Paper

Boolean feedback functions for full-length nonlinear shift registers

Izabela Janicka-Lipska and Janusz Stokłosa

Abstract—In the paper a heuristic algorithm for a random generation of feedback functions for Boolean full-length shift register sequences is presented. With the help of the algorithm one can generate *n*-stage Boolean full-length shift register sequences for (potentially) arbitrary $n \geq 6$. Some properties of the generated feedback functions are presented.

***Keywords*— cryptography, shift registers, Boolean functions.**

1. Introduction

Nonlinear shift registers generating full-length sequences, also referred to as de Bruijn sequences, have many applications in modern communications systems, especially in cryptography as components of complex devices and algorithms in cipherment and decipherment processes. There exists a number of methods for the generation of full-length sequences (*cf.* [1, 2, 6]).

In the paper we present some results of experiments done on nonlinear Boolean functions. The functions used as feedback functions of shift registers give full-length sequences generated by these shift registers. We generated all functions for *n*-stage shift registers, for $n = 3$, 4, 5 and 6. The experiments led us to the heuristic algorithm for generating *n*-stage full-length shift registers, where the number n of stages is sufficiently great.

2. Preliminaries

Let Z_2^n be n-dimensional vector space over the finite field GF(2). An n-argument Boolean function is a mapping $f : Z_2^n \to Z_2$. Let A_n be the set of all n-argument affine Boolean functions. Every Boolean function which is not affine is said to be nonlinear.

Let $f(x_{n-1}, x_{n-2}, \ldots, x_0) = y_z$ and let z be the decimal equivalent of the function's argument, i.e., such a positive integer that for each argument $(x_{n-1}, x_{n-2}, \ldots, x_0)$ we have

$$z = x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \ldots + x_0 \cdot 2^0 .$$

Then $[y_{2^n-1}, \ldots, y_1, y_0]$ is called the truth table of f. The value y_0 is the least significant value in the truth table and y_{2^n-1} is the most significant value. We divide the table $[y_{2^n-1}, \ldots, y_1, y_0]$ into two subtables: least significant

$$[y_{2^{n-1}-1}, \ldots, y_1, y_0]$$

and most significant

$$[y_{2^n-1}, \ldots, y_{2^{n-1}+1}, y_{2^{n-1}}] .$$

For each truth table of an n-argument Boolean function we compute the decimal value of its 4-bit codes in the following way:

1) divide the truth table into 2^{n-2} words; each word is composed of 4 bits;
2) compute the decimal equivalent for every word;
3) compute the algebraic sum of all decimal equivalents.

As an example let the truth table of 5-argument Boolean function be given:

$$[00000101100101001111101001101011].$$

There are eight 4-bit words in the table:

$$0000\ 0101\ 1001\ 0100\ 1111\ 1010\ 0110\ 1011$$

The decimal equivalents of the words and their sum are as follows:

$$\begin{array}{cccccccccc} [0000 & 0101 & 1001 & 0100 & 1111 & 1010 & 0110 & 1011] \\ 0 & +5 & +9 & +4 & +15 & +10 & +6 & +11 & = 60. \end{array}$$

The definitions presented below are taken from [4]. The Boolean function is said to be balanced if in its truth table the number of ones equals the number of zeros. An n-argument Boolean function f is a function with linear structure if there exists $a \in Z_2^n$ such that $a \neq (0, 0 \ldots, 0)$ and for every $x \in Z_2^n$ either $f(x) \oplus f(x \oplus a) = 0$ or $f(x) \oplus f(x \oplus a) = 1$. The Hamming distance of two n-argument Boolean functions f and g, presented with the help of their truth tables, is the number of positions in which the two truth tables differ. The distance of a function f to the set A_n is defined as the minimum of the Hamming distances to all functions of A_n. The nonlinearity of f, denoted by N_f, is the minimal Hamming distance between f and A_n. If f is n-argument, $n \geq 3$, and it is balanced then [5]:

$$\mathrm{N}_f \leq \begin{cases} 2^{n-1} - 2^{\frac{1}{2}n-1} - 2, & \text{for } n \text{ even} \\ \lfloor \lfloor 2^{n-1} - 2^{\frac{1}{2}n-1} \rfloor \rfloor, & \text{for } n \text{ odd,} \end{cases}$$

where $\lfloor \lfloor x \rfloor \rfloor$ denotes the maximum even integer less than or equal to x.

An n-stage nonlinear feedback shift register over GF(2) (nNFSR for short) consists of n cells ($n \geq 1$) joined as in Fig. 1, where symbols of a nonempty alphabet $\{0, 1\}$ may be put in as a Boolean function f of n arguments. The content of all n cells is said to be a state of nNFSR. The nNFSR works in the discrete time. The state of nNFSR at

a given moment $t+1$ $(t \geq 0)$ is determined by its state at the moment t and results from shifting the content of the cell number r to the cell number $r+1$ $(0 \leq r \leq n-2)$, and putting the value $f(x_{n-1}, x_{n-2}, \ldots, x_0)$ of the function f for the state of this nNFSR at the moment t into the cell number 0.

Let us mention that the numbering of register cells is crucial for the algorithm of feedback functions choosing presented later.

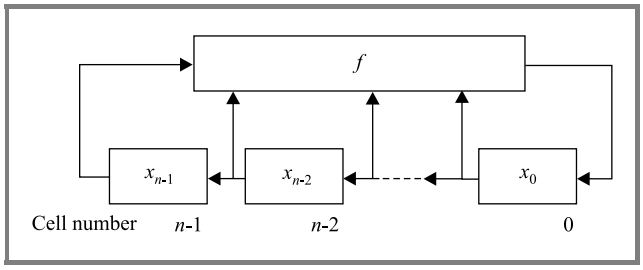

Fig. 1. An n-stage shift register with the feedback function f.

Let $s = (x_{n-1}, x_{n-2}, \ldots, x_0)$ be the state of an nNFSR. The state s_0 from which the nNFSR starts the work is said to be initial. A sequence (s_i) of states of an nNFSR is periodic with period equal to T if T is the smallest positive integer such that for each $i = 1, 2, \ldots$ the condition $s_{i+T} = s_i$ holds. An nNFSR generates periodic sequences of the period $T \leq 2^n$. A sequence of states generated by nNFSR is called a full-length sequence if $T = 2^n$. Each nNFSR generates $B(n) = 2^{2^{n-1}-n}$ full-length sequences [2].

In the sequel a sequence (s_i) of states generated by an nNFSR with a nonlinear n-argument Boolean function f will be called a sequence generated by the function f. An nNFSR that generates a full-length sequence is called full-length shift register.

3. Properties of feedback functions of full-length shift registers

Using the exhaustive search in the set of all n-argument (for $n = 3, 4$ and 5) Boolean functions we chose all functions which give, when used as feedback functions, full-length sequences. We can state the following facts.

Fact 1: The least and the most significant bits of the most significant truth subtable equal 0.

Fact 2: The most significant truth subtable has an odd number of 1s.

Fact 3: The least significant truth subtable is the negation (i.e., respective bits are negated) of the most significant truth subtable.

Fact 4: The sum of decimal equivalents of 4-bit words in the truth tables equals $2^{n+1} - 2^{n-3} = 15 \cdot 2^{n-3}$ (i.e., 15, 30, 60 for $n = 3, 4, 5$, respectively).

Fact 5: Nonlinear n-argument (for $n = 3, 4, 5$) Boolean function generating full-length sequence is balanced (follows from the Fact 3) and is of linear structure.

Fact 6: The greater number of 1s in the truth table the greater value of nonlinearity.

Fact 7: The nonlinearity N_f of obtained function has one of the values: 2, 6, 10, ..., and in general it is equal to $2+4i$ for $i = 0, 1, 2, \ldots$. The nonlinearities never have the maximum value $2^{n-1} - 2^{\frac{1}{2}n-1} - 2$ for n even and $\left\lfloor \left\lfloor 2^{n-1} - 2^{\frac{1}{2}n-1} \right\rfloor \right\rfloor$ for n odd.

There is 2^{26} 6-argument nonlinear Boolean functions generating full-length sequences. Choosing all of them by exhaustive search in the set of 2^{64} 6-argument Boolean functions is difficult with respect to the time needed for computation. Therefore, it was assumed that Facts 1–4 are true also for n-argument $(n \geq 6)$ functions. This assumptions leads to the following algorithm.

4. Algorithm for choosing all n-argument $(n \geq 6)$ nonlinear Boolean functions generating full-length sequences

Input: The most significant truth subtable of n-argument nonlinear Boolean function given by $[y_{2^n-1}, \ldots, y_{2^{n-1}+1}, y_{2^{n-1}}]$.

Output: The set of all n-argument nonlinear Boolean functions (given by the truth tables) generating full-length sequences, $n \geq 6$.

Method:

1. Let $y_{2^n-1} = 0$ and $y_{2^{n-1}} = 0$.
2. For $i = 1, 3, 5, 7, 9, \ldots, 2^{n-1} - 3$ generate in the lexicographical order the words $y_{2^n-2}, \ldots, y_{2^{n-1}+2}$, $y_{2^{n-1}+1}$ having i 1s; for each word:
   a. Construct the most significant subtable.
   b. Construct the least significant subtable by the negation of all bits in the most significant subtable.
   c. Concatenate the tables constructed in steps 2a and 2b.
   d. Verify whether the sum of decimal equivalents of all 4-bit words equals $15 \cdot 2^{n-3}$; if not then process for the next i.
   e. Verify whether the n-stage shift register with the feedback function given with the help of the truth table constructed in the step 2c generates the sequence of period 2^n; if so then store the truth table.

The number of all words of the form $y_{2^n-2}, \ldots, y_{2^{n-1}+2}$, $y_{2^{n-1}+1}$ with i 1s equals $\binom{2^{n-1}-2}{i}$.
Hence, the number of all words having the odd number $1, 3, 5, 7, 9, \ldots, 2^{n-1}-3$ of 1s is equal to

$$s(n) = \binom{2^{n-1}-2}{1} + \binom{2^{n-1}-2}{3} + \binom{2^{n-1}-2}{5} + \ldots + \binom{2^{n-1}-2}{2^{n-1}-3} = 2^{2^{n-1}-3}.$$

If by the efficiency of the algorithm we understand the quotient η of the number $B(n)$ of all n-arguments nonlinear Boolean functions generating full-length sequences and the number $s(n)$ of all examined functions, then

$$\eta = \frac{B(n)}{s(n)} = 2^{-n+3}.$$

For example, if $n = 16$ the efficiency $\eta = 2^{-13}$. It means that on average one function in the set of 8192 functions has the required property. The efficiency of the algorithm is quite satisfactory.
If in the algorithm instead of "generate in the lexicographical order the words $y_{2^n-2}, \ldots, y_{2^{n-1}+2}, y_{2^{n-1}+1}$ having i 1s" we allow "generate randomly the words $y_{2^n-2}, \ldots, y_{2^{n-1}+2}, y_{2^{n-1}+1}$ having i 1s" we can use the algorithm for random generation of Boolean functions generating full-length sequences for an arbitrary n.
The computational experiment confirmed the efficiency of the algorithm for 6-argument functions; for all 6-argument functions Facts 1–7 are true. We verified the randomly generated functions for n from 7 to 20 and in every case Facts 1–7 were true.
The algorithm were successfully used in the synthesis of nNFSRs for FSR-255 family of hash functions [3].

References

[1] T. Etzion and A. Lempel, "Algorithms for the generation of full-length shift register sequences", *IEEE Trans. Inform. Theory*, vol. IT-30, no. 3, pp. 480–484, 1984.

[2] H. Fredricksen, "A survey of full length nonlinear shift registers cycle algorithms", *SIAM Rev.*, vol. 24, no. 2, pp. 195–221, 1982.

[3] T. Gajewski, I. Janicka-Lipska, and J. Stokłosa," The FSR-255 family of hash functions with variable length of hash result", in *Artificial Intelligence and Security in Computing Systems*, J. Sołdek and L. Drobiazgiewicz, Eds. Boston: Kluwer, 2003, pp. 239–248.

[4] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions", in *Advances in Cryptology – EUROCRYPT'89*, J.-J. Quisquater and J. Vandewalle, Eds., *LNCS*. Berlin: Springer, 1990, vol. 434, pp. 549–562.

[5] J. Seberry, X.-M. Zhang, and Y. Zheng, "Nonlinearly balanced Boolean functions and their propagation characteristics", in *Advances in Cryptology – CRYPTO'93*, D. R. Stinson, Ed., *LNCS*. Berlin: Springer, 1994, vol. 773, pp. 49–60.

[6] J.-H. Yang and Z.-D. Dai, "Construction of m-ary de Bruijn sequences (extended abstract)", in *Advances in Cryptology – AUSCRYPT'92*, J. Seberry and Y. Zheng, Eds., *LNCS*. Berlin: Springer, 1993, vol. 718, pp. 357–363.

Izabela Janicka-Lipska is an adjunct at Poznań University of Technology, Poland. Her research interest includes data security in information systems and cryptology, especially methods of designing Boolean functions, hash functions and shift registers. Her doctoral thesis is "Nonlinear feedback functions of maximal shift registers and their application to the design of a cryptographic hash function" (2001, in Polish).
e-mail: Janicka-Lipska@sk-kari.put.poznan.pl
Institute of Control and Information Engineering
Poznań University of Technology
Marii Skłodowskiej-Curie Sq. 5
60-965 Poznań, Poland

Janusz Stokłosa is a Professor at Poznań University of Technology, Poland. His research interest includes data security in information systems and cryptology, especially methods of designing cryptographic algorithms. He is author of a number of publications, also books: *Algebraic and Structural Automata Theory* (North Holland, 1991, coauthor), *Cryptographic Method of Data Protection* (1992, in Polish), *Cryptographic Algorithms* (1994, in Polish), *Data Security in Information Systems* (2001, in Polish, coauthor), *Data Protection and Safeguards in IT Systems* (2003, in Polish, coauthor).
e-mail: Stoklosa@sk-kari.put.poznan.pl
Institute of Control and Information Engineering
Poznań University of Technology
Marii Skłodowskiej-Curie Sq. 5
60-965 Poznań, Poland