numbers. In the following proposition, we estimate the time complexity for computing the function I_k^n.

Table 4. Product terms of the recurrence relation (7)

$T_{o,15}^{n} = \prod_{i \in Z_o^{n}} Q_{15,i}$	$T_{o,14}^{n+1} = \prod_{i \in Z_o^{n+1}} Q_{14,i}$
$T_{o,13}^{n+2} = \prod_{i \in Z_o^{n+2}} Q_{13,i}$	$T_{o,12}^{n+3} = \prod_{i \in Z_o^{n+3}} Q_{12,i}$
$T_{o,11}^{n+4} = \prod_{i \in Z_o^{n+4}} Q_{11,i}$	$T_{o,10}^{n+5} = \prod_{i \in Z_o^{n+5}} Q_{10,i}$
$T_{o,9}^{n+6} = \prod_{i \in Z_o^{n+5}} Q_{9,i} \cdot W_{9,n+6}$	$T_{o,8}^{n+7} = \prod_{i \in Z_o^{n+5}} Q_{11,i} \prod_{i=n+6,odd}^{n+7} W_{8,i}$
$T_{o,7}^{n+8} = \prod_{i \in Z_o^{n+5}} Q_{7,i} \cdot \prod_{i=n+6,odd}^{n+8} W_{7,i}$	$T_{o,6}^{n+9} = \prod_{i \in Z_o^{n+5}} Q_{6,i} \cdot \prod_{i=n+6,odd}^{n+9} W_{6,i}$
$T_{o,5}^{n+10} = \prod_{i \in Z_o^{n+5}} Q_{5,i} \cdot \prod_{i=n+6,odd}^{n+10} W_{5,i}$	$T_{o,4}^{n+11} = \prod_{i \in Z_o^{n+5}} Q_{4,i} \cdot \prod_{i=n+6,odd}^{n+11} W_{4,i}$
$T_{o,3}^{n+12} = \prod_{i \in Z_o^{n+5}} Q_{3,i} \cdot \prod_{i=n+6,odd}^{n+11} W_{3,i} \cdot Z_{3,n+12}$	$T_{o,2}^{n+13} = \prod_{i \in Z_o^{n+5}} Q_{2,i} \cdot \prod_{i=n+6,odd}^{n+11} W_{2,i} \cdot \prod_{i=n+12,odd}^{n+13} Z_{2,i}$
$T_{o,1}^{n+14} = \prod_{i \in Z_o^{n+5}} Q_{1,i} \cdot \prod_{i=n+6,odd}^{n+11} W_{1,i} \cdot \prod_{i=n+12,odd}^{n+13} Z_{1,i} \cdot (x_{n+14} + x_{n+15})$	$T_{o,0}^{n+15} = \prod_{i \in Z_o^{n+16}} x_i$
$T_{e,15}^{n} = \prod_{i \in Z_e^{n}} Q_{15,i}$	$T_{e,14}^{n+1} = \prod_{i \in Z_e^{n+1}} Q_{14,i}$
$T_{e,13}^{n+2} = \prod_{i \in Z_e^{n+2}} Q_{13,i}$	$T_{e,12}^{n+3} = \prod_{i \in Z_e^{n+3}} Q_{12,i}$
$T_{e,11}^{n+4} = \prod_{i \in Z_e^{n+4}} Q_{11,i}$	$T_{e,10}^{n+5} = \prod_{i \in Z_e^{n+5}} Q_{10,i}$
$T_{e,9}^{n+6} = \prod_{i \in Z_e^{n+5}} Q_{9,i} \cdot W_{9,n+6}$	$T_{e,8}^{n+7} = \prod_{i \in Z_e^{n+5}} Q_{11,i} \prod_{i=n+6,even}^{n+7} W_{8,i}$
$T_{e,7}^{n+8} = \prod_{i \in Z_e^{n+5}} Q_{7,i} \cdot \prod_{i=n+6,even}^{n+8} W_{7,i}$	$T_{e,6}^{n+9} = \prod_{i \in Z_e^{n+5}} Q_{6,i} \cdot \prod_{i=n+6,even}^{n+9} W_{6,i}$
$T_{e,5}^{n+10} = \prod_{i \in Z_e^{n+5}} Q_{5,i} \cdot \prod_{i=n+6,even}^{n+10} W_{5,i}$	$T_{e,4}^{n+11} = \prod_{i \in Z_e^{n+5}} Q_{4,i} \cdot \prod_{i=n+6,even}^{n+11} W_{4,i}$
$T_{e,3}^{n+12} = \prod_{i \in Z_e^{n+5}} Q_{3,i} \cdot \prod_{i=n+6,even}^{n+11} W_{3,i} \cdot Z_{3,n+12}$	$T_{e,2}^{n+13} = \prod_{i \in Z_e^{n+5}} Q_{2,i} \cdot \prod_{i=n+6,odd}^{n+11} W_{2,i} \cdot \prod_{i=n+12,even}^{n+13} Z_{2,i}$
$T_{e,1}^{n+14} = \prod_{i \in Z_e^{n+5}} Q_{1,i} \cdot \prod_{i=n+6,even}^{n+11} W_{1,i} \cdot \prod_{i=n+12,even}^{n+13} Z_{1,i} \cdot (x_{n+14} + x_{n+15})$	$T_{e,0}^{n+15} = \prod_{i \in Z_e^{n+16}} x_i$

Proposition 8. *The time complexity for computing the function I_k^n is approximately given by $\sum_{p=n-1}^{n+k-1} \lceil \log_2 p \rceil$.*

Proof. To compute a product term X_k^p, $n \leq p \leq n+k-1$, one requires at most $\lceil \log_2 p \rceil$-time. Since the function I_k^n contains $(k+1)$ product terms, the time complexity for computing I_k^n is given by $\sum_{p=n-1}^{n+k-1} \lceil \log_2 p \rceil$. □

7 Conclusions

In this paper, we first refined a technique by *Mykkeltveit et al.* for producing a long period de Bruijn sequence from a short period span n sequence through the composition operation. We then performed an analysis of the feedback function of the long period de Bruijn sequence from the cryptographic point of view. In our analysis, we studied an approximation of the feedback functions and the cycle structure of an approximated feedback function, and determined the linear complexity of a sequence generated by an approximated feedback function. In addition, we presented a compact algebraic normal form representation of an $(n+16)$-stage NLFSR and a few instances of de Bruijn sequences with periods in the range of 2^{35} and 2^{40} together with the discussions of their implementation issues. A long period de Bruijn sequence produced by this technique can be used as a building block to design secure lightweight cryptographic primitives such as pseudorandom sequence generators and stream ciphers with desired randomness properties.

References

1. A.H. Chan, and R.A. Games. On the Quadratic Spans of de Bruijn Sequences, *IEEE Transactions on Information Theory*, Vol. 36, No. 4, pp. 822 –829, July 1990.

2. A.H. Chan, R.A. Games, and E.L. Key. On the Complexities of de Bruijn Sequences, *Journal of Combinatorial Theory, Series A*, Vol. 33, No. 3, pp. 233 - 246, 1982.
3. N.G. de Bruijn. A Combinatorial Problem, *Proc. Koninklijke Nederlandse Akademie v. Wetenschappen*, Vol. 49, pp. 758 –764, 1946.
4. The eStream Project. `http://www.ecrypt.eu.org/stream/`.
5. T. Etzion. and A. Lempel. Construction of de Bruijn Sequences of Minimal Complexity, *IEEE Transactions on Information Theory,* Vol. 30, No. 5, pp. 705 – 709, September 1984.
6. H. Fredricksen. A Survey of Full Length Nonlinear Shift Register Cycle Algorithms, *SIAM Review*, 24(2):pp. 195–221, 1982.
7. H. Fredricksen. A Class of Nonlinear de Bruijn Cycles, *Journal of Combinatorial Theory,* Series A, Vol. 19, Issue 2, pp. 192 – 199 September 1975.
8. S.W. Golomb. On the Classification of Balanced Binary Sequences of Period $2^n - 1$, *IEEE Transformation on Information Theory,* Vol. 26, No. 6, pp. 730 – 732, November 1980.
9. S. W. Golomb. *Shift Register Sequences.* Aegean Park Press, Laguna Hills, CA, USA, 1981.
10. S. W. Golomb, and G. Gong. *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*, Cambridge University Press, New York, NY, USA, 2004.
11. G. Gong, and A. Youssef. Cryptographic Properties of the Welch-Gong Transformation Sequence Generators, *IEEE Transactions on Information Theory*, Vol. 48, No. 11, pp. 2837 – 2846, November 2002.
12. G. Gong. Randomness and Representation of Span n sequences, In *Proceedings of the 2007 International Conference on Sequences, Subsequences, and Consequences*, SSC'07, pp. 192 – 203, Springer-Verlag, 2007.
13. I.J. Good. Normal Recurring Decimals, *Journal of London Math. Soc.*, Vol. 21 (Part 3), 1946.
14. D. H. Green and K. R. Dimond. Nonlinear Product-Feedback Shift Registers, *Proceeding IEE 117*, pp. 681 – 686, 1970.
15. D. H. Green and K. R. Dimond. Some Polynomial Compositions of Nonlinear Feedback Shift Registers and their Sequence-Domain Consequences, *Proc. IEE 117,* pp. 1750 – 1756, 1970.
16. C.J.A. Jansen, W.G. Franx, and D.E. Boekee. An Efficient Algorithm for the Generation of de Bruijn Cycles, *IEEE Transactions on Information Theory*, Vol. 37, No. 5, pp. 1475 –1478, September 1991.
17. K. Kjeldsen. On the Cycle Structure of a Set of Nonlinear Shift Registers with Symmetric Feedback Functions, *Journal Combinatorial Theory Series A*, Vol. 20, pp. 154 – 169, 1976.
18. A. Lempel. On a Homomorphism of the de Bruijn Graph and its Applications to the Design of Feedback Shift Registers, *IEEE Transactions on Computers,* Vol. C-19, Issue 12, pp. 1204 - 1209, December 1970.
19. K. Mandal, and G. Gong. Probabilistic Generation of Good Span n Sequences from Nonlinear Feedback Shift Registers, CACR Technical Report, 2012.
20. G.L. Mayhew, and S.W. Golomb. Characterizations of Generators for Modified de Bruijn Sequences, *Advanced Applied Mathematics*, Vol. 13, pp. 454–461, December 1992.
21. J. Mykkeltveit, M-Keung. Siu, and P. Tong. On the Cycle Structure of Some Nonlinear Shift Register Sequences, *Information and Control*, pp. 202 – 215, 1979.
22. T. Rachwalik, J. Szmidt, R. Wicik, J. Zablocki. Generation of Nonlinear Feedback Shift Registers with Special-Purpose Hardware, Report 2012/314, Cryptology ePrint Archive, 2012. `http://eprint.iacr.org/`
23. `http://www.ecrypt.eu.org/stream/ciphers/achterbahn/achterbahn.pdf`