

The LILI-128 Keystream Generator

E. Dawson¹ A. Clark¹ J. Golić² W. Millan¹ L. Penna¹
L. Simpson¹

¹ Information Security Research Centre, Queensland University of Technology

GPO Box 2434, Brisbane Q 4001, Australia

Email {dawson,aclark,millan,penna,simpson}@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

The LILI-128 keystream generator is a LFSR based synchronous stream cipher with a 128 bit key. The design offers large period and linear complexity, and is resistant to currently known styles of attack. LILI-128 is simple to implement in hardware or software.

1 Introduction

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 [14].

However, keystream generators using regularly clocked LFSRs are susceptible to correlation attacks, including fast correlation attacks, a concept first introduced in [12]. 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 the single design principles of either a nonlinear function of regularly clocked LFSR sequences [17, 15] or on irregular clocking of LFSRs [6, 18], both of these approaches are combined for the LILI keystream generators. LILI-128 is a specific cipher from the LILI family of keystream

generators, which was first introduced in [20]. The use of both nonlinear functions and irregular clocking is not novel, having been employed in previous constructions such as ORYX [21] and SOBER [13]. Weaknesses in the design of ORYX resulted in the provision of a very low level of cryptographic security [22]. Some attacks on the SOBER proposal have also been identified [3]. Although the design for the LILI-128 keystream generator 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.

We now briefly summarise the security claims for LILI-128. Firstly, the period at around 2^{128} exceeds the length of any practical plaintext. Secondly, the linear complexity is conjectured to be at least 2^{68} , so that at least 2^{69} consecutive bits of known plaintext are required for the Berlekamp-Massey [11] attack. This is an infeasible amount of text to collect. Thirdly, we conjecture that the complexity of divide and conquer attacks on LILI are at least 2^{112} operations, requiring at least 1700 known keystream bits. This is a conservative estimate, and the true level of security may be higher. Taken together, these results indicate that LILI-128 is a secure synchronous stream cipher.

2 Description of LILI-128 Keystream Generator

The LILI-128 keystream generator is a simple and fast keystream generator that uses two binary LFSRs and two functions to generate a pseudorandom binary keystream sequence. The structure of the LILI keystream generators is 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 [14] (nonlinear filter generator).

The state of LILI-128 is defined to be the contents of the two LFSRs. The functions f_c and f_d are evaluated on the current state data, and the feedback bits are calculated. Then the LFSRs are clocked and the keystream bit is output. At initialisation, the 128 bit key is used directly to form the initial values of the two shift registers, from left to right, the first 39 bits in $LFSR_c$ then the remaining 89 bits in $LFSR_d$. In the rare event that either register is initialised as all zeroes, then that key is declared invalid. All valid keys produce a different keystream and there are no known weak keys.

The LILI-128 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. For LILI-128, $LFSR_d$ is clocked at least once and at most four times between the production of consecutive keystream bits.

2.1 Clock Control Subsystem

The clock-control subsystem of LILI-128 uses a pseudorandom binary sequence produced by a regularly clocked LFSR, $LFSR_c$, of length 39 and a function, f_c , operating on the contents

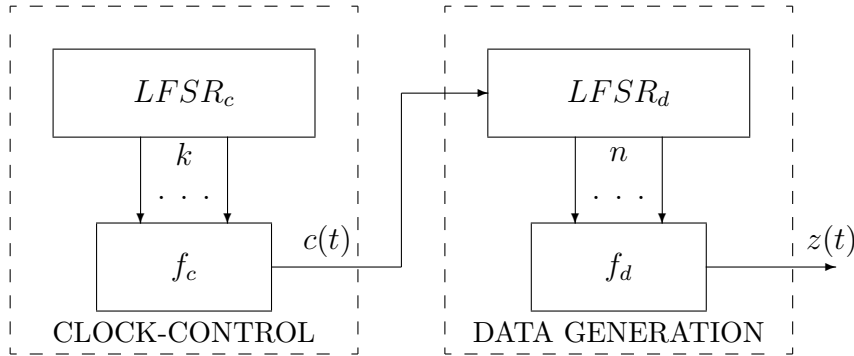


Figure 1: Overview of LILI keystream generators.

of $k = 2$ stages of $LFSR_c$ to produce a pseudorandom integer sequence, $c = \{c(t)\}_{t=1}^{\infty}$. The feedback polynomial of $LFSR_c$ is chosen to be the primitive polynomial

$$x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1$$

and the initial state of $LFSR_c$ is never allowed to be the all zero state. It follows that $LFSR_c$ produces a maximum-length sequence of period $P_c = 2^{39} - 1$.

To remove any possible ambiguity, we now present the recursion that corresponds to the feedback polynomial for $LFSR_c$. Let the stages of $LFSR_c$ be labelled $s[0], s[1], \dots, s[38]$ from left to right. Now, let the LFSR shift left. Then at time t , we have the following formula to calculate the feedback bit:

$$s[39 + t] = s[37 + t] \oplus s[25 + t] \oplus s[24 + t] \oplus s[22 + t] \oplus s[8 + t] \oplus s[6 + t] \oplus s[4 + t] \oplus s[t]$$

where \oplus indicates the exclusive-or operation on bits (equivalent to addition modulo 2).

At time instant t , the contents of stages 12 and 20 of $LFSR_c$ are input to the function f_c and the output of f_c is an integer $c(t)$, such that $c(t) \in \{1, 2, 3, 4\}$. The function f_c is given by

$$f_c(x_{12}, x_{20}) = 2(x_{12}) + x_{20} + 1.$$

This function was chosen to be a bijective mapping 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 $P_c = 2^{39} - 1$.

2.2 Data Generation Subsystem

The data-generation subsystem of LILI-128 uses the integer sequence c produced by the clock-control subsystem to control the clocking of a binary LFSR, $LFSR_d$, of length $L_d = 89$. The contents of a fixed set of $n = 10$ stages of $LFSR_d$ are input to a specially chosen Boolean function, f_d . The truth table for this function is given in the Appendix. The binary output of f_d is the keystream bit $z(t)$. After $z(t)$ is produced, The two LFSRs are clocked and the process repeated to form the keystream $z = \{z(t)\}_{t=1}^{\infty}$.

The feedback polynomial of $LFSR_d$ is chosen to be the primitive polynomial

$$x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1$$

and the initial state of $LFSR_d$ is never the all zero state. Then $LFSR_d$ produces a maximum-length sequence of period $P_d = 2^{89} - 1$, which is a Mersenne Prime.

To remove any possible ambiguity, we now present the recursion that corresponds to the feedback polynomial for $LFSR_d$. Let the stages of $LFSR_d$ be labelled $u[0], u[1], \dots, u[88]$ from left to right. Now, let the LFSR shift left. Then at time t , we have the following formula to calculate the feedback bit:

$$u[89 + t] = u[88 + t] \oplus u[50 + t] \oplus u[47 + t] \oplus u[36 + t] \oplus u[34 + t] \oplus u[9 + t] \oplus u[6 + t] \oplus u[t]$$

where \oplus indicates the exclusive-or operation on bits (equivalent to addition modulo 2).

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)(see [9]). The function f_d has been chosen to be balanced, highly nonlinear and to satisfy the third order of correlation immunity relative to the positions of 10 stages used as inputs to f_d . It was constructed following the technique recently introduced in [16]). The function f_d chosen has a nonlinearity of 480 and an algebraic order of 6. The Boolean truth table of f_d is listed in the Appendix.

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 [11] 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-128 keystream generator 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 (see [20]).

Theorem 1 *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 $f_c(0, \dots, 0) = 1$), then the period of the output sequence z is given by the product $P_z = (2^{L_c} - 1)(2^{L_d} - 1)$.*

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 1 are satisfied. Thus the period of the keystream is $P_z = (2^{39} - 1)(2^{89} - 1) \approx 2^{128}$.

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 [18, 13].

3.2 Linear Complexity

For the proposed keystream generator, the output of a nonlinear filter generator with period $P_d = 2^{89} - 1$ or a divisor of P_d is non-uniformly decimated by means of a sequence with period $P_c = 2^{39} - 1$. In [5], the following upper bound on the linear complexity of irregularly decimated maximum-length sequences is given. Let the length of the LFSR be denoted L . When a maximum-length sequence of period P_d is non-uniformly 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 $L \cdot P_c$ only if the multiplicative order of 2 modulo $P_d / \gcd(P_d, S)$ is equal to L . 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 and P_c .

For a non-uniformly decimated nonlinearly filtered LFSR sequence, the maximal attainable linear complexity is $L' \cdot P_c$, where L' is the linear complexity of the (regularly clocked) nonlinearly filtered sequence. It is known (e.g., see [14]) that L' depends on the filter function and on the positions of stages used for its inputs and that L' is very likely to be lower bounded by $\binom{L}{r}$, where r is the nonlinear algebraic order of the filter function. Accordingly, our conjecture is that the linear complexity of a non-uniformly decimated nonlinearly filtered sequence is very likely to be lower-bounded by $\binom{L}{r} \cdot P_c$. As a consequence, it is also lower-bounded by $L \cdot P_c$.

To investigate this conjecture, computer simulations were performed for other members of the LILI family of keystream generators as described in [20], with 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.

According to these results, 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 [11] attack. As the key will be changed well before even a fraction of this amount of data is generated, LILI-128 is considered to be secure from such an attack.

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 $f_d(0, \dots, 0) = 0$, 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

$k = 2$					$k = 3$				
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.

same proportion of zeroes and ones as in d . The ratio of the number of ones to the number of zeroes is given by $\frac{(2^{L_d-1})}{(2^{L_d-1}-1)}$. Note that this value approaches unity for large values of L_d , as for example in LILI-128 where $L_d = 89$.

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 4 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 $\sum_{t=1}^{P_c} c(t) = (5 * 2^{38}) - 1$ times so, on average, $LFSR_d$ is clocked $\frac{(5*2^{38})-1}{2^{39}-1}$ times per keystream symbol produced. This is approximately $\frac{5}{2}$. Thus, for large L_c , the throughput rate is approximately $\frac{2}{5}$ of the rate at which $LFSR_d$ is clocked. However a hardware implementation can use multiple copies of the feedback function to allow the irregular clocking to be performed more efficiently. We suggest that, for example, 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 $c(t)$. In hardware, the irregular clocking can then be performed in one step only. 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-128 keystream generator. These are known-plaintext attacks conducted under the standard 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-128 generator, 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 $LFSR_c$. We shall describe these attacks in relation to the general LILI keystream generator as described in [20] and show how such attacks are not feasible for LILI-128.

4.1 Attacks on Irregularly Clocked $LFSR_d$

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, (so that there is a fixed maximum number of times the data shift register may be clocked before an output bit must be produced), 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 LILI. The rest of this section describes how such an attack would be performed.

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 [18]. 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 keystream 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 super-exponential in 2^k (see [8]). For $k = 2$, the required keystream length [24] is prohibitively large.

4.2 Attacks Targeting $LFSR_c$

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.

4.2.1 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 [23]. The feasibility of such an approach depends on the number 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. For example, this attack is complicated if the tap positions are chosen according to a full positive difference set (see [9]).

4.2.2 Summary: Attacks on $LFSR_c$

The choice of parameters for the data-generation subsystem, in particular the Boolean function f_d , make attacks targeting $LFSR_c$ infeasible. In [15], 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.

4.3 Attacks on Regularly Clocked $LFSR_d$

An alternative approach would be to use a correlation attack on the nonlinear filter generator [15] 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 the number of inputs to the filter function and on the tap positions. The use of a full positive difference set for the tap positions, as suggested in [9], and of a filter function with correlation-immunity order greater than zero renders 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 bit has 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. This is especially true 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.

4.3.1 Summary: Attacks on $LFSR_d$

The length of $LFSR_d$ makes attacks targeting $LFSR_d$ 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 at least $O((2^{89} - 1)(3N^2))$, where the required length of the known keystream, N , is very likely to be very large even for $k = 2$. In [18], 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 at least $O(2^{112})$, requiring approximately 1700 bits of known plaintext.

4.4 Summary of Security Claims

In this section we summarize the claims we make for the security of LILI-128. Firstly, the period at around 2^{128} is sufficiently large. Secondly the linear complexity is conjectured to be at least 2^{68} , so that at least 2^{69} consecutive bits of known plaintext are required for the Berlekamp-Massey attack. This is an infeasible amount of text to collect. Thirdly, we conjecture that the complexity of divide and conquer attacks on LILI-128 is at least 2^{112} operations, requiring knowledge of at least 1700 known keystream bits. This is a conservative estimate, and the true level of security may be much higher. Taken together, these results indicate that LILI-128 is a secure synchronous stream cipher.

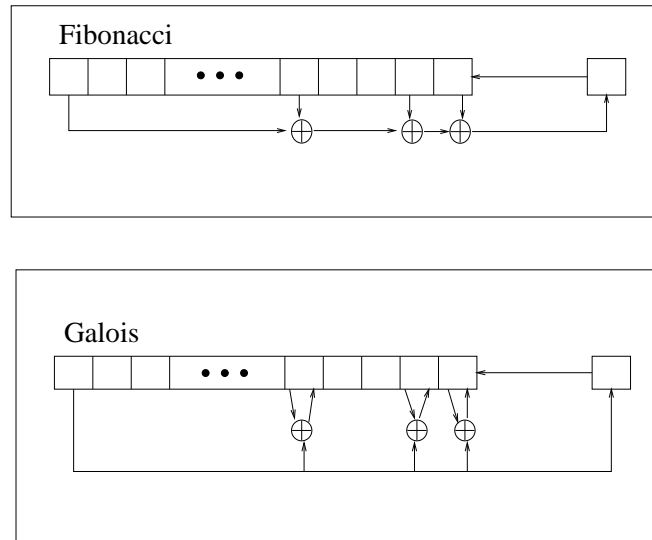


Figure 2: Fibonacci and Galois styles of LFSR implementation

5 Efficiency and Implementation

This submission includes a reference C implementation of LILI-128 that runs at 4.8 Megabits per second (Mbps) (1200 clock cycles per byte) on a 650Mhz Pentium III processor with 128MB RAM using the Microsoft Visual C++ V5.0 compiler. A current implementation of LILI-128 has been created using C and runs at 7.5 Mbps on the same processor. It is expected that an optimised implementation written in assembly language would exceed this speed.

Our reference implementation uses a Fibonacci configuration whereas our current implementation uses the Galois configuration. The Fibonacci configuration and the Galois configuration (see Figure 2) are two common methods of configuring a LFSR. The Fibonacci configuration is efficient in hardware as it only requires a single shift register N bits long and a few XOR gates, however it is inefficient in software as the individual bits must be collected by a sequence of shifts and a mask then XORed together. The Galois configuration is more efficient in software as it applies an XOR as a single operation on the register where the XOR value is the primitive polynomial [10].

By initialising a Galois or Fibonacci register using reverse processing, the same output sequence can be generated by different states although the final state of the registers will differ.

As the LILI-128 stream cipher is designed to use the state information of the register to create the keystream, our current implementation uses a Galois configuration to calculate the output bit efficiently then shifts the output bit into another word which is then used to extract tap bits.

The initial calculation of the Galois key value produces extra overhead in the total key initialisation section of the implementation, however this is at a negligible speed cost due to this only occurring once.

When using more than one processor word for an LFSR register a possible optimisation technique for a Galois configuration involves reversing the order of every second word value

and the feedback polynomial associated with that word. Other speed gains were created in our current implementation by using the word size of the processor, unsigned values where possible, and left shifting of the words was found to involve less instructions than right shifting due to the sizes of the LILI-128 registers.

In hardware, LILI-128 can be made to run very quickly by exploiting the parallelism between the two stages. To achieve the maximum throughput rate, instead of irregularly clocking the shift register a given number of steps, four copies of the feedback function can be maintained. In this fashion, the irregular clocking can then be performed in hardware in only one step. Thus there is a tradeoff between hardware space and timing regularity. Note that the use of this parallel-feedback method would provide resistance against timing attacks. Although we have no hardware simulation results, we expect an optimised LILI-128 to produce output at a rate close to that of the underlying clock.

In order to facilitate discussion, a web site devoted to LILI-128 is being developed by the authors and maintained at <http://www.isrc.qut.edu.au/lili>. It is anticipated that future developments with LILI-128 will be reported there.

6 Conclusion

In this paper, the LILI-128 keystream generator, intended for use in stream cipher applications, is proposed. The design is simple: the LILI-128 generator is based on two binary LFSRs and use two combining functions. The security of this keystream generator has been investigated with respect to currently known styles of attack. With the chosen parameters, LILI-128 provides the basic security requirements for cryptographic sequences, such as a long period and high linear complexity. Also, LILI-128 is 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.

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.

The design is transparent, relying on basic known results in LFSR theory. In addition LILI-128 is easy to implement in software or hardware and, as it employs only simple components, LILI-128 can be implemented efficiently on any platform. Finally, the designers would like to state that no weakness has been inserted into the LILI-128 design.

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 Generalised 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 Generalised 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] D. Knuth, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd Ed, Cambridge University Press, pp. 296-300, 1992.
- [11] J. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Trans. Inform. Theory*, IT-15:122-127, January 1969.
- [12] W. Meier and O. Staffelbach. Fast Correlation Attacks on Certain Stream Ciphers. *Journal of Cryptology*, vol. 1(3), pp. 159–167, 1989.
- [13] 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.
- [14] R. Rueppel. *Analysis and design of stream ciphers*. Springer–Verlag, Berlin, 1986.
- [15] 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.
- [16] 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.

- [17] T. Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Trans. Computers*, vol. C-34(1), pp. 81–85, 1985.
- [18] 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.
- [19] 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.
- [20] L. Simpson, E. Dawson, J. Dj. Golić and W. Millan. LILI Keystream Generator. *Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptology - SAC'2000* to appear in Springer-Verlag LNCS, 2000.
- [21] TIA TR45.0.A, *Common Cryptographic Algorithms*. Telecommunications Industry Association, Vienna V A., USA, June 1995, Rev B.
- [22] 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 Cryptology - SAC'98*, volume 1556 of *Lecture Notes in Computer Science*, pages 296–305. Springer–Verlag, 1998.
- [23] 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.
- [24] 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.

