On the Security of Nonlinear Filter Generators

Jovan Dj. Golić *

Information Security Research Centre, Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia
School of Electrical Engineering, University of Belgrade
Email: golic@fit.qut.edu.au

Abstract. By regarding a nonlinear filter keystream generator as a finite input memory combiner, it is observed that a recent, important attack introduced by Anderson can be viewed as a conditional correlation attack. Necessary and sufficient conditions for the output sequence to be purely random given than the input sequence is such are pointed out and a new, so-called inversion attack is introduced, which may work for larger input memory sizes in comparison with the Anderson's attack. Large input memory size and use of full positive difference sets and correlation immune nonlinear filter functions are proposed as new design criteria to ensure the security against the considered attacks.

1 Introduction

A nonlinear filter generator is a keystream generator consisting of a single linear feedback shift register (LFSR) and a nonlinear function whose inputs are taken from some shift register stages to produce the output. Such a generator realizes a nonlinear feedforward transform of a LFSR sequence. It can be used as a keystream generator itself (for various proposals, see references in [1]) or as a building block in a more complex shift register based keystream generator. For simplicity, we will assume that the LFSR is binary and that the nonlinear filter function is boolean. If the feedback polynomial is a primitive polynomial of degree r, then the LFSR sequence is a maximum-length sequence of period $2^r - 1$. For a boolean function of algebraic order k, it is very likely that the linear complexity of the keystream sequence is not smaller than about $\binom{r}{k}$ and that its period remains equal to $2^r - 1$ (for more details, see [20, 21]). So, if r is relatively large and k is not small, then the standard cryptographic criteria (large period and high linear complexity) are easily satisfied. However, good statistics, as another standard cryptographic criterion, is not readily satisfied because of the necessarily correlated, rather than independent, inputs to the filter function. This problem, which, interestingly enough, has not been addressed in the literature so far, will be among those considered in this paper, see Section 2.

* This research was supported in part by the Science Fund of Serbia, Grant #0403, through the Institute of Mathematics, Serbian Academy of Arts and Sciences.

The secret key is assumed to control the initial state of the LFSR, and may also control its feedback polynomial and/or the filter function as well. The objective of cryptanalytic attacks is to determine the unknown initial state and the structure of a nonlinear filter generator, given a long enough segment of its keystream sequence. The basic attack [23] exploits the (unconditional) correlations between the filter function output and linear functions of its inputs resulting in bitwise correlations between the keystream sequence and various linear feedforward transforms of the LFSR sequence. Such correlations always exist since the squares of the corresponding correlation coefficients sum up to one [16]. The attack can be used to reconstruct the initial state and the filter function of the same or equivalent generator that produces the same keystream sequence, given a known feedback polynomial. It is successful if the number of nondegenerate inputs to the filter function is relatively small compared with the LFSR length r, but requires an exhaustive search through all possible initial states which is infeasible for large r.

A considerable speed-up can be obtained by applying a modified fast correlation attack [3], but the computational complexity remains exponential in r because the attack employs the information set decoding rather than the iterative error-correction algorithm, both introduced in [15]. Note that standard techniques for fast correlation attacks [15] (for references on other related algorithms, see [5]) may have problems as a consequence of simultaneous bitwise correlations between the keystream sequence and various linear feedforward transforms of the LFSR sequence. Namely, since all of these transforms satisfy the same linear recurrence as the original LFSR sequence, the attacks may fail to recover the initial states yielding significant correlations. This is especially the case if the maximum correlation coefficient is not well distinguished (e.g., if the filter function is close to a bent function [16]). The attacks work better if the filter function is known. The unknown feedback polynomial can be reconstructed regardless of the initial state and the filter function by taking the linear model approach [6, 7] according to which the keystream sequence satisfies the same linear recurrence as the LFSR sequence with probability different from one half. This probability depends on the correlation coefficients of the filter function to linear functions and on the weight (number of nonzero terms) of the feedback polynomial. Finally, it may be interesting to note that the linear complexity stability approach to nonlinearly filtered LFSR sequences [2] is also related to unconditional correlations between the filter function and the linear functions.

Anderson [1] was the first to point out in the open literature that there exist other correlations in nonlinear filter generators that may be useful for improving the success of fast correlation attacks. They are based on the so-called augmented filter function and are effectively determined for a number of candidate filter functions in [1]. In order to briefly review the Anderson's attack, we now introduce some necessary notation. Let $x = (x(t))_{t=-r}^{\infty}$ be a binary maximum-length sequence of period $2^r - 1$ ($(x(t))_{t=-r}^{-1}$ is the LFSR initial state), let $f(z_1, z_2, \ldots, z_n)$ be a boolean function of n, $n \leq r$, nondegenerate input variables, and let $\gamma = (\gamma_i)_{i=1}^{n}$ be an increasing sequence of nonnegative integers such that $\gamma_1 = 0$ and $\gamma_n \leq r - 1$. Then the output sequence $y = (y(t))_{t=0}^{\infty}$ of a nonlinear filter generator is defined by

$$y(t) = f(x(t - \gamma_1), \ldots, x(t - \gamma_n)), \quad t \geq 0. \tag{1}$$

Let $y_t^m = (y(i))_{i=t-m+1}^{t}$ and $x_t^m = (x(i))_{i=t-m+1}^{t}$ denote blocks of m successive output and input bits at time t, respectively. It is clear that $y_t^m = F_m(x_t^{m+\gamma_n})$ where F_m is an m-dimensional vectorial boolean function of $m + \gamma_n$ not necessarily nondegenerate variables which depends on the boolean function f and the tapping sequence γ only.

In the examples considered in [1], it is assumed that γ is a sequence of consecutive integers ($\gamma_n = n - 1$), and output blocks of n consecutive bits are considered. The function F_n is called the augmented filter function. Hence $y_t^n = F_n(x_t^{2n-1})$. The main idea in [1] is to study the statistical dependence of the input block x_t^{2n-1} upon the known output block y_t^n by analyzing the truth table of F_n. In particular, one may consider the correlation coefficient (to the zero boolean function) of various linear functions of the input x_t^{2n-1} given a known output y_t^n. It turns out that some of these correlation coefficients can be much larger than what one could expect from the (unconditional) correlation coefficients of the filter function f, and some of them can even be equal to ± 1. These new conditional correlations can be used to dramatically improve the success of fast correlation attacks on the unknown LFSR sequence given the feedback polynomial and the filter function f with the tapping sequence γ. Note that standard fast correlation techniques directly incorporate uneven initial noise probabilities resulting from conditional correlations. Of course, the correlation coefficients equal to ± 1 can directly be used for algebraic reconstruction of the unknown LFSR initial state by solving the corresponding linear equations. Regarding the computational complexity of the basic attack, for each assumed out of $2^{2\gamma_n+1}$ possible input linear functions, the amount of computation needed to determine the conditional correlation coefficients is proportional to $2^{2\gamma_n+1}$. This also holds if γ is not a sequence of consecutive integers, but then other possibilities might be explored, see Section 3. The examples analyzed in [1] include a second-order correlation immune function of five variables, a bent and an almost bent function of six variables, and a de Bruijn function of six variables, and all of them exhibit considerable information leakage through the augmented filter function. To minimize the leakage, further investigation into the properties of boolean functions is recommended in [1]. In Section 3, however, we will show that it is the choice of the tapping sequence γ, defining the input stages to the filter function f, that is in fact more important than the choice of f itself, with respect to the conditional correlation weakness demonstrated in [1]. The proposed solution is based on positive difference sets and correlation immune boolean functions [22].

In order to shed some more light on the Anderson's attack and to make a basis for another, more efficient attack on nonlinear filter generators, we now emphasize that *every nonlinear filter generator is a finite input memory combiner* *with one input and one output.* Its memory size is clearly $M = \gamma_n - \gamma_1 = \gamma_n$ and is upper-bounded by $r - 1$. Interestingly enough, although implicit in [20], this rather fundamental and apparent fact has not been explicitly pointed out as such in the literature. We will show in Section 2 that it leads to a new, so-called inversion attack which may work for four times as large a size of the input memory as what is generally required for success of the Anderson's attack. On the other hand, it then turns out that the augmented filter function [1] can be viewed as a special case of the function introduced in [4] for a general combiner with M bits of memory which represents $M + 1$ successive output bits as a vectorial boolean function of the corresponding $M+1$ successive inputs and the preceding internal state. It is proved in [4] that it takes at most $M+1$ successive output bits to observe the statistical dependence between the output and the corresponding input. In particular, it is shown that the dependence can be studied through the (unconditional) termwise correlation between linear functions (feedforward transforms) of the input and output. The fact that linear functions of the input when conditioned on the output may have large correlation coefficients to the zero function was first demonstrated in [17] for the summation generator with two inputs (see [21]), which is a particular combiner with one bit of memory. Similar conditional correlation weakness of the stop-and-go cascade generator (see [10]) consisting of two stages was established in [18], and was later extended to an arbitrary number of stages in [19].

The proposed design criteria for nonlinear filter generators are summarized in Section 4. The proofs of mathematical results are given in the Appendix.

2 Inversion Attack

Since the output sequence of a nonlinear filter generator is produced at the same speed as the input one, the inputs to the filter function are necessarily correlated, regardless of the choice of the tapping sequence. As a consequence, even if we assume that the input sequence is purely random, that is, a sequence of balanced (uniformly distributed) and independent bits (binary random variables) and that the filter function is balanced (produces balanced output given a balanced input), the output sequence is not necessarily such. The first problem we are concerned with here is finding the conditions for the output sequence to be purely random given that the input sequence is such. This is a natural design criterion for general combiners with memory proposed in [4, 8].

Assume a probabilistic model in which the input sequence $x = (x(t))_{t=-r}^{\infty}$ is regarded as a sequence of balanced and independent bits (for simplicity, we keep the same notation for random variables and their values). Clearly, the output sequence $y = (y(t))_{t=0}^{\infty}$ is a sequence of balanced bits if and only if the filter function f is balanced. In general, y is a sequence of balanced and independent bits if and only if the vectorial boolean function F_m, associated with m successive output bits, is balanced for each $m \geq 1$. More precisely, because of the finite input memory, we have the following stronger characterization.

Lemma 1. *For a nonlinear filter generator with input memory size M, the output sequence is purely random given that the input sequence is such if and only if F_{M+1} is balanced.*

This is not yet a characterization in terms of the filter function f and the tapping sequence γ.

Theorem 2. *For a nonlinear filter generator with the filter function f and independent of the tapping sequence γ, the output sequence is purely random given that the input sequence is such if (and only if) $f(z_1, \dots, z_n)$ is balanced for each value of $(z_2, \dots, z_n)$, that is, if*

$$f(z_1, \dots, z_n) = z_1 + g(z_2, \dots, z_n), \tag{2}$$

or if $f(z_1, \dots, z_n)$ is balanced for each value of $(z_1, \dots, z_{n-1})$, that is, if

$$f(z_1, \dots, z_n) = z_n + g(z_1, \dots, z_{n-1}). \tag{3}$$

We have only proved the sufficiency of the conditions. To prove their conjectured necessity ('and only if'), a subtle underlying combinatorial problem remains to be solved. One may be tempted to guess that the linearity in any of the intermediate variables may also be sufficient, but counterexamples are easily found. Of course, if one assumes that the initial memory state $(x(t))_{t=-M}^{-1}$ is fixed rather than random, then by considering the first output bit it trivially follows that in order for y to be purely random, f must be balanced for any fixed value of $(z_2, \dots, z_n)$. However, this assumption might be considered too strong, since the initial state affects only the first M output bits.

We are now ready to formulate the *inversion attack*. The objective of the attack is to reconstruct the LFSR initial state from a segment of the keystream sequence, given the LFSR feedback polynomial of degree r, the filter function f, and the tapping sequence γ. Assume that the filter function has the form as in (2). Then (1) can be put into the following form

$$x(t) = y(t) + g(x(t-\gamma_2), \dots, x(t-\gamma_n)), \quad t \geq 0, \tag{4}$$

which means that a nonlinear filter generator as a combiner with one input and one output is invertible if the initial memory state is known. The forward inversion attack then goes as follows. The backward inversion attack is based on (3) and is essentially the same as the forward one, but works backwards in time.

1. Assume (not previously checked) M bits $(x(t))_{t=-M}^{-1}$ of the unknown initial memory state.
2. By using (4), generate a segment $(x(t))_{t=0}^{r-M-1}$ of the input sequence from a known segment $(y(t))_{t=0}^{r-M-1}$ of the keystream sequence.
3. By using the LFSR linear recursion, generate a sequence $(x(t))_{t=r-M}^{N-1}$ from the first r bits $(x(t))_{t=-M}^{r-M-1}$.
4. By using (1), compute $(\hat{y}(t))_{t=r-M}^{N-1}$ from $(x(t))_{t=r-2M}^{N-1}$ and compare with the observed $(y(t))_{t=r-M}^{N-1}$. If they are the same, then accept the assumed initial memory state and stop. Otherwise, go to step 1.

One may also compute the first part of the initial LFSR state, $(x(t))_{t=-r}^{-M-1}$, by the backward LFSR linear recursion. It takes 2^{M-1} trials on average to find a correct initial memory state. One may as well examine all 2^M initial memory states. In that case, the algorithm yields all the LFSR sequences that produce the given keystream sequence of length N. The found candidate initial states could then be examined on a longer sequence as well, which may reduce their number. If the determined LFSR sequence is not unique, then any such sequence is a satisfactory solution (equivalent LFSR initial states yielding the same keystream sequence), but for most filter functions this situation is very unlikely. More precisely, under a reasonable assumption that different LFSR initial states give rise to bitwise uncorrelated periodic keystream sequences, the expected number of false alarms for candidate initial states does not exceed 2^{-c} if the length of the keystream sequence is only $N = r + c$.

What is interesting about the inversion attack besides its simplicity is its computational complexity of the order of 2^M which is exponential in the input memory size, M, rather than the LFSR memory size, r. So, to ensure the resistance against the inversion attack M should be large and preferably close to its maximum possible value $r - 1$ which is, as a design criterion, overlooked in the literature. However, in some cases depending on the tapping sequence γ, the input memory size can effectively be reduced. For example, consider an equidistant tapping sequence $\gamma = (i\delta)_{i=0}^{n-1}$ where δ is a positive integer. Then instead of considering the input and output sequences themselves, we should consider their uniform decimations by δ which effectively reduces the input memory size δ times. More precisely, we consider the decimations of δ successive phase shifts, so that we can reconstruct the original LFSR sequence by interleaving. Note that a uniformly decimated LFSR sequence satisfies a feedback polynomial which can be determined by known algebraic techniques (e.g., see [9]). It can as well be computed by the Berlekamp-Massey algorithm [13] knowing that its degree is not bigger than the degree of the original feedback polynomial. If the original feedback polynomial is primitive, then the feedback polynomial of the decimated sequence is an irreducible polynomial of degree dividing r (in our case δ is relatively small, so that the degrees are equal).

More generally and perhaps less obviously, the same trick works if γ consists of integer multiples of the same positive integer δ. In particular, if γ consists of even integers only, then the input memory size is halved. This is an interesting trapdoor to nonlinear filter generators with the LFSR size around sixty, as is the case in many existing proposals. For example, it is suggested in [14] that r can be around sixty, that the filter functions can be derived from de Bruijn functions of the form (3), that one should not use input taps from adjacent LFSR stages, and that the input taps should be uniformly distributed over the LFSR length (the choice of equidistant taps satisfies all the requirements, but the corresponding nonlinear filter is easily inverted). It should be noted that the same effective reduction of the input memory size also applies to the Anderson's attack whose computational complexity is in general of the order of 2^{4M} if a systematic search is performed.

What if one chooses a balanced filter function f that does not satisfy the conditions from Theorem 2? This means that there exists a fraction p_+ of values of the input variables $(z_2, \ldots, z_n)$ where f is equal to zero or one (equally likely) regardless of z_1 and, similarly, a fraction p_- of values of the input variables $(z_1, \ldots, z_{n-1})$ where f is equal to zero or one (equally likely) regardless of z_n. In this case one should first find the minimum of p_+ and p_- and then accordingly apply a generalized inversion attack in the forward or backward direction. In the generalized inversion attack the objective is to find all possible input sequences of length $r - M$ given a segment of the keystream sequence of the same length, for each assumed initial memory state. The input sequence is now not unique in general. We also employ the basic feedforward equation (1) which, depending on the current memory state, may have a unique solution for $x(t)$, may have no solution for $x(t)$, or may have two solutions for $x(t)$ (both zero and one). One can in principle store all possible solutions for an input sequence in a binary tree structure of depth $r - M$. It is even conceivable that the solution may still be unique (inversion with a positive delay), especially if the conditions from Theorem 2 are not necessary and f is picked according to Lemma 1. The (generalized) inversion attack thus exploits the dependence between the input and the output sequence to the maximum possible extent. To analyze the number of solutions of a given length, it may be possible to apply the theory of random branching processes, but that is out of the scope of this paper. In any case, one should bear in mind that if the filter function does not fulfill the conjectured necessary requirements from Theorem 2, then the keystream sequence has a statistical weakness which, according to Lemma 1, takes at most $M + 1$ successive output bits to emerge. Also, for any given input memory size M, the bigger the complexity of the generalized inversion attack, the easier the statistical weakness is to detect.

Another point to be discussed is the case of nonlinear filter generators with multiple outputs produced from a single input at the same speed, as is proposed in [14] and [24]. Let k be the number of outputs and let M denote the input memory size of the associated combiner with a single input and with k outputs. Note that M is equal to the difference between the maximum and the minimum integers in all the k tapping sequences. As is clear from the information-theoretic standpoint, if $k > 1$, then a purely random binary input sequence can not yield a purely random k-dimensional binary output sequence at the same speed. So, even if individual binary output sequences are purely random, they can not be mutually independent. More precisely, it follows that no $\lfloor M/(k-1) \rfloor + 1$ successive k-dimensional output blocks can be uniformly distributed. The inversion attack can be executed on any individual output, whereas the generalized inversion attack can work on individual or combined outputs.

3 Positive Difference Sets and Correlation Immunity

The analysis from the previous section has revealed the following design criteria for nonlinear filter generators. To guarantee good statistical properties of the keystream sequence, the filter function f should satisfy the conditions from

Theorem 2. To render the inversion attack, with computational complexity of the order of 2^M, infeasible, one should choose a tapping sequence γ such that the input memory size M is large and preferably close to its maximum possible value $r-1$, where r is the LFSR length. In addition, to ensure that the uniform decimation technique can not reduce the input memory size, the greatest common divisor of the elements of γ should be equal to one, where the first element of γ is without loss of generality assumed to be equal to zero. On the other hand, when based on the vectorial boolean function F_{M+1}, associated with $M+1$ successive output bits, the Anderson's conditional correlation attack may generally require the computational complexity of the order of 2^{4M}. So, one might be tempted to conclude that the above design criteria also ensure the resistance against the conditional correlation attack. This is the case only if the number, n, of nondegenerate input variables to the filter function f is close to $M+1$ (and M is by assumption large). However, if n is much smaller than $M+1$, then one should be cautious. Namely, it is in principle possible that, depending on f and γ, a considerable information leakage may be observed on a subfunction of F_{M+1} corresponding to a relatively small subset of the output bits (not necessarily successive) so that the number of input variables to be examined is much smaller than $2M+1$. In fact, it is not even clear whether one should also consider the functions F_m for $m > M+1$ or not. Note that a very similar situation occurs if f does not satisfy the requirements from Theorem 2 so that F_{M+1} is not balanced: its subfunctions may not be balanced so that the statistical weakness becomes easier to detect. Our objective in this section is to study this problem more closely and to introduce additional design criteria regarding the choice of f and γ.

To this end, we first introduce a few more definitions. Let $\Gamma = \{\gamma_i : 1 \leq i \leq n\}$ be the set of n nonnegative integers corresponding to an increasing nonnegative integer sequence $\gamma = (\gamma_i)_{i=1}^n$, where $\gamma_1 = 0$ and $\gamma_n = M \leq r-1$, and let $\Gamma_\tau = \{\gamma_i + \tau : 0 \leq i \leq n-1\}$, $\tau \geq 0$, denote a phase shift of Γ. Let $I(\tau) = |\Gamma_\tau \cap \Gamma|$ denote the cardinality of the intersection between Γ_τ and Γ which is called the *intersection coefficient*. It follows that $I(\tau) = 0, \tau > M$. Let $I_{\max} = \max\{I(\tau) : 1 \leq \tau \leq M\}$. Further, a set Γ is called *equidistant* with distance δ if its elements are equidistant, that is, if for some positive integer δ, $\Gamma = \{\gamma_1 + i\delta : 1 \leq i \leq n\}$. A set Γ is called a *full positive difference set* if all the positive pairwise differences between its elements are distinct. These sets are used in the design of self-orthogonal convolutional codes, for example, see [11].

The value of $I(\tau)$ is the number of input variables shared in common by f and its phase shift by τ. It is then intuitively clear that the information leakage through F_{M+1} is related to the values of the intersection coefficients: roughly speaking, the bigger the intersection coefficients, the greater the leakage. Accordingly, it is desirable to minimize the maximum intersection coefficient. The basic properties of the intersection coefficients are established by the following three simple lemmas. The first lemma gives an interpretation of $I(\tau)$ in terms of the positive pairwise differences of elements of Γ, the second one shows that the total intersection coefficient is independent of Γ, and the third one specifies the minimum and maximum values of $I_{\max}$ and the necessary and sufficient conditions for them to be achieved.

Lemma 3. *For any $\tau \geq 1$, the intersection coefficient $I(\tau)$ is equal to the number of pairs of elements of Γ at distance τ.*

Lemma 4. *For any Γ, the total intersection coefficient is given by*

$$\sum_{\tau=1}^{M} I(\tau) = \frac{n(n-1)}{2}. \tag{5}$$

Lemma 5. *The maximum intersection coefficient $I_{\max}$ satisfies the bounds*

$$1 \leq I_{\max} \leq n-1 \tag{6}$$

where the maximum and the minimum are achieved if and only if Γ is an equidistant set and a full positive difference set, respectively.

Consequently, a natural choice for Γ is a full positive difference set. However, for any positive difference set Γ of n elements, the maximum difference M can not be smaller than $n(n-1)/2$. Since $M \leq r-1$, it follows that in this case n must be smaller than approximately $\sqrt{2r}$. If Γ is a full positive difference set and n is fixed, then $I_{\max}$ remains equal to one regardless of M, but large values of M are preferable with respect to the inversion attack. Full positive difference sets can be obtained by a systematic search (a sort of integer linear programming, see [12]) or from lists already available in the literature, for example, see [11], [12].

The lower bound in (6) cannot be achieved if $r \leq n(n-1)/2$, which is equivalent to n being greater than approximately $\sqrt{2r}$. The basic design principle would then be to choose Γ that minimizes $I_{\max}$ given r and n. In view of Lemma 3, this is equivalent to minimizing the maximum number of pairs of elements of Γ at the same mutual distance. Accordingly, for a positive integer λ, call Γ a λth-order positive difference set if λ is the maximum number of pairs of its elements with the same mutual difference (for $\lambda = 1$, we get a full positive difference set). A necessary condition for such a set to exist is clearly that $n(n-1)/(2\lambda) \leq M \leq r-1$, where M denotes the maximum positive difference. Since our objective is to minimize $I_{\max} = \lambda$, we first pick $\lambda = \lceil n(n-1)/(2(r-1)) \rceil$ and then find a λth-order positive difference set Γ such that $n(n-1)/(2\lambda) \leq M \leq r-1$. A λth-order positive difference set can be constructed by a systematic search in a similar way as a full positive difference set. Examining the existence of such sets is out of the scope of this paper.

To summarize, given r and n, we first find Γ so that $I_{\max}$ is minimized: the solution is a full positive difference set or a λth-order positive difference set with λ minimum possible. In the next step, we would like to choose an appropriate filter function f. It is clear already that the solution is among balanced correlation immune boolean functions [22]. Recall that a balanced mth-order correlation immune boolean function remains balanced if any subset of m input variables is fixed, and m is maximum possible. For example, if we pick a balanced λth-order correlation immune boolean function, then we obtain the pairwise independence in the keystream sequence: any two output bits constitute a balanced 2-dimensional vectorial boolean function. However, this is not exactly what we want to achieve, especially if f is chosen to satisfy the conditions from Theorem 2. Note that in this case f is balanced and mth-order correlation immune if and only if g is balanced and $(m-1)$th-order correlation immune. Our main goal is to study the statistical dependence between the input and the keystream sequence. For this purpose, we need a suitable mathematical formulation of the problem.

Let $X(t) = (x(t-\gamma_i))_{i=1}^n$ denote an ordered set of input bits (binary random variables) from the input sequence x used to produce an output bit $y(t)$, for any $t \geq 0$. Then the filter equation (1) becomes $y(t) = f(X(t))$. More generally, let $T^k = (t_i)_{i=1}^k$ denote an increasing sequence (ordered set) of k different nonnegative integers (times) and let $y(T^k) = (y(t_i))_{i=1}^k$ and $X(T^k) = \cup_{i=1}^k X(t_i)$ where $X(T^k)$ is ordered. Then $y(T^k)$ is a k-dimensional vectorial boolean function of $X(T^k)$. Further, let for any T^{k_1} and T^{k_2}, $P(X(T^{k_1})|y(T^{k_2}))$ denote the conditional probability that the random input corresponding to T^{k_1} takes a particular value $X(T^{k_1})$ given that the random output corresponding to T^{k_2} is equal to a particular value $y(T^{k_2})$ (for simplicity, we keep the same notation for random variables and their values).

Lemma 6. *Let Γ be a λth-order positive difference set and let f be a balanced mth-order correlation immune boolean function. Then for every $1 \leq k \leq \lfloor m/\lambda \rfloor + 1$ and every T^k, $y(T^k)$ is a balanced function of $X(T^k)$.*

Lemma 7. *Let Γ and f be as in Lemma 6. Then for every $k_1, k_2 \geq 1$, $2 \leq k_1 + k_2 \leq \lfloor m/\lambda \rfloor + 1$, and every disjoint T^{k_1} and T^{k_2}, $y(T^{k_2})$ is a balanced function of $X(T^{k_2})$ for any fixed $X(T^{k_1})$.*

Theorem 8. *Let Γ, f, k_1, k_2, T^{k_1}, and T^{k_2} be as in Lemma 7. Then*

$$P(X(T^{k_1})|y(T^{k_1} \cup T^{k_2})) = P(X(T^{k_1})|y(T^{k_1})) \qquad (7)$$

where $T^{k_1} \cup T^{k_2}$ denotes the ordered union of T^{k_1} and T^{k_2}. Furthermore, for every $1 \leq k \leq \lfloor m/\lambda \rfloor + 1$ and every $T^k = (t_i)_{i=1}^k$

$$P(X(T^k)|y(T^k)) = P(X(t_1)|y(t_1)) \prod_{i=2}^{k} P(X(t_i)|y(t_i), X((t_j)_{j=1}^{i-1})). \qquad (8)$$

Theorem 8 shows that the statistical dependence between the input sequence and any $\lfloor m/\lambda \rfloor + 1$ or less output bits is only due to the filter function, f, itself, not to the interaction between f and shifted versions of f. So, for the conditional correlation attack to be effective (in other words, to provide more information about the input sequence than the unconditional correlation attack based on f), one has to observe at least $\lfloor m/\lambda \rfloor + 2$ output bits and to analyze the corresponding vectorial boolean function. The complexity of the attack depends on the number of nondegenerate input variables to this function. If we take the minimum number, $k = \lfloor m/\lambda \rfloor + 2$, of output bits at positions determined by a set $T^k = (t_i)_{i=1}^k$, then the number of nondegenerate input variables to $y(T^k)$ as a vectorial boolean function of $X(T^k)$ is exactly the cardinality of the set $X(T^k)$ which is the union of individual sets $X(t_i), 1 \leq i \leq k$. The cardinality of each of these sets is exactly equal to the number, n, of nondegenarate input variables to f, and since the set Γ is a λth-order positive difference set, the cardinality of their pairwise intersections is at most λ according to Lemma 3. Then by the well-known inclusion-exclusion principle, the cardinality of their union is at least

$$K = (\lfloor m/\lambda \rfloor + 2)\left(n - \lambda \frac{\lfloor m/\lambda \rfloor + 1}{2}\right). \tag{9}$$

For each candidate linear function of the input, the complexity of the conditional correlation attack is lower-bounded by 2^K, and to check all 2^K of them the complexity is at least 2^{2K}. Of course, for large conditional correlation coefficients to appear it typically takes more than $\lfloor m/\lambda \rfloor + 2$ output bits to examine, so that the complexity is in fact bigger. If m is large enough compared with λ and n is relatively large, then K can easily be made large enough. If one chooses a full positive difference set Γ, then $\lambda = 1$ and $K = (m+2)(n-(m+1)/2)$. In particular, if $m \approx n/2$, then K is close to the minimum possible memory size for a full positive difference set. However, if Γ is an equidistant set, then $\lambda = n - 1$ and $K = n + 1$, since for a nonlinear f, $m \leq n - 2$, see [22]. It thus takes only two output bits for the conditional correlations to emerge, but for large correlations it takes more. In an example from [1] involving a second-order correlation immune boolean function, $n = 5$ and $m = 2$, so that K increases from 6 to 14 if Γ is a full positive difference rather than equidistant set. It should be noted that large conditional correlations have been found in [1] between 4 successive output bits and 9 successive input bits for an equidistant set, and we may need more than 4 output bits and 14 input bits for a full positive difference set.

It is interesting to analyze K as a function of λ if the LFSR length r and the ratio n/m are both fixed, bearing in mind that n can not be larger than approximately $\sqrt{2r\lambda}$. It turns out that K then decreases with λ, though not significantly. This means that the increase of n and m can not make up for the increase of λ. On the other hand, in order to obtain small unconditional correlation coefficients of f, larger values of n may be preferable.

Finally, similar results can be obtained for nonlinear filter generators with multiple outputs produced from a single input at the same speed (e.g., see [14] and [24]). Then, a good design criterion is using disjoint λth-order positive difference sets (the values of λ can be different for different outputs), that is, λth-order positive difference sets with mutually different positive pairwise differences (for $\lambda = 1$, see [11]). In this case, the number of input variables shared in common by phase shifts of any two output filter functions is at most one. Although the methods for finding such sets are essentially the same as for a single set, these sets are more difficult to find and the LFSR length r must be larger.

4 Conclusions

It is pointed out that every nonlinear filter keystream generator is a finite input memory combiner with one input and one output, and a recent attack introduced by Anderson is viewed as a conditional correlation attack. Necessary and sufficient conditions for a purely random input sequence to produce a purely random output sequence are established and a so-called inversion attack on nonlinear filter generators is proposed. The new attack may in general work for much larger input memory sizes in comparison with the Anderson's attack. To ensure the security against the conditional correlation attack, the use of full positive difference sets and correlation immune filter functions is investigated.

According to the performed analysis, the following design criteria for nonlinear filter generators are proposed:

- To achieve large period and high linear complexity, the LFSR length r and the algebraic order k of the filter function f should be large enough so that $\binom{r}{k}$ is much bigger than the expected keystream sequence length in applications.
- To guarantee good statistical properties, the filter function f should satisfy the conditions from Theorem 2.
- To render the inversion attack infeasible,
    - choose a tapping sequence γ such that the input memory size M is large and preferably close to its maximum possible value $r-1$
    - to ensure that the uniform decimation technique can not reduce the input memory size, the greatest common divisor of elements of γ should be equal to one (the first element of γ is assumed to be zero).
- To make the conditional correlation attack ineffective, the complexity parameter K given by (9) should be large enough. To this end,
    - the number n of nondegenerate inputs to f should be sufficiently large
    - γ should be chosen according to a full or a λth-order positive difference set, with λ as small as possible given r and n
    - the correlation immunity order m of f should be relatively large compared with λ.
- To prevent from fast correlation attacks, the nonzero correlation coefficients of f to linear functions (both unconditional and conditioned on its binary output) should be relatively small and mutually close in magnitude. To accomplish this, n should be large enough, and one can use a composition of a linear vectorial boolean function based on a linear error-correcting code with a specified minimum distance and a random balanced boolean function of less than n input variables (e.g., see [25], [27], [26]).
- The number of nonzero terms in the LFSR feedback polynomial and in any of its 'low' degree polynomial multiples should not be 'small'. This is important both for the resistance against fast correlation attacks and for reducing the linear statistical weakness [6, 7].

The design criteria are easily satisfied simultaneously.

Appendix

Proof of Lemma 1. The output sequence y is purely random if and only if for each $t \geq 0$ the output bit $y(t)$ is balanced for any fixed value of the previous output bits $(y(i))_{i=0}^{t-1}$. Since $y(t)$ depends only on the current input bit $x(t)$ and the M preceding input bits $(x(i))_{i=t-M}^{t-1}$, this is satisfied if and only if $y(0)$ is balanced and for each $1 \leq t \leq M$, $y(t)$ is balanced for any fixed value of the preceding output bits $(y(i))_{i=\max(0,t-M)}^{t-1}$, that is, if and only if F_{M+1} is balanced. □

Proof of Theorem 2. We will prove the sufficiency of the conditions, whereas the necessity seems natural but remains to be proved. In view of Lemma 1, it should be proved that F_{M+1} is balanced if either of the two conditions is satisfied. Assume that f is balanced for each value of $(z_2, \ldots, z_n)$. It then follows that f is balanced. The vectorial boolean function F_{M+1} is balanced if and only if $y(0)$ is balanced and for each $1 \leq t \leq M$, $y(t)$ is balanced for any fixed value of the preceding output bits $(y(i))_{i=\max(0,t-M)}^{t-1}$. The first output bit $y(0)$ is balanced because f is balanced. For each $1 \leq t \leq M$, $x(t)$ remains balanced when conditioned on the preceding output bits, and so does $y(t)$ due to the assumed property of f, regardless of γ. The form (2) of f is a simple consequence of the fact that the only two balanced boolean functions of a single variable are the identity and complement mappings. The sufficiency of the other condition is proved analogously, going backwards in time, starting from the last output bit of F_{M+1}. □

Proof of Lemma 4. We start from

$$| \Gamma_\tau \cap \Gamma | = \sum_{i=1}^{n} [\gamma_i \in \Gamma_\tau] \tag{10}$$

where $[\gamma_i \in \Gamma_\tau]$ is a boolean predicate evaluating to one or zero depending on whether $\gamma_i \in \Gamma_\tau$ or not, respectively. Since for each $1 \leq i \leq n$, γ_i appears in exactly $i - 1$ out of M sets Γ_τ, $1 \leq \tau \leq M$, we have

$$\sum_{\tau=1}^{M} I(\tau) = \sum_{i=1}^{n} \sum_{\tau=1}^{M} [\gamma_i \in \Gamma_\tau] = \sum_{i=1}^{n} (i-1) = \frac{n(n-1)}{2}. \tag{11}$$

□

Proof of Lemma 5. Since the element γ_1 appears only in Γ, we have that $I(\tau) \leq n - 1$, $1 \leq \tau \leq M$, and hence the upper bound in (6) follows. The maximum is clearly achieved if the values of γ in Γ are equidistant with distance δ and $\tau = \delta$. On the other hand, suppose that $I(\tau) = n - 1$ for some τ^*. Since the values γ_1 from Γ and $\gamma_n + \tau^*$ from Γ_{τ^*} are the only ones that are not in common, we then have $\{\gamma_i + \tau^* : 1 \leq i \leq n-1\} = \{\gamma_i : 2 \leq i \leq n\}$. Since $\tau^* > 0$ and the sequence γ is increasing, this is equivalent to Γ being an equidistant set with distance $\delta = \tau^*$.

The lower bound in (6) is a direct consequence of $|\Gamma_M \cap \Gamma| = 1$. The bound is achieved if and only if $I(\tau) \leq 1$, for every $1 \leq \tau \leq M$. By Lemma (3), this is true if and only if all the positive differences $\gamma_i - \gamma_j$, $1 \leq j < i \leq n$, are distinct, that is, if and only if Γ is a full positive difference set. □

Proof of Lemma 6. If $T^k = (t_i)_{i=1}^k$, then $y(T^k)$ is a balanced function of $X(T^k)$ if and only if $y(t_1)$ is a balanced function of $X(t_1)$ and for each $2 \leq j \leq k$, $y(t_j)$ is a balanced function of $X(t_j)$ for any fixed value of $y((t_i)_{i=1}^{j-1})$. The first condition is satisfied since f is balanced. The second condition is satisfied if for each $2 \leq j \leq k$, $y(t_j)$ is a balanced function of $X(t_j)$ for any fixed value of $X((t_i)_{i=1}^{j-1})$. This is itself satisfied if the number of bits (binary random variables) shared in common by $X(t_j)$ and $X((t_i)_{i=1}^{j-1})$ is not greater than m, because f is an mth-order correlation immune function. This is in turn satisfied since the number of bits shared in common by $X(t_j)$ and each of $X(t_i)$, $1 \leq i \leq j-1$, is at most λ, as a consequence of Γ being a λth-order positive difference set, and $\lambda(j-1) \leq \lambda\lfloor m/\lambda \rfloor \leq m$. □

Proof of Lemma 7. In essentially the same way as in the proof of Lemma 6, it follows that $y(T^{k_2})$ is a balanced function of $X(T^{k_2})$ for any fixed $X(T^{k_1})$ if for each $t \in T^{k_2}$, the cardinality of the intersection between $X(t)$ and the union of $X(T^{k_1})$ and $X(T^{k_2} \setminus t)$ is not greater than m. Since $t \notin T^{k_1}$, as T^{k_1} and T^{k_2} are disjoint, the rest of the proof is then similar as for Lemma 6. □

Proof of Theorem 8. Equation (7) results from the following equalities

$$\begin{aligned}
P(X(T^{k_1})|y(T^{k_1} \cup T^{k_2})) &= \frac{P(y(T^{k_1} \cup T^{k_2})|X(T^{k_1}))\, P(X(T^{k_1}))}{P(y(T^{k_1} \cup T^{k_2}))} \\
&= \frac{P(y(T^{k_1})|X(T^{k_1}))\, P(y(T^{k_2})|X(T^{k_1}))\, P(X(T^{k_1}))}{P(y(T^{k_1}))\, P(y(T^{k_2}))} \\
&= \frac{P(y(T^{k_1})|X(T^{k_1}))\, P(X(T^{k_1}))}{P(y(T^{k_1}))} \\
&= P(X(T^{k_1})|y(T^{k_1})). \qquad (12)
\end{aligned}$$

The first and the fourth equality in (12) hold by definition of conditional probability, the second equality is a consequence of $P(y(T^{k_2})|X(T^{k_1}), y(T^{k_1})) = P(y(T^{k_2})|X(T^{k_1}))$ ($y(T^{k_1})$ is a function of $X(T^{k_1})$) and $P(y(T^{k_1} \cup T^{k_2})) = P(y(T^{k_1}))P(y(T^{k_2}))$ (Lemma 6), while the third equality follows from $P(y(T^{k_2})| X(T^{k_1})) = P(y(T^{k_2}))$ (Lemma 7).

Equation (8) is obtained by applying Lemma 7 to

$$P(X(T^k)|y(T^k)) \;=\; P(X(t_1)|y(T^k)) \prod_{i=2}^{k} P(X(t_i)|y(T^k), X((t_j)_{j=1}^{i-1})) \quad (13)$$

which holds by definition of joint and conditional probabilities. □

References

1. R. J. Anderson, "Searching for the optimum correlation attack," Fast Software Encryption – Leuven '94, *Lecture Notes in Computer Science*, vol. 1008, B. Preneel ed., Springer-Verlag, pp. 137-143, 1995.
2. C. Ding, G. Xiao, and W. Shan, *The Stability Theory of Stream Ciphers. Lecture Notes in Computer Science*, vol. 561, Springer-Verlag, 1991.
3. R. Forré, "A fast correlation attack on nonlinearly feedforward filtered shift-register sequences," Advances in Cryptology – EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, J.-J. Quisquater and J. Vandewalle eds., Springer-Verlag, pp. 586-595, 1990.
4. J. Dj. Golić, "Correlation via linear sequential circuit approximation of combiners with memory," Advances in Cryptology – EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 658, R. A. Rueppel ed., Springer-Verlag, pp. 113-123, 1993.
5. J. Dj. Golić, "On the security of shift register based keystream generators," Fast Software Encryption – Cambridge '93, *Lecture Notes of Computer Science*, vol. 809, R. J. Anderson ed., Springer-Verlag, pp. 90-100, 1994.
6. J. Dj. Golić, "Intrinsic statistical weakness of keystream generators," Advances in Cryptology – ASIACRYPT '94, *Lecture Notes in Computer Science*, vol. 917, J. Pieprzyk and R. Safavi-Naini eds., Springer-Verlag, pp. 91-103, 1995.
7. J. Dj. Golić, "Linear cryptanalysis of stream ciphers," Fast Software Encryption – Leuven '94, *Lecture Notes in Computer Science*, vol. 1008, B. Preneel ed., Springer-Verlag, pp. 154-169, 1995.
8. J. Dj. Golić, "Correlation properties of a general binary combiner with memory," *Journal of Cryptology*, to appear.
9. J. Dj. Golić, "On decimation of linear recurring sequences," *The Fibonacci Quarterly*, vol. 33, pp. 407-411, Nov. 1995.
10. D. Gollmann and W. Chambers, "Clock-controlled shift registers: a review," *IEEE J. Sel. Ar. Commun.*, vol. 7(4), pp. 525-533, May 1989.
11. S. Lin and D. J. Jr. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
12. R. Lorentzen and R. Nilsen, "Application of linear programming to the optimal difference triangle set problem," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1486-1488, Sep. 1991.
13. J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122-127, Jan. 1969.
14. G. Mayhew, "A low cost, high speed encryption system and method," in *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE Computer Society Press, pp. 147-154, 1994.
15. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1(3), pp. 159-176, 1989.
16. W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," Advances in Cryptology – EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, J.-J. Quisquater and J. Vandewalle eds., Springer-Verlag, pp. 549-562, 1990.
17. W. Meier and O. Staffelbach, "Correlation properties of combiners with memory in stream ciphers," *Journal of Cryptology*, vol. 5(1), pp. 67-86, 1992.
18. R. Menicocci, "Cryptanalysis of a two-stage Gollmann cascade generator," in *Proceedings of SPRC '93*, Rome, Italy, pp. 62-69, 1993.

19. S.-J. Park, S.-J. Lee, and S.-C. Goh, "On the security of the Gollmann cascades," Advances in Cryptology – CRYPTO '95, *Lecture Notes in Computer Science*, vol. 963, D. Coppersmith ed., Springer-Verlag, pp. 148-157, 1995.
20. R. A. Rueppel, *Analysis and Design of Stream Ciphers.* Berlin: Springer-Verlag, 1986.
21. R. A. Rueppel, "Stream ciphers," in *Contemporary Cryptology: The Science of Information Integrity*, G. Simmons ed., pp. 65-134. New York: IEEE Press, 1991.
22. T. Siegenthaler, "Correlation immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Inform. Theory.*, vol. IT-30, pp. 776-780, Sep. 1984.
23. T. Siegenthaler, "Cryptanalyst's representation of nonlinearly filtered ML-sequences," Advances in Cryptology – EUROCRYPT '85, *Lecture Notes in Computer Science*, vol. 219, F. Pichler ed., Springer-Verlag, pp. 103-110, 1986.
24. B. Snow, "Multiple independent binary bit stream generator," U.S. Patent No. 5,237,615, 1993.
25. D. R. Stinson and J. L. Massey, "An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions," *Journal of Cryptology*, vol. 8(3), pp. 167-173, 1995.
26. C.-K. Wu, "Boolean functions in cryptology," Ph.D. thesis, Xidian University, China, 1993.
27. X.-M. Zhang and Y. Zheng, "On nonlinear resilient functions," Advances in Cryptology – EUROCRYPT '95, *Lecture Notes in Computer Science*, vol. 921, L. C. Guillou ed., Springer-Verlag, pp. 274-288, 1995.