LILI Keystream Generator

L. Simpson[1], E. Dawson[1], J. Golić[2], and W. Millan[1]
[1] Information Security Research Centre, Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia
Email {simpson, dawson, millan}@fit.qut.edu.au
[2] Faculty of Electrical Engineering, University of Belgrade
Bulevar Revolucije 73, 11001 Belgrade, Yugoslavia
Email golic@galeb.etf.bg.ac.yu

Abstract. A family of keystream generators, called the LILI keystream generators, is proposed for use in stream cipher applications and the security of these generators is investigated with respect to currently known attacks. The design is simple and scalable, based on two binary linear feedback shift registers combined in a simple way, using both irregular clocking and nonlinear functions. The design provides the basic security requirements such as a long period and high linear complexity, and is resistant to known cryptanalytic attacks.

1 Introduction

In this paper, a family of keystream generators based on irregularly clocked LFSRs, intended for use in stream cipher applications, is proposed. We call these the LILI generators. The security of the LILI keystream generators is investigated with respect to currently known attacks on stream ciphers. The keystreams produced are shown to possess the basic security requirements for cryptographic sequences, such as a long period and high linear complexity. It is shown that, provided suitable parameters are selected, the generators are resistant to currently known cryptanalytic attacks. Security implications of parameter selection are discussed.

The LILI family of keystream generators are based on two binary linear feedback shift registers (LFSRs). Many keystream generator designs are based on shift registers, both for the simplicity and speed of LFSR implementation in hardware and for the long period and good statistical properties LFSR sequences possess. To make use of the good keystream properties while avoiding the inherent linear predictability of LFSR sequences, many constructions introduce nonlinearity, by applying a nonlinear function to the outputs of regularly clocked LFSRs or by irregular clocking of the LFSRs [13]. However, keystream generators using regularly clocked LFSRs are susceptible to correlation attacks, including fast correlation attacks, a concept first introduced in [11]. In a fast correlation attack, the initial states of the component shift registers are reconstructed from a known segment of the generator output sequence, without performing a blind search over all possible shift register initial states. As a means of achieving immunity to these correlation attacks, keystream generators consisting of irregularly clocked LFSRs were proposed. These keystream generators are also susceptible to certain correlation attacks, such as the generalised correlation attack proposed in [6]. However, no fast correlation attacks on these generators have been published.

As correlation attacks have been successful against keystream generators based on either a nonlinear function of regularly clocked LFSR sequences [16, 14] or on irregular clocking of LFSRs [6, 17], both approaches are combined for the LILI keystream generators. The use of both nonlinear functions and irregular clocking is not novel, having been employed in previous constructions such as ORYX [19] and SOBER [12]. Both ORYX and SOBER are designs for single generators, with fixed size LFSRs and fixed combining functions. In contrast, this proposal is scalable and so describes a family of keystream generators. Also, weaknesses in the design of ORYX resulted in the provision of a very low level of cryptographic security [20]. Some attacks on the SOBER proposal have also been identified [3]. Although the design for the LILI keystream generators described in this paper is conceptually simple, it produces output sequences with provable properties with respect to basic cryptographic security requirements and also provides security against currently known cryptanalytic attacks.

2 Description of LILI Keystream Generators

The LILI keystream generators are simple and fast keystream generators that use two binary LFSRs and two functions to generate a pseudorandom binary keystream sequence, as illustrated in Figure 1. The components of the keystream generator can be grouped into two subsystems based on the functions they perform: clock control and data generation. The LFSR for the clock-control subsystem is regularly clocked. The output of this subsystem is an integer sequence which controls the clocking of the LFSR within the data-generation subsystem. If regularly clocked, the data-generation subsystem is a simple nonlinearly filtered LFSR [13] (nonlinear filter generator). Hence the LILI generator may be viewed as a clock-controlled nonlinear filter generator. Such a system, with the clock control provided by a stop-and-go generator, was examined in [4]. However, the use of stop-and-go clocking produces repetition of the nonlinear filter generator output in the keystream, which may permit attacks. This system is an improvement on that proposal, as stop-and-go clocking is avoided.

The clock-control subsystem of the keystream generator uses a pseudorandom binary sequence produced by a regularly clocked LFSR, $LFSR_c$, of length L_c and a function, f_c, operating on the contents of k stages of $LFSR_c$ to produce a pseudorandom integer sequence, $c = \{c(t)\}_{t=1}^{\infty}$. For practical applications, it is assumed that the feedback polynomial of $LFSR_c$ is primitive and that the initial state of $LFSR_c$ is not the all zero state. Then $LFSR_c$ produces a maximum-length sequence of period $P_c = 2^{L_c} - 1$. At time instant t, the contents of a fixed set of k stages of $LFSR_c$ are input to f_c and the output of f_c is an integer $c(t)$, such that $c(t) \in \{1, 2, \ldots, 2^k\}$. The function f_c is a bijective mapping

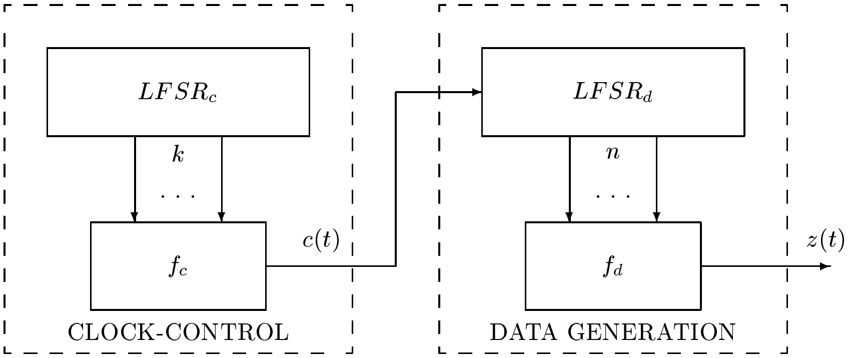

Fig. 1. LILI keystream generator.

$\{0,1\}^k \to \{1,\ldots,2^k\}$, so that the distribution of integers $c(t)$ is close to uniform. Thus $c = \{c(t)\}_{t=1}^{\infty}$ is a periodic integer sequence with period equal to P_c. For example, $f_c(x_1,\ldots,x_k) = 1 + x_1 + 2x_2 + \ldots + 2^{k-1}x_k$ is appropriate.

The variable parameters in the clock-control subsystem are L_c, the feedback function of $LFSR_c$, k, the positions of k stages of $LFSR_c$ used as inputs to the clocking function f_c and f_c itself.

The data-generation subsystem of the keystream generator uses the integer sequence c produced by the clock-control subsystem to control the clocking of a binary LFSR, $LFSR_d$, of length L_d. At time instant t, $LFSR_d$ is clocked $c(t)$ times. The contents of a fixed set of n stages of $LFSR_d$ are input to a Boolean function, f_d. The binary output of f_d forms the keystream bit $z(t)$. After $z(t)$ is produced, $LFSR_c$ is clocked and the process repeated to form the keystream $z = \{z(t)\}_{t=1}^{\infty}$.

If $LFSR_d$ is regularly clocked, then the data-generation subsystem is simply a nonlinear filter generator. It is assumed that the feedback polynomial of $LFSR_d$ is primitive and that the initial state of $LFSR_d$ is not the all zero state. Then $LFSR_d$ produces a maximum-length sequence of period $P_d = 2^{L_d}-1$. The output of a regularly clocked nonlinear filter generator is a periodic binary sequence, $g = \{g(i)\}_{i=1}^{\infty}$, with period dividing P_d. The following basic result is proved in [18].

Theorem 1. *Let $LFSR_d$ have a primitive feedback polynomial and a nonzero initial state. If f_d is balanced, or if P_d is a prime and f_d is not a constant function (zero or one), then the period of g is P_d.*

Now, considering the irregular clocking of $LFSR_d$, the keystream z may be viewed as an irregularly decimated version of the nonlinearly filtered $LFSR_d$ sequence g, with the decimation under the control of $LFSR_c$, so that $z(t) = g(\sum_{j=1}^{t} c(j))$.

The variable parameters in the data-generation subsystem are L_d, the feedback function of $LFSR_d$, n, the positions of n stages of $LFSR_d$ used as inputs to the filter function f_d and f_d itself. The function f_d should be balanced, highly nonlinear and offer some order of correlation immunity relative to the positions of n stages used as inputs to f_d (see [9]). The nonlinearity of a Boolean function is defined to be the minimum Hamming distance between the function and any affine function.

3 Keystream Properties

Several properties of pseudorandom binary sequences are considered basic security requirements: a sequence that does not possess these properties is generally considered unsuitable for cryptographic applications. Basic requirements for pseudorandom binary sequences are a long period, high linear complexity and good statistics regarding the distribution of zeroes and ones in the output. High linear complexity avoids an attack using the Berlekamp-Massey [10] algorithm, which requires a length of keystream only twice the linear complexity of the sequence to produce the entire keystream. A bias in the distribution of zeroes and ones in the keystream can be used to reduce the unpredictability of the keystream sequence. These basic requirements are addressed with respect to the LILI family of keystream generators in the remainder of this section.

3.1 Period

The maximum value for the period of z and the conditions under which this value is obtained are given in the following theorem. The result is easily obtained from Theorem 1 and the application of a result regarding the period of irregularly decimated sequences from [2].

Theorem 2. *Let both $LFSR_c$ and $LFSR_d$ have primitive feedback polynomials and nonzero initial states. If $2^{L_d}-1$ is a prime and f_d is not a constant function or if f_d is balanced and $2^{L_c-1}(2^k+1)-1$ is relatively prime to $2^{L_d}-1$ (provided that $f_c(0,\ldots,0)=1$), then the period of the output sequence z is given by the product $P_z=(2^{L_c}-1)(2^{L_d}-1)$.*

Note that this period implies that each distinct initial state results in the production of a distinct keystream, avoiding the reduction in keyspace which commonly occurs in keystream generators using irregular clocking, where several initial states produce the same keystream [17, 12].

3.2 Linear Complexity

For the proposed keystream generator, the output of a nonlinear filter generator with period $P_d=2^{L_d}-1$ or a divisor of P_d is nonuniformly decimated by means of a sequence with period $P_c=2^{L_c}-1$. In [5], the following upper bound on the linear complexity of irregularly decimated maximum-length sequences is given. When a maximum-length sequence of period P_d is nonuniformly decimated by means of a decimating sequence of period P_c, if the sum modulo P_d of P_c successive values of the decimating sequence equals S, then the decimated sequence has a maximum linear complexity of $L_d \cdot P_c$ only if the multiplicative order of 2 modulo $P_d / \gcd(P_d, S)$ is equal to L_d. Note that this condition is satisfied if $\gcd(P_d, S) = 1$. In [5] it is also shown that if the decimating sequence is randomly chosen, then the probability that maximum linear complexity is obtained can be made arbitrarily close to one for appropriately chosen L_d and P_c.

For a nonuniformly decimated nonlinearly filtered LFSR sequence, the maximal attainable linear complexity is $L'_d \cdot P_c$, where L'_d is the linear complexity of the (regularly clocked) nonlinearly filtered $LFSR_d$ sequence. It is known (e.g., see [13]) that L'_d depends on the filter function and on the positions of stages used for its inputs and that L'_d is very likely to be lower bounded by $\binom{L_d}{r}$, where r is the nonlinear algebraic order of the filter function. Accordingly, our conjecture is that the linear complexity of a nonuniformly decimated nonlinearly filtered $LFSR_d$ sequence is very likely to be lower-bounded by $\binom{L_d}{r} \cdot P_c$. As a consequence, it is also lower-bounded by $L_d \cdot P_c$.

To investigate this conjecture, computer simulations were performed for keystream generators as described in Section 2, for various small shift register lengths. In each case, a nonlinear 3-input balanced nonlinear Boolean function, with $r = 2$, was used as a nonlinear combining function, and the stages of $LFSR_d$ used for inputs to the filter function were selected to form a full positive difference set. That is, the distances between any two stages are distinct. For each keystream generator, a keystream sequence of length greater than the maximum period of the keystream was produced and the period, P_z, and linear complexity, L_z, of the sequence were determined. These values are recorded in Table 1, and support both the theorem regarding the period and the conjecture regarding the linear complexity.

$k = 2$					$k = 3$				
L_c	L_d	P_z	L_z	$\binom{L_d}{2} \cdot P_c$	L_c	L_d	P_z	L_z	$\binom{L_d}{2} \cdot P_c$
3	4	105	64	42	4	4	225	150	90
3	6	441	147	105	4	6	945	303	225
3	7	889	196	147	4	7	1905	420	315
3	12	28665	546	462	4	12	61425	1170	990
7	4	1905	1001	762	6	4	945	503	378
7	6	8001	2667	1905	7	6	8001	2373	1905
7	7	16129	3556	2667	7	7	16129	3556	2667

Table 1. Period and linear complexity of binary sequences produced by LILI keystream generators.

3.3 Statistical Properties of Output Sequence

Under regular clocking, one period of the sequence d produced by $LFSR_d$ when regularly clocked contains $2^{L_d-1}-1$ zeroes and 2^{L_d-1} ones. For a balanced filter function such that $f_d(0,\ldots,0)=0$, a segment of length $2^{L_d}-1$ of the regularly clocked nonlinear filter generator output sequence g has the same distribution of zeroes and ones as d. When the clocking of $LFSR_d$ is under the control of $LFSR_c$ and when the period of z is $(2^{L_c}-1)(2^{L_d}-1)$, then each pair of $LFSR_c$ and $LFSR_d$ states occurs exactly once in a period of z. Therefore one period of z contains $(2^{L_c}-1)(2^{L_d-1}-1)$ zeroes and $(2^{L_c}-1)2^{L_d-1}$ ones, thus maintaining the same proportion of zeroes and ones as in d.

At a more detailed level, the choice of filter function has an effect on the keystream statistics. For a regularly clocked nonlinear filter generator the output sequence may not possess good statistics as the inputs to the filter function are correlated rather than independent. To guarantee good statistical properties, the nonlinear filter function can be chosen to be linear in either the first or the last variable [9].

3.4 Throughput Rate

In producing the keystream, $LFSR_d$ is clocked $c(t)$ times before $z(t)$ is produced. Thus $LFSR_d$ is clocked at least once and at most 2^k times before each keystream bit is produced, with the distribution of values of $c(t)$ almost uniform. Over one period of c, $LFSR_d$ is clocked $\sum_{t=1}^{P_c} c(t) = 2^{L_c-1}(2^k+1)-1$ times so, on average, $LFSR_d$ is clocked $\frac{2^{L_c-1}(2^k+1)-1}{2^{L_c}-1}$ times per keystream symbol produced. For large L_c, this is approximately $\frac{2^k+1}{2}$. Thus, for large L_c, the throughput rate is approximately $\frac{2}{2^k+1}$ of the rate at which $LFSR_d$ is clocked, provided an appropriate buffer is used. If not, then one must allow 2^k clocks of $LFSR_d$ per each keystream bit. However, the use of a buffer is very sensitive in high-speed applications.

Alternatively, to achieve the the maximum throughput rate of 1, instead of irregularly clocking the shift register a given number of steps, multiple copies of the feedback function can be maintained, one for each possible value of $c(t)$. The irregular clocking can then be performed in one step only (both in hardware and software). Thus there is a tradeoff between hardware space and timing regularity. Note that the use of either a buffer or parallel-feedback method would provide resistance against timing attacks.

4 Possible Attacks

A number of attacks should be considered with respect to the LILI family of keystream generators. These are known-plaintext attacks conducted under the assumption that the cryptanalyst knows the complete structure of the generator, and the secret key is only the initial states of the component shift registers. For all attacks, the given keystream is viewed as an irregularly decimated version of a nonlinearly filtered $LFSR_d$ sequence, with the decimation under the control of $LFSR_c$. For keystream generators based on more than one LFSR where the key consists of the initial states of the LFSRs, such as the LILI generators, divide-and-conquer attacks on individual LFSRs should be considered. We deal firstly with divide-and-conquer attacks that target $LFSR_d$, and then with those attacks that target $LFSR_c$.

4.1 Attacks on Irregularly Clocked $\boldsymbol{LFSR_d}$

Suppose a keystream segment of length N is known, say $\{z(t)\}_{t=1}^{N}$. This is a decimated version of a segment of length M of the underlying regularly clocked nonlinearly filtered $LFSR_d$ sequence, $g = \{g(i)\}_{i=1}^{M}$, where $M \geq N$. The objective of correlation attacks targeting $LFSR_d$ is to recover the initial state of $LFSR_d$ by identifying the segment $\{g(i)\}_{i=1}^{M}$ that $\{z(t)\}_{t=1}^{N}$ was obtained from through decimation, using the correlation between the regularly clocked sequence and the keystream, without knowing the decimating sequence.

For clock-controlled shift registers with constrained clocking, correlation attacks based on a constrained Levenshtein distance and on a probabilistic measure of correlation are proposed in [6] and [7], respectively, and further analysed in [8]. These attacks could be adapted to be used as the first stage of a divide-and-conquer attack on the LILI keystream generators.

For a candidate initial state of $LFSR_d$, say $\{\hat{d}(i)\}_{i=1}^{L_d}$, use the known $LFSR_d$ feedback function to generate a segment of the $LFSR_d$ sequence, $\{\hat{d}(i)\}_{i=1}^{M+L_d-1}$, for some $M \geq L_d$. Then use the known filter function f_d to generate a segment of length M of the output of the nonlinear filter generator when regularly clocked, $\{\hat{g}(i)\}_{i=1}^{M}$. A measure of correlation between $\{\hat{g}(i)\}_{i=1}^{M}$ and $\{z(t)\}_{t=1}^{N}$ is calculated, (either the Constrained Levenshtein Distance (CLD) [6], or the Probabilistic Constrained Edit Distance (PCED) [7]) and the process repeated for all $LFSR_d$ initial states.

In either case, the attack is considered successful if only a few initial states are identified. As the correlation attack based on the PCED takes into account the probability distribution of the decimating sequence, it is statistically optimal and may be successful in cases where the embedding attack based on the CLD is not, such as for larger values of k. The value of M is a function of N and k. If $M = 2^k \times N$, then the probability of not identifying the correct $LFSR_d$ initial state is zero.

The second stage of a divide-and-conquer attack on the generator is the recovery of the initial state of the second shift register. This can be performed as in [17]. From the calculation of the edit distance (either CLD or PCED) between $\{\hat{g}(i)\}_{i=1}^{M}$ and $\{z(t)\}_{t=1}^{N}$, form the edit distance matrix, and use this to find possible edit sequences. From each possible edit sequence, form a candidate integer sequence $\{\hat{c}(t)\}_{t=1}^{N}$. From this, the underlying binary sequence $\{\hat{a}(t)\}_{t=1}^{N}$ and hence the candidate initial state of $LFSR_c$ can be recovered. To determine whether the correct initial states of both LFSRs have been recovered, use both candidate initial states to generate a candidate keystream and compare it with the known keysteam segment.

To conduct either of these correlation attacks requires exhaustive search of $LFSR_d$ initial states. For each $LFSR_d$ initial state, the attacks require calculation of either the CLD or the PCED, with computational complexity $O(N(M-N))$. Finally, further computational complexity is added in finding the corresponding $LFSR_c$ initial state. For either correlation attack, the minimum length of keystream required for a successful attack on $LFSR_d$ is linear in L_d, but exponential or even superexponential in 2^k (see [8]). For $k=1$, the required keystream length [22] is reasonably small, but a small increase in k will render this length prohibitively large.

4.2 Attacks Targeting $\boldsymbol{LFSR_c}$

A possible approach to attacking the proposed generator is by targeting the clock-control sequence produced by $LFSR_c$. Guess an initial state of $LFSR_c$, say $\{\hat{a}(t)\}_{t=1}^{L_c}$. Use the known $LFSR_c$ feedback function and the function f_c to generate the decimating sequence $\{\hat{c}(t)\}_{t=1}^{N}$ for some $N \geq L_c$. Then position the known keystream bits $\{z(t)\}_{t=1}^{N}$ in the corresponding positions of $\{\hat{g}(i)\}_{i=1}^{\infty}$, the nonlinear filter generator output when regularly clocked. At this point we have some (not all consecutive) terms in the nonlinear filter generator output sequence and are trying to reconstruct a candidate initial state for $LFSR_d$. The attack could then proceed in several ways.

Consistency Attack One method is to use the known filter function f_d to write equations relating terms in the underlying $LFSR_d$ sequence to terms in $\{\hat{g}(i)\}_{i=1}^{\infty}$. Reject the guessed initial state $\{\hat{c}(t)\}_{t=1}^{L_c}$ when the equations are inconsistent. This is a generalisation of the linear consistency test [21]. The feasibility of such an approach depends on the number, n, of inputs to f_d, on the tap positions producing these inputs and on some properties of f_d such as its nonlinearity and order of correlation immunity.

Attacks on Regularly Clocked $\boldsymbol{LFSR_d}$ An alternative approach would be to use a correlation attack on the nonlinear filter generator [14] to recover a linear transform of the $LFSR_d$ sequence, and then recover the $LFSR_d$ initial state. However, this is complicated by not having consecutive terms in the regularly clocked nonlinear filter generator sequence. The feasibility of such an attack primarily depends on the use of a feedback polynomial of $LFSR_d$ that is of low weight or has low weight polynomial multiples and on the nonlinearity of f_d.

An alternative correlation attack on a (regularly clocked) nonlinear filter generator which could be applied at this point is the conditional correlation attack [1], with a difference that the known output bits are not consecutive. The feasibility of such an attack depends on n and on the tap positions. The use of a full positive difference set for the tap positions, as suggested in [9], and of filter functions with correlation-immunity order greater than zero would render this attack infeasible.

Finally, the inversion attack [9] can be adapted to deal with the case of non-consecutive output bits, but the associated branching process is then supercritical, because more than one bits have to be guessed at a time. As a consequence, the computational complexity may be prohibitively high even if the tap positions are not spread across the $LFSR_d$ length.

Applying any of these approaches requires exhaustive search over the $LFSR_c$ initial state space and additional computation for each candidate $LFSR_c$ state. However, as only some (not all consecutive) terms in the nonlinear filter generator output sequence are available, the required additional computation appears to be prohibitive, especially for highly nonlinear filter functions with a large number of inputs and sufficiently high correlation-immunity order, for the tap positions chosen according to a full positive difference set and for the feedback polynomial of $LFSR_d$ not having low-weight polynomial multiples of relatively small degrees.

5 Choice of Parameters

As an initial security consideration, we should choose the sizes of the shift registers so that exhaustive search of the initial states is prohibitive; at present we recommend that $L_c + L_d > 100$. For a keysize in line with the AES specifications for block ciphers, use $L_c + L_d = 128$. To prevent divide-and-conquer attacks, neither L_d nor L_c should be small. To ensure a large period and good statistical properties, the feedback polynomials of both $LFSR_c$ and $LFSR_d$ should be primitive. In addition, as noted in Section 3, for generator parameters satisfying the conditions of Theorem 2, the period of the output sequence z attains a maximum value of $(2^{L_c} - 1)(2^{L_d} - 1)$, implying that every initial state generates a distinct keystream. Furthermore, the selection of parameters should reduce the possibility of the attacks discussed in Section 4. We address each subsystem of the keystream generators in turn.

5.1 Clock Control

The number, k, of taps from $LFSR_c$ used to form the clocking sequence c affects the period of the output sequence and the resistance against the correlation attacks on irregularly clocked $LFSR_d$, described in Section 4.1, and is the sole factor determining the output rate of the generator. To this end, we recommend $k > 1$ (e.g., $k = 2$ or $k = 3$). The choice of the tap positions does not seem to be important with respect to known attacks, but to be on the safe side, we recommend the use of full positive difference sets.

Also, if the conditions of Theorem 2 are not satisfied, then the period of z is upper-bounded by the product of the period of c and any factors of the period of the nonlinear filter generator output (if regularly clocked) which are relatively prime to $2^{L_c - 1}(2^k + 1) - 1$. Thus, for any chosen value of k, $\gcd(2^{L_c - 1}(2^k + 1) - 1, 2^{L_d} - 1)$ should be calculated, and the keystream period is maximised when this is one.

5.2 Data Generation

Firstly, the feedback polynomial of $LFSR_d$ should not have low-weight polynomial multiples of relatively small degrees, in order to avoid the vulnerability to fast correlation attacks on $LFSR_d$ when regularly clocked.

Secondly, the number, n, and positions of taps for the filter function, f_d, should be chosen so as to ensure the resistance to attacks discussed in Section 4.2. For example, we recommend that $n \geq 10$ and that the tap positions form a full positive difference set if possible.

Thirdly, the filter function, f_d, should be balanced in order to achieve good statistical properties and a large period (Theorem 1).

Fourthly, f_d should be chosen so as to reduce the possibility of attacks discussed in Section 4.2 (especially if $k = 1$). To this end, f_d should have high correlation-immunity order and high nonlinearity. The proportion of balanced Boolean functions which offer any nonzero order of correlation immunity is small, making it unlikely that a randomly generated function will meet these criteria. Instead, a filter function should be constructed to obtain the required properties. Since there are tradeoffs between nonlinearity, correlation immunity, and algebraic order,we seek functions that optimise these bounds.

In [15], it was proven that balanced, correlation immune order 3 Boolean functions exist with 10 inputs, order 6 and nonlinearity 480. In the same paper a function with CI(1), order 8 and nonlinearity 484 was constructed. Both of these Boolean functions maximise the Siegenthaler tradeoff and they have the highest possible nonlinearity for their given order of correlation immunity, so either would be a good choice for the output function f_d. For our example, we choose a CI(3) function as we believe that gives a greater resistance to conditional correlation attacks.

6 Example

For a 128-bit key, we select the lengths of $LFSR_c$ and $LSFR_d$ to be 39 and 89 respectively. The feedback polynomials of both $LFSR_c$ and $LFSR_d$ are the primitive polynomials of degrees 39 and 89 respectively listed in the Appendix.

For the clock-control subsystem, the length of $LFSR_c$ is $L_c = 39$, from which $k = 2$ bits are selected to determine the number of data clocks by the natural mapping: $f_c = x_1 + 2x_2 + 1$.

For the data-generation subsystem, we let $n = 10$. Now, we have $L_d \geq 80$ and this permits the positions of inputs to f_d to form a full positive difference set, shown in the Appendix. Also, we select $f_d(x)$ from [15] to be a balanced, CI(3) function of 10 inputs, with nonlinearity 480, and algebraic order $r = 6$ (see the Appendix for the truth table).

6.1 Properties

As the feedback polynomial of $LFSR_d$ is primitive, f_d is balanced and in addition $2^{89} - 1$ is a Mersenne prime, the conditions of Theorem 2 are satisfied. Thus the period of the keystream is $P_z = (2^{39} - 1)(2^{89} - 1)$. According to Section 3.2, the linear complexity of the keystream sequence is conjectured to be at least $\binom{L_d}{r} \cdot P_c = \binom{89}{6} \cdot (2^{39} - 1) \approx 2^{68}$. With regard to the security offered by this value, we note that this means that 2^{69} known plaintext bits must be intercepted in order to perform the Berlekamp-Massey [10] attack. As the key will be changed well before even a fraction of this amount of data is generated, LILI is considered to be secure from such an attack

6.2 Possible Attacks

Both the period and the conjectured linear complexity of the keystream are too large to be used in cryptanalytic attacks.

The choice of parameters for the data-generation subsystem, in particular the Boolean function f_d, make attacks targetting $LFSR_c$, outlined in Section 4.2, infeasible. In [14], fast correlation attacks on regularly clocked nonlinear filter generators with low-weight feedback polynomials and a known keystream segment of 20,000 bits were not successful when the probability of noise, p, exceeded 0.45. The computational complexity of these attacks is proportional to the length of keystream used and the average number of parity checks used per keystream bit. For the assumed function f_d, the probability of noise is given as $p = 0.46875$, so that the amount of keystream required would be much greater than 20,000 bits. This is likely to make the complexity of an attack on a regularly clocked nonlinear filter generator prohibitive, even if enough low-weight polynomial multiples of the $LFSR_d$ feedback polynomial, used to form parity checks, could be obtained. Given that the keystream segment is from a clock-controlled nonlinear filter generator and that the $LFSR_d$ feedback polynomial does not have low-weight polynomial multiples, such an attack appears infeasible.

The length of $LFSR_d$ makes attacks targetting $LFSR_d$, outlined in Section 4.1, infeasible as these attacks require exhaustive search of the initial states of $LFSR_d$, performing some calculation of the correlation for each state. The complexity of such attacks is $O((2^{89} - 1)(3N^2))$, where the required length of the known keystream, N, is very likely to be very large even for $k = 2$. In [17], successful probabilistic correlation attacks were performed on the shrinking generator for given keystream lengths of twenty times the length of the underlying LFSR. The deletion rate for this example is similar, so an estimate of the complexity of these attacks is $O(2^{112})$.

7 Conclusion

In this paper, a family of keystream generators, intended for use in stream cipher applications, is proposed. The design is both simple and scalable: the generators are based on two binary LFSRs and use two combining functions. The security of these keystream generators is investigated. For appropriately chosen components, the generators are shown to provide the basic security requirements for cryptographic sequences, such as a long period and high linear complexity. Also, they are immune to current known-plaintext attacks, conducted under the assumption that the cryptanalyst knows the entire structure of the generator and the secret key is only the initial states of the two LFSRs.

To select an instance from the proposed family, it is necessary to select appropriate values for L_c, L_d, k and n, to have primitive feedback polynomials of both LFSRs and a highly nonlinear balanced Boolean function of an appropriate correlation-immunity order for the filter function. The selection of components can maximise the period and minimise the chances of a successful cryptanalytic attack. The use of both nonlinear combining functions and irregular clocking in LFSR based stream ciphers is not a novel proposal, and has been employed in previous constructions. However, in this proposal the two approaches are combined in a manner that produces output sequences with provable properties with respect to basic cryptographic security requirements and also provides security against currently known cryptanalytic attacks.

References

1. R. Anderson. Searching for the Optimum Correlation Attack. In *Fast Software Encryption - Leuven'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 137–143. Springer–Verlag, 1995.
2. G. R. Blakley and G. B. Purdy. A Necessary and Sufficient Condition for Fundamental Periods of Cascade Machines to be Products of the Fundamental Periods of their Constituent Finite State Machines. *Information Sciences*, vol. 24, pp. 71–91, 1981.
3. D. Bleichenbacher and S. Patel. SOBER Cryptanalysis. In *Fast Software Encryption - Rome'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 305–316. Springer–Verlag, 1999.
4. C. Ding, G. Xiao and W. Shan. *The Stability Theory of Stream Ciphers*. Volume 561 of *Lecture Notes in Computer Science*. Springer–Verlag, 1991.
5. J. Dj. Golić and M. Živković. On the Linear Complexity of Nonuniformly Decimated PN–Sequences. *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 1077–1079, 1988.
6. J. Dj. Golić and M. J. Mihaljević. A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenshtein Distance. *Journal of Cryptology*, vol. 3(3), pp. 201–212, 1991.
7. J. Dj. Golić and S. Petrović. A Generalized Correlation Attack with a Probabilistic Constrained Edit Distance. In *Advances in Cryptology - EUROCRYPT'92*, volume 658 of *Lecture Notes in Computer Science*, pages 472–476. Springer–Verlag, 1992.
8. J. Dj. Golić and L. O'Connor. Embedding and Probabilistic Correlation Attacks on Clock-Controlled Shift Registers. In *Advances in Cryptology - EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 230–243. Springer–Verlag, 1994.
9. J. Dj. Golić. On the Security of Nonlinear Filter Generators. In *Fast Software Encryption - Cambridge'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 173–188. Springer–Verlag, 1996.
10. J. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Trans. Inform. Theory*, IT-15:122-127, January 1969.

11. W. Meier and O. Staffelbach. Fast Correlation Attacks on Certain Stream Ciphers. *Journal of Cryptology*, vol. 1(3), pp. 159–167, 1989.
12. G. Rose. A Stream Cipher Based on Linear Feedback over GF(2^8). In *Information Security and Privacy - Brisbane '98*, volume 1438 of *Lecture Notes in Computer Science*, pages 135–146. Springer–Verlag, 1998.
13. R. Rueppel. *Analysis and design of stream ciphers.* Springer–Verlag, Berlin, 1986.
14. M. Salmasizadeh, L. Simpson, J. Dj. Golić and E. Dawson. Fast Correlation Attacks and Multiple Linear Approximations. In *Information Security and Privacy - Nepean '97*, volume 1270 of *Lecture Notes in Computer Science*, pages 228–239. Springer–Verlag, 1997.
15. P. Sarkar and S. Maitra. Nonlinearity Bounds and Constructions of Resilient Boolean Functions. In *Advances in Cryptology - CRYPTO'2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 515–532. Springer–Verlag, 2000.
16. T. Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Trans. Computers*, vol. C-34(1), pp. 81–85, 1985.
17. L. Simpson, J. Dj. Golić and E. Dawson. A Probabilistic Correlation Attack on the Shrinking Generator. In *Information Security and Privacy - Brisbane '98*, volume 1438 of *Lecture Notes in Computer Science*, pages 147–158. Springer–Verlag, 1998.
18. L. Simpson. *Divide and Conquer Attacks on Shift Register Based Stream Ciphers.* PhD thesis, Information Security Research Centre, Queensland University of Technology, Brisbane, Australia, November 1999.
19. TIA TR45.0.A, *Common Cryptographic Algorithms.* Telecommunications Industry Association, Vienna V A., USA, June 1995, Rev B.
20. D. Wagner, L. Simpson, E. Dawson, J. Kelsey, W. Millan and B. Schneier. Cryptanalysis of ORYX. In *Proceedings of the Fifth Annual Workshop on Selected Areas in Cryptography - SAC'98*, volume 1556 of *Lecture Notes in Computer Science*, pages 296–305. Springer–Verlag, 1998.
21. K. C. Zeng, C. H. Yang and T. R. N. Rao. On the Linear Consistency Test (LCT) in Cryptanalysis with Applications. In *Advances in Cryptology - CRYPTO'89*, volume 434 of *Lecture Notes in Computer Science*, pages 164–174. Springer–Verlag, 1990.
22. M. Živković. An Algorithm for the Initial State Reconstruction of the Clock-Controlled Shift Register. *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1488–1490, Sept. 1991.

APPENDIX part 1

Full Details of Example LILI with 128 bit key

The LFSRs have these feedback polynomials:
$LFSR_c$: $x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1$
$LFSR_d$: $x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1$

The two inputs x_1,x_2 to f_c are taken from $LFSR_c$ positions 12 and 20, where the range is [0,38].

The 10 inputs to f_d are taken from $LFSR_d$ positions according to this full positive difference set:

(0,1,3,7,12,20,30,44,65,80)

APPENDIX part 2

This is the truth table of the output function f_d:
0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,
0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,
0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,
1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,
0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,
0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,
0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,
1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,
0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,
0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,
0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0,
0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0,
0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,
1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,
0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,0,0,1,1,0,0,1,1,
1,1,0,0,1,1,0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0,
0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,
1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,
0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,
1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,1,1,0,0,0,0,1,1,0,0,1,1,1,1,0,0,
0,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,
1,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,
0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,
1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,
0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,
1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,
0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,
1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,
0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,
1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,
0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,
1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1,1,0

The Boolean Function has 10 inputs and these properties:
Balanced, CI(3), Order=6, Nonlinearity=480, No Linear Structures.