Computers and Electrical Engineering 36 (2010) 993–1001

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Modified self-shrinking generator ☆

Ali Kanso *

Department of Mathematics and Computer Science, College of Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait

ARTICLE INFO

Article history:
Received 16 December 2008
Received in revised form 6 January 2010
Accepted 15 February 2010
Available online 12 March 2010

Keywords:
Stream ciphers
Linear feedback shift registers
m-Sequences
Self-shrinking generators

ABSTRACT

The self-shrinking generator *SSG*, an elegant keystream generator proposed by Meier and Staffelbach, is built up from a single n-stage primitive linear feedback shift register (LFSR) to produce a keystream of period $P \geqslant 2^{\lfloor \frac{n}{2} \rfloor}$, and linear complexity greater than half its period. In this article, we propose a new variant of the self-shrinking generator called the modified self-shrinking generator *MSSG*. This new generator is based on a primitive n-stage LFSR and uses an extended selection rule based on the XORed value of a pair of bits. We prove that the keystreams of the *MSSG* are balanced, and have period greater than or equal to $2^{\lfloor \frac{n}{3} \rfloor}$, linear complexity greater than half the period, and possess good statistical properties. We investigate the security of the generator against various powerful cryptanalytic attacks. We show that the *MSSG* is more secure than the *SSG* against most of these attacks. Moreover, experiments show that for odd values of n, $3 < n < 20$, the period of the keystreams generated by the *MSSG* attains its maximum value 2^{n-1}, and the linear complexity of these keystreams is very close to its upper bound. The NIST statistical test suite is applied to several thousands keystreams of the *MSSG* to demonstrate their good randomness properties. The obtained results show that keystreams of the *MSSG* have better randomness properties than those of the *SSG*.

1. Introduction

Due to their efficient hardware implementation, linear feedback shift registers (LFSRs) [1] are involved in most hardware-oriented stream cipher designs such as A5, shrinking generator, self-shrinking generator, etc. [2]. One design technique to obtain suitable keystreams for stream cipher applications involves applying Boolean functions on the outputs or the inner states of one or several LFSRs. Another technique is to use one LFSR to control outputs of one or several LFSRs. However, these techniques may result in a stream cipher that fails to meet the high performance requirements.

The technique of irregularly decimating the output of an LFSR is another way of producing suitable keystreams for cryptographic applications. Two well-known keystream generators based on this technique are the shrinking generator *SG* [3] and the self-shrinking generator *SSG* [4]. These two generators are attractive since they are fast, simple to implement, and have good cryptographic properties. These properties make them appropriate for use in light-weight and low-cost stream cipher applications.

In this paper, we propose a modified self-shrinking generator *MSSG* intended for hardware implementation. We show that the self-shrunken sequences (i.e., the keystreams) of the *MSSG* are balanced, and have period greater than or equal to $2^{\lfloor \frac{n}{3} \rfloor}$, and linear complexity greater than half the period. Although the keystreams of the *MSSG* have lower bounds on the period and linear complexity that are less than those of the *SSG*, they are shown to provide a higher level of security against various

☆ Reviews processed and proposed for publication to the Editor-in-Chief by Associate Editor Dr. M. Malek.
* Tel.: +965 2 498 5357.
E-mail address: akanso@hotmail.com

0045-7906/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2010.02.004

well-known cryptanalytic attacks. Furthermore, the keystreams of the *MSSG* are demonstrated (using the NIST test suite) to possess better randomness properties than those of the *SSG*. Moreover, as the self-shrinking generator, the period of keystreams of the modified version has been investigated for odd values of n, $3 \leqslant n < 20$, and as a result it turned out that all tested values of n (except $n = 3$) yield a maximum period 2^{n-1}. For the case $n = 3$, as for the *SSG*, the period was found to be the minimum value 2 instead of the maximum value 4.

This paper is organized as follows: Section 2 describes the self-shrinking generator *SSG* and reviews the results on the period and the linear complexity of its keystreams. It also includes the complexity of various well-know cryptanalytic attacks on these keystreams. Section 3 presents the modified self-shrinking generator. In Section 4, we show that the self-shrunken sequences are balanced and prove the lower bounds on the period and linear complexity of these sequences. Moreover, we provide lower bounds on the appearance of strings of length $k \leqslant \lfloor \frac{n}{3} \rfloor$. In Section 5, we present some cryptanalytic attacks and analyze their effect on the self-shrunken sequences. Section 6 consists of some experimental results. Finally, we conclude with the summary of the results proved in this paper.

2. The self-shrinking generator

At EUROCRYPT'94, Meier and Staffelbach [4] proposed the self-shrinking generator *SSG*, an elegant keystream generator based on the shrinking principle [3]. The *SSG* is attractive as it has remarkably low hardware requirements, very simple structure and good cryptographic properties. The *SSG* uses only one primitive linear feedback shift register (LFSR) to generate a pseudorandom binary keystream according to the following rule: Let A be a primitive LFSR of length n. Let $\underline{a}_0$ and $f(x)$ denote the initial state and the primitive characteristic feedback polynomial of A, respectively. Let $(a_t) = a_0, a_1, a_2, \ldots$ be the binary sequence generated by this LFSR. At time i, we consider the bit-pair (a_{2i}, a_{2i+1}) from the output of A. If the bit $a_{2i} = 1$, output a_{2i+1} as a keystream bit of the *SSG*, otherwise no output is produced. Denote by (s_t) the keystream of the *SSG*.

For example, suppose that the output sequence (a_t) of a primitive LFSR of length 4 is the sequence $(a_t) = 1111000100110101111000100110 10\ldots$ of period $(2^4 - 1)$. Then, the output sequence generated by the *SSG* that is based on this LFSR is the sequence $(s_t) = 1111000011110000\ldots$ of period 2^3.

The basic security requirements of the keystream sequences of a stream cipher include long period, high linear complexity and good statistical properties. One of the most interesting aspects of the *SSG* is that reasonable bounds on the period and the linear complexity of the keystream (s_t) may be easily achieved. Furthermore, (s_t) possesses good statistical properties. Meier and Staffelbach [4] have shown that the period P of (s_t) is lower bounded by $2^{\lfloor \frac{n}{2} \rfloor}$ and upper bounded by 2^{n-1}, and the linear complexity L of (s_t) is lower bounded by $2^{\lfloor \frac{n}{2} \rfloor - 1}$. In [5], Blackburn has proven that the linear complexity L of (s_t) is at most $\lceil 2^{n-1} - (n-2) \rceil$.

From 1994, a lot of efforts have been made to analyze the security of the *SSG* [4,6–11], resulting in more and more powerful attacks on it. In [4], two simple methods have been introduced for attacking the *SSG*, namely, exhaustive search and entropy attack, whose time complexities are $O(2^{0.79n})$ and $O(2^{0.75n})$, respectively. Later, Zenner and Krause [7] have reduced the time complexity to $O(2^{0.695n})$. In [8], Krause has introduced the BDD-attack which has time complexity $O(2^{0.656n})$ at the expense of $O(2^{0.656n})$ from $\lceil 2.41n \rceil$ keystream bits. The BDD-attack was later improved by Hell and Johansson [10]. The main advantage of the last attack over the existing BDD-attack is to have almost the same time complexity with only $O(n^2)$ memory from n keystream bits. In 2006, Zhang and Feng [11] proposed a new guess-and-determine attack on the *SSG*. This attack can recover the initial state of the LFSR defining the *SSG* with time complexity $O(2^{0.556n})$, memory complexity $O(n^2)$ from $O(2^{0.161n})$ keystream bits for $n \geqslant 100$, and time complexity $O(2^{0.571n})$, memory complexity $O(n^2)$ from $O(2^{0.194n})$ keystream bits for $n < 100$.

Although the *SSG* has not yet been completely broken, its security has been challenged little by little and the rapid progress in the computing power and the cryptanalysis techniques bring our attention to search for alternative keystream generators. Hu and Xiao [12] have proposed a variant of the self-shrinking generator called generalized self-shrinking generator. However, the security of this variant has been investigated [13,14] and it turned out that this generator cannot be more secure than the *SSG*.

3. Modified self-shrinking generator

The modified self-shrinking generator is based on one single primitive LFSR of length n and is structured as follows: let $\underline{a}_0$ and $f(x)$ denote the initial state and the primitive characteristic feedback polynomial of A, respectively. Let $(a_t) = a_0, a_1, a_2, \ldots$ be the binary sequence generated by this LFSR. Therefore, (a_t) is an m-sequence of period $(2^n - 1)$ (see [1]). At time i, we consider the bit-triple $(a_{3i}, a_{3i+1}, a_{3i+2})$ from the output of A. If the bit $a_{3i} \oplus a_{3i+1} = 1$, output a_{3i+2} as a keystream bit of the *MSSG*, otherwise no output is produced. Let (z_t) denote the keystream of the *MSSG*. We refer to (z_t) as the self-shrunken sequence.

For example, suppose that the output sequence (a_t) of a primitive LFSR of length 5 is the sequence $(a_t) = 1111100011011010100001001011001 11\ldots$ of period $(2^5 - 1)$. Then, the output sequence generated by the *MSSG* that is based on this LFSR is the sequence $(z_t) = 1100100101110010110010010111 00\ldots$ of period 2^4.

Remark 1. On average, the modified self-shrinking generator requires six clocks to produce one output bit, while the self-shrinking generator requires four clocks, on average, to produce one output bit.