Attacks Against Filter Generators Exploiting Monomial Mappings

Anne Canteaut, Yann Rotella

Attacks against Filter Generators Exploiting Monomial Mappings*

Anne Canteaut and Yann Rotella

Inria, Paris, France
Anne.Canteaut@inria.fr, Yann.Rotella@inria.fr

Abstract. Filter generators are vulnerable to several attacks which have led to well-known design criteria on the Boolean filtering function. However, Rønjom and Cid have observed that a change of the primitive root defining the LFSR leads to several equivalent generators. They usually offer different security levels since they involve filtering functions of the form $F(x^k)$ where k is coprime to $(2^n - 1)$ and n denotes the LFSR length. It is proved here that this monomial equivalence does not affect the resistance of the generator against algebraic attacks, while it usually impacts the resistance to correlation attacks. Most importantly, a more efficient attack can often be mounted by considering non-bijective monomial mappings. In this setting, a divide-and-conquer strategy applies based on a search within a multiplicative subgroup of $\mathbb{F}_{2^n}^*$. Moreover, if the LFSR length n is not a prime, a fast correlation involving a shorter LFSR can be performed.

Keywords. Stream ciphers, correlation attacks, LFSR, filter generator, nonlinear equivalence, monomials.

1 Introduction

The running-key used in a stream cipher is produced by a pseudo-random generator whose initialization is the secret key shared by the users. Linear feedback shift registers (LFSR) are building-blocks used in many keystream generators since they are appropriate to low-cost implementations, produce sequences with good statistical properties and have a simple mathematical description. While basic LFSR-based generators, like combination generators or filter generators, are not used directly as keystream generators in modern stream ciphers, they are still widely used either as a part of the generator or in modified form [13]. This situation then motivates an in-depth evaluation of the security of LFSR-based generators. Actually, several modern ciphers have been analyzed by enhanced variants of attacks, which were first dedicated to simple LFSR-based generators (e.g. [29,26,34]).

At this aim, our work investigates the security of the so-called filter generator, which consists of a single LFSR whose content is filtered by a nonlinear

* Partially supported by the French Agence Nationale de la Recherche through the BRUTUS project under Contract ANR-14-CE28-0015.

Boolean function. These generators have been extensively studied and are known to be vulnerable to several types of attacks, mainly algebraic attacks and their variants [10,9,38,17] and (fast) correlation attacks [32]. These attacks have led to the definition of design criteria, especially related to the choice of the filtering function, and they have initiated a whole line of research on the constructions of appropriate filtering functions. However, it has been observed more recently by Rønjom and Cid [36] that a simple change of the primitive characteristic polynomial of the LFSR (i.e., a change of the primitive root of the underlying finite field), may lead to an equivalent generator whose filtering function corresponds to the composition of a monomial permutation with the original filtering function, $x \mapsto F(x^k)$ for some k coprime to $(2^n - 1)$ where n is the LFSR length. This observation opens the door to new weaknesses since the main security criteria, like the nonlinearity, the degree or the algebraic immunity of the filtering function, are not invariant under this *nonlinear equivalence*. Hence, this raises many open questions about the relevance of the usual criteria, as noted by Rønjom and Cid. In this context, the objective of our paper is to answer most of these questions by evaluating the minimal security offered by all generators derived by monomial equivalence, and to further investigate the possibilities to transform the constituent LFSR by applying a monomial mapping, especially a *non-bijective* monomial mapping.

Our contributions. Our contributions are then two-fold: first, we show that, even if the degree and the algebraic-immunity of a Boolean function may highly vary within an equivalence class, the monomial equivalence defined by Rønjom and Cid has no impact on the resistance of a filter generator against algebraic attacks and their variants. The reason is that the degree and the algebraic immunity are not the relevant parameters for estimating the security of a filter generator as shown in [28,17,20]. Instead, the complexities of these attacks are determined by the linear complexity and the spectral immunity of the filtering function, which are derived from the univariate representation of the function and are therefore invariant under monomial equivalence. On the other hand, the second family of attacks, namely (fast) correlation attacks, are highly affected by monomial equivalence, implying that the associated criterion must be the generalized nonlinearity of the filtering function as defined in [41]. But we show that the non-bijective monomial mappings also play a very important role, usually much more important than monomial permutations, because the LFSR can then be transformed into an LFSR producing a sequence with smaller period τ. A divide-and-conquer attack can then be mounted exploiting this property, where the number of values to be examined decreases from $(2^n - 1)$ to τ. Moreover, if the LFSR length n is not a prime, the new LFSR involved in the attack may be shorter than the original one, leading to a much more efficient fast correlation attack.

Organization of the paper. We first introduce the monomial equivalence between filter generators as described by Rønjom and Cid [36] and show that the univariate representation of both the LFSR and the filtering function is well-suited for

analyzing its impact. Section 3 then focuses on algebraic attacks and proves that all filter generators obtained by monomial equivalence have the same behaviour with respect to this family of attacks. Section 4 then investigates correlation attacks and their variants, and shows that the situation is very different. Also, we describe a new setting for (fast) correlation attacks where non-bijective monomials are used. Two types of attacks are then presented: fast correlation involving a shorter LFSR which can be mounted when the LFSR length is not a prime, and correlation attacks based on FFT which recover $\log_2 \tau$ bits of the initial state where τ is a divisor of $(2^n - 1)$.

2 Equivalence between filtered LFSR

2.1 Filtered LFSRs

In the following, we focus on binary filtered LFSRs. The binary LFSR of length n with *characteristic polynomial*, $P(X) = X^n + \sum_{i=0}^{n-1} c_i X^i \in \mathbb{F}_2[X]$, is the finite-state automaton which produces the binary sequences $\mathbf{s} = (s_t)_{t \geq 0}$, satisfying the linear recurrence relation

$$s_{t+n} = \sum_{i=0}^{n-1} c_i s_{t+i}, \quad \forall t \geq 0 \ .$$

In this paper, we implicitly assume that the LFSRs we consider are non-singular, i.e., the constant term c_0 in the characteristic polynomial does not vanish. Otherwise the transition function of the LFSR is not bijective, leading to a possible loss of entropy of the internal state, which is clearly not suitable in cryptographic applications. Also, the characteristic polynomial is assumed to be irreducible, which guarantees that, for any nonzero initial state of the LFSR, the generated sequence cannot be produced by a shorter LFSR [42]. In other words, the *linear complexity* of any sequence generated by the LFSR from a nonzero initial state is equal to the LFSR length. A well-known property of LFSR sequences is that any sequence produced by an LFSR with an irreducible characteristic polynomial P (and a nonzero initial state) is periodic and its least period is equal to the order of P, i.e., to the smallest positive integer r for which $P(X)$ divides $X^r + 1$. Hence, the characteristic polynomials of LFSRs used in practical applications are chosen primitive. More details on the properties of LFSR sequences can be found e.g. in [19,25].

In this context, a *filter generator* (aka filtered LFSR), is a keystream generator composed of a single binary LFSR of length n whose content is filtered by a nonlinear Boolean function of n variables. More precisely, the output sequence $(s_t)_{t \geq 0}$ of the filter generator is given by

$$s_t = f(u_{t+n-1}, u_{t+n-2}, \ldots, u_t), \quad \forall t \geq 0 \ ,$$

where $(u_t)_{t \geq 0}$ denotes the sequence generated by the LFSR.

It is worth noticing that, in most practical proposals, the filtering function does not depend on all n bits of the internal state. For obvious implementation

reasons, f is usually chosen in such a way that it depends on $m < n$ variables only. It can then be equivalently described by an m-variable Boolean function f' and a decreasing sequence $(\gamma_i)_{1\leq i\leq m}$, with $1 \leq \gamma_i \leq n$, such that for any n-tuple $(x_1, \ldots, x_n)$,

$$f(x_1, \ldots, x_n) = f'(x_{\gamma_1}, \ldots, x_{\gamma_m}) \ .$$

Here, unless explicitly mentioned, the filtering function will be defined as a function of n variables, where n is the LFSR length, even if some (or most) of these variables are not involved in the evaluation of the function.

2.2 Univariate representation of filtered LFSRs

Filter generators have been extensively studied and are known to be vulnerable to several types of attacks which have led to the definition of some security criteria on the tapping sequence $(\gamma_i)_{1\leq i\leq m}$ [14] and on the Boolean filtering function (see e.g. [4] for a survey). For instance, it is well-known that f must have a high algebraic degree in order to generate a keystream sequence with a high linear complexity [39], a high algebraic-immunity in order to resist algebraic attacks [10,31] and a high nonlinearity in order to resist fast correlation attacks [32]. These design criteria on the filtering function must be considered up to some equivalence in the sense that several filtered LFSR may generate the same set of sequences. This equivalence between filtered LFSR can be simply described by defining the LFSR next-state function over the finite field with 2^n elements instead of the vector space $\mathbb{F}_2^n$.

In this field-oriented description, we will use the following classical notation. The finite field with 2^n elements is denoted by $\mathbb{F}_{2^n}$. The multiplicative order of a nonzero element α in a finite field, $\mathsf{ord}(\alpha)$, is the smallest positive integer r such that $\alpha^r = 1$. The trace function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_2$ is denoted by Tr^n, i.e.,

$$\mathsf{Tr}^n(x) = \sum_{i=0}^{n-1} x^{2^i} \ .$$

The index n will omitted if it is clear from the context.

Proposition 1 (Th. 9.2 in [30]). *Let P be an irreducible polynomial in $\mathbb{F}_2[X]$ with degree n. Let $\alpha \in \mathbb{F}_{2^n}$ be a root of P and $\{\beta_0, \ldots, \beta_{n-1}\}$ denote the dual basis of $\{1, \alpha, \ldots, \alpha^{n-1}\}$, i.e.,*

$$\mathsf{Tr}^n(\alpha^i \beta_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \ .$$

Then, the content of the LFSR with characteristic polynomial P at time $(t+1)$ is equal to its content at time t multiplied by α, where these vectors are identified with elements in the field $\mathbb{F}_{2^n}$ decomposed on the basis $\{\beta_0, \ldots, \beta_{n-1}\}$.

With the notation of the previous proposition, we consider the isomorphism φ from $\mathbb{F}_2^n$ into $\mathbb{F}_{2^n}$ defined by the basis $\{\beta_0, \ldots, \beta_{n-1}\}$. Then, the internal state at time t of the LFSR initialized by $X_0 = \varphi(u_0, \ldots, u_{n-1})$ corresponds to

$$X_t = X_0 \alpha^t$$

and the keystream bit at time t is given by

$$s_t = f \circ \varphi^{-1}(X_0\alpha^t) \ .$$

Therefore, any filter generator has an equivalent *univariate representation* defined by a root $\alpha \in \mathbb{F}_{2^n}$ of the LFSR characteristic polynomial, and a function F from $\mathbb{F}_{2^n}$ into $\mathbb{F}_2$. This generator produces from any initial state $X_0 \in \mathbb{F}_{2^n}$ the sequence $s_t = F(X_0\alpha^t)$. For the sake of clarity, univariate functions defined over $\mathbb{F}_{2^n}$ will be denoted by capital letters, while small letters will be used for multivariate functions over $\mathbb{F}_2^n$. Clearly, the multivariate representation of a filter generator, (P, f), can be recovered from its univariate representation (α, F): since P is irreducible, it corresponds to the minimal polynomial of α and f is equal to $F \circ \varphi$ where φ is the isomorphism associated to the dual basis of $\{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$. Conversely, a given multivariate representation (P, f) corresponds to n univariate representations (α, F) since there are several possible values for α corresponding to the conjugate roots of P, i.e., $\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}$. The univariate filtering functions F associated to the different choices for α are then linearly equivalent because they only differ from the composition with the Frobenius map. However, composing F with a linear permutation does not change its cryptographic properties (see the next section for details).

As a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^n}$, F can be written as a univariate polynomial in $\mathbb{F}_{2^n}[X]$ and the coefficients of this polynomial are computed from the values of F by the discrete Fourier Transform (DFT) of F (aka Mattson-Solomon transform) (see e.g. [2,27,15]).

Proposition 2 (Discrete Fourier transform of a function). *Let F be a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^n}$. Then, there exists a unique univariate polynomial in $\mathbb{F}_{2^n}[X]/(X^{2^n} + X)$ such that*

$$F(X) = \sum_{i=0}^{2^n-1} A_i X^i \ .$$

Moreover, $A_0 = F(0)$, $A_{2^n-1} = \sum_{x \in \mathbb{F}_{2^n}} F(x)$ and the coefficients A_i, $1 \leq i \leq 2^n - 2$, are given by the discrete Fourier transform of the values of F at all nonzero inputs, namely

$$A_i = \sum_{k=0}^{2^n-2} F(\gamma^k)\gamma^{-ki}, \ 1 \leq i \leq 2^n - 2$$

where γ is a primitive element in $\mathbb{F}_{2^n}$.

It is worth noticing that, in our context, the value of $F(0)$ does not affect the security of the filter generator: this value is only involved when the LFSR internal state vanishes, which is obviously always avoided since the sequence generated from the all-zero state is constant. Therefore, we will always consider in the following that the coefficient of degree $(2^n - 1)$ in the univariate form of F is equal to zero. In other words, the univariate form of F is identified with

$(A_0, \ldots, A_{2^n-2})$ which is the DFT of the values of F. In our situation also, F takes its values in $\mathbb{F}_2$, implying that $A_{2i} = A_i^2$ for any $1 \leq i \leq 2^n - 2$. In this case, the coefficients A_i for all i in the same cyclotomic coset modulo $(2^n - 1)$, $\mathcal{C}(i) = \{i, 2i \bmod (2^n - 1), 2^2 i \bmod (2^n - 1), \ldots, 2^{n-1} i \bmod (2^n - 1)\}$ can be gathered, leading to the so-called trace representation:

$$F(X) = \sum_{k \in \Gamma} \mathsf{Tr}^{n_k}(A_k X^k),$$

where Γ is a set of representatives of all cyclotomic cosets modulo $(2^n - 1)$, n_k denotes the size of the cyclotomic coset of k and $A_k \in \mathbb{F}_{2^{n_k}}$.

2.3 Monomial equivalence between filtered LFSR

Using the univariate representation, it is easy to observe that, for any nonzero $\lambda \in \mathbb{F}_{2^n}$, the sequence generated by the filtered LFSR with characteristic polynomial P and filtering function F from the initial state $X_0 \in \mathbb{F}_{2^n}$ is the same as the sequence obtained by filtering the same LFSR with $G(x) = F(\lambda x)$ from the initial state $Y_0 = \lambda^{-1} X_0$. It follows that not only F but also any function $G(x) = F(\lambda x)$ can be attacked when cryptanalyzing the generator. But, this equivalence does not affect the security of filter generators since all design criteria are known to be invariant under linear equivalence, i.e., under the composition of the filtering function by an $\mathbb{F}_2$-linear permutation of $\mathbb{F}_{2^n}$.

However, Rønjom and Cid [36] exhibited some nonlinear equivalence relations between filtered LFSR when the LFSR characteristic polynomial P is primitive. This equivalence relation boils down to changing the primitive root of $\mathbb{F}_{2^n}$ in the univariate representation of the generator. Let us consider two primitive elements in $\mathbb{F}_{2^n}$, namely α and β, implying that $\beta = \alpha^k$ for some integer k with $\gcd(k, 2^n - 1) = 1$. Let P_α and P_β denote their minimal polynomials. Then, we observe that, at any time $t \geq 0$, the internal state X_t of the LFSR with characteristic polynomial P_α and the internal state Y_t of the LFSR with characteristic polynomial P_β initialized with $Y_0 = X_0^k$ satisfy

$$Y_t = Y_0 \beta^t = \left(X_0 \alpha^t\right)^k = X_t^k \, .$$

This implies that the set of all sequences obtained by filtering by F the LFSR defined by α corresponds to the sequences generated by filtering by $G(x) = F(x^r)$ the LFSR defined by $\beta = \alpha^k$ where $rk \equiv 1 \bmod (2^n - 1)$. From now on, this equivalence between filter generators will be named *monomial equivalence*[1]. It follows that there exist $\frac{\Phi(2^n-1)}{n}$ monomial transformations which are not linearly equivalent and nevertheless provide equivalent filtering LFSR, where Φ is the Euler's totient function. Any attack against one among these $\frac{\Phi(2^n-1)}{n}$ generators then provides an attack against the whole class. Most notably, an initial-state recovery attack against the generator defined by β enables the attacker to recover

[1] Note that, among all monomials, only the *permutations* of $\mathbb{F}_{2^n}$, i.e., $X \mapsto X^k$ with $\gcd(k, 2^n - 1) = 1$ provide an equivalence relation.

the initial state X_0 of the LFSR defined by α by using that $X_0 = Y_0^r$. Therefore, the security level offered by a filter generator is clearly the minimal security among all generators in its equivalence class.

3 Monomial equivalence and algebraic attacks

Determining the cryptographic properties of a Boolean function up to any change of the primitive element seems rather complicated, since the major properties of the function, like its degree or its nonlinearity, are not invariant under these non-linear transformations (see e.g. [36, Appendix A]). However, the recent works by Gong, Helleseth and Rønjom [38,37,20,17] point out that this difficulty mainly comes from the fact that the multivariate representation of the function is usually not relevant for evaluating its security level. Instead, the univariate representation provides a much more powerful tool which allows to directly determine the security offered by a generator against algebraic attacks (and its variants). Indeed, the action of the monomial equivalence can be described in a much simpler way when the univariate expression of the function is considered: the class of all filtering functions in the equivalence class of F consists of all functions $G = \sum_{i=0}^{2^n-2} B_i X^i$ whose univariate representation $(B_0, \ldots, B_{2^n-2})$ is obtained by decimating the univariate representation of F by some integer k coprime to $(2^n - 1)$, i.e., $B_i = A_{ik \bmod (2^n-1)}$. Using this simple transformation, it becomes possible to determine how the complexity of algebraic-type attacks varies within the equivalence class of a filtering function.

3.1 Linear complexity

The simplest algebraic attack consists in writing the Boolean equations defining the successive keystream bits. We then obtain a multivariate system depending on n binary unknowns, which are the bits of the initial state. The degree of each equation is equal to the degree of the filtering function f, which tends to show that the complexity for solving this algebraic system highly depends on the degree of f. Instead of linearizing the system of degree $\deg(f)$ derived from f, another strategy consists in exploiting the fact that the keystream sequence produced by a filter generator can also be seen as the output of a single LFSR. The length of the shortest LFSR generating the sequence is its *linear complexity* Λ. It determines the complexity of solving the smallest linear system expressing each output bit of the generator as a linear function of its initial state. It is widely believed that, exactly as for the combination generator, the linear complexity of a filter generator increases with the degree of the filtering function (see e.g. [24,39]). For instance, it has been shown by Rueppel that, when the LFSR length n is a large prime, $\Lambda \geq \binom{n}{d}$ for most functions f of degree d [39, Chapter 5]. However, as explained in [28], the well-known Blahut's theorem [2] implies that Λ is entirely determined by the univariate form of the filtering function, $F(X) = \sum_{i=0}^{2^n-2} A_i X^i$:

$$\Lambda = \#\{0 \leq i \leq 2^n - 2 \ : A_i \neq 0\} \ .$$

Then, it clearly appears from this formula that the linear complexity of the filter generator is invariant under monomial equivalence since decimating the vector $(A_0, \ldots, A_{2^n-2})$ by some k coprime to $(2^n - 1)$ does not modify the number of its nonzero terms.

A major observation due to Rønjom and Helleseth [38] is that the linear complexity is always smaller than or equal to the number of unknowns we expect in a linearized version of the system of equations derived from the multivariate representation. Indeed, the resulting linear system considers as unknowns all monomials of degree at most $\deg(f)$ in the bits of the initial state, i.e. roughly

$$\Lambda = \Lambda(F) \triangleq \sum_{i=1}^{\deg f} \binom{n}{i} \text{ unknowns.}$$

Using that the multivariate degree of the univariate monomial X^k is the number of ones in the binary representation of k, which is identified with $\mathsf{w}_H(k)$, we get that all coefficients A_k with $\mathsf{w}_H(k) > \deg f$ vanish. Therefore, the linear complexity Λ of the generator, i.e., the number of nonzero A_k, is at most the number of k such that $\mathsf{w}_H(k) \leq \deg(f)$, which corresponds to the number of unknowns in the multivariate linear system. Therefore, for any filter generator obtained by monomial equivalence, the best basic algebraic attack has data complexity $\mathcal{O}(\Lambda)$. The on-line step of the attack has time complexity $\mathcal{O}(\Lambda)$ (since the knowledge of Λ keystream bits determines the initial state of the equivalent LFSR and the whole output sequence). The precomputation step consists in computing the linear complexity and the minimal polynomial of the keystream. This can be done by applying Berlekamp-Massey algorithm to the filter generator initialized by any chosen value, with time complexity $\mathcal{O}(\Lambda^2)$. This can also be done by inverting a $\Lambda \times \Lambda$ Vandermonde matrix, with time complexity $\mathcal{O}(\Lambda \log^2 \Lambda)$ as noticed in [38,17,35]. Another equivalent point of view, which yields the same complexity, is the so-called selective discrete Fourier spectra attack [16,17]. The complexities of all variants of this attack are then invariant under monomial equivalence.

3.2 Algebraic attacks

The fact that algebraic attacks can be applied to any generator obtained by monomial equivalence has led Rønjom and Cid to define the *general algebraic immunity* of a filtering function F [36, Def. 6] as the smallest algebraic immunity for a function in the monomial equivalence class of F. But, exactly as algebraic attacks allow to decrease the degree of the equations below the degree of the filtering function by considering an annihilator g of f [10], the same idea can be used for improving the previously described attack based on the univariate approach [17]. Then, the complexity of the best attack is determined by the smallest linear complexity for an annihilator of F. This quantity has been named the *spectral immunity* of F [17, Def. 1]. As we discussed before, for any function

G, including any annihilator of F,

$$\Lambda(G) \leq \sum_{i=0}^{\deg G} \binom{n}{i},$$

implying that this attack based on the univariate approach is always faster than the usual algebraic attack.

Suppose now that the previously described attack is applied to some equivalent filter generator involving the filtering function F' defined as $F'(x) = F(x^k)$, for some k with $\gcd(k, 2^n - 1) = 1$. The attack then exploits the linear complexity of an annihilator G' of F'. But, it can be observed that a function G' is an annihilator of F' if and only if $G(x) = G'(x^r)$ is an annihilator of F where $rk \equiv 1 \bmod (2^n - 1)$. Then, the linear complexity of G' is then equal to the linear complexity of G, the corresponding annihilator of F. It follows that the attack applied to F' has the same complexity as the attack against the original filter generator. In other words, the spectral immunity of a filtering function F is invariant under monomial equivalence.

Therefore, it appears that the monomial equivalence does not affect the complexity of algebraic attacks since the optimal versions of these attacks are based on the univariate representation and involve the number of nonzero coefficients in this representation which is invariant under monomial equivalence.

4 Univariate correlation attacks

4.1 Correlation-like attacks on filtered LFSR

Another type of attacks against LFSR-based stream ciphers is the correlation attack and its variants. For generators using many LFSR combined by a Boolean function, a divide-and-conquer technique can be used by exploiting an approximation of the combining function f by a function g with fewer variables [40]. The attack then consists in performing an exhaustive search for the internal state of the small generator (called the target generator) composed of fewer LFSR combined by g, and in deciding which one of the states gives an output sequence having the expected correlation with the keystream. A well-known improved variant, named *fast correlation attack* [32] applies when g is linear. It identifies the problem with a decoding problem. Then an exhaustive search for the initial state of the target generator is not required anymore. Instead, a decoding algorithm for a linear code is used, for instance an algorithm exploiting sparse parity-check relations [32,6,8]. In the case of filtered LFSR, the situation is different since the only relevant target generator producing sequences correlated to the keystream, consists of an LFSR of the same size as the original generator filtered by a linear approximation of f. In this situation, the classical correlation attack cannot be faster than a brute-force attack, implying that only fast correlation attacks are relevant on filtered LFSR. To avoid these attacks, filtering functions must have a high nonlinearity.

Rønjom and Cid [36, Section 6.2] have then pointed out that the monomial equivalence requires extending the nonlinearity criterion. As the nonlinearity of a Boolean function f is the distance of f to all affine functions, the distance to all monomial functions with an exponent coprime to (2^n-1) must also be taken into account. Indeed, the fast correlation attack can be generalized as follows. Let us consider an LFSR of size n, of primitive root α and of initial state X_0, filtered by a Boolean function F. We suppose now that there exist $\lambda \in \mathbb{F}_{2^n}\backslash\{0\}$ and k coprime to (2^n-1) such that the function F is highly correlated to $G(x) = \mathsf{Tr}^n(\lambda x^k)$. Because k is coprime to (2^n-1), the monomial equivalence can be applied to the LFSR filtered by G, as depicted on Figure 1. Then we

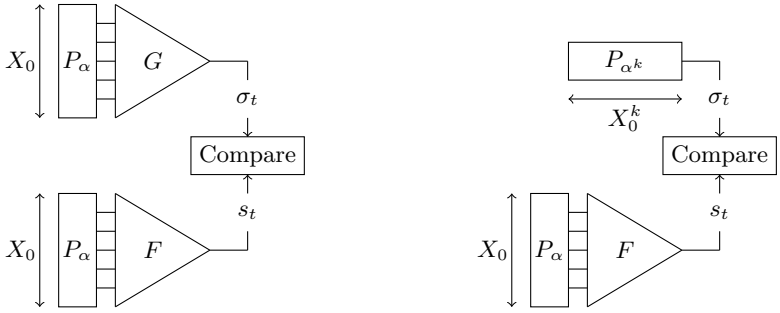

Fig. 1: Generalized correlation attack where $\gcd(k, 2^n-1) = 1$ and $G(x) = \mathsf{Tr}^n(\lambda x^k)$.

can perform a fast correlation attack and recover the initial state of the LFSR defined by α^k, which corresponds to X_0^k. As k is coprime to (2^n-1), we then recover X_0. In other words, a fast correlation attack can be mounted even if the approximation G of F is nonlinear but has a trace representation with a single term, $\mathsf{Tr}^n(\lambda x^k)$ with $\gcd(k, 2^n-1) = 1$. The corresponding design criterion is that the filtering function F must have a high generalized nonlinearity. This notion has been first introduced by Youssef and Gong in 2001 [41], but was not motivated by any attack.

Definition 1 (Extended Walsh-transform [41]). *Let F a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_2$, then its extended Walsh transform is*

$$\widehat{F}(\lambda, k) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{F(x)+\mathsf{Tr}(\lambda x^k)}$$

where $\lambda \in \mathbb{F}_{2^n}$ and $\gcd(k, 2^n-1) = 1$. Then, the generalized nonlinearity*:*

$$\mathsf{NLG}(F) = 2^{n-1} - \frac{1}{2} \max_{\substack{\lambda \in \mathbb{F}_{2^n} \\ k:\gcd(k,2^n-1)=1}} |\widehat{F}(\lambda, k)|$$

is the distance of F to the components of all monomial permutations *of $\mathbb{F}_{2^n}$.*

4.2 A more efficient correlation attack

The previously described attack applies when F is correlated with a monomial function whose exponent k is coprime to (2^n-1). However, the exponents k with $\gcd(k, 2^n-1) > 1$ must also be taken into account even if they do not provide an equivalence relation. Let us now consider some k which is not coprime to (2^n-1) and some Boolean function H such that F is correlated to $G : x \mapsto H(x^k)$. We can then also apply some monomial transformation to the target generator which is composed of the LFSR defined by α filtered by G. Indeed, the LFSR

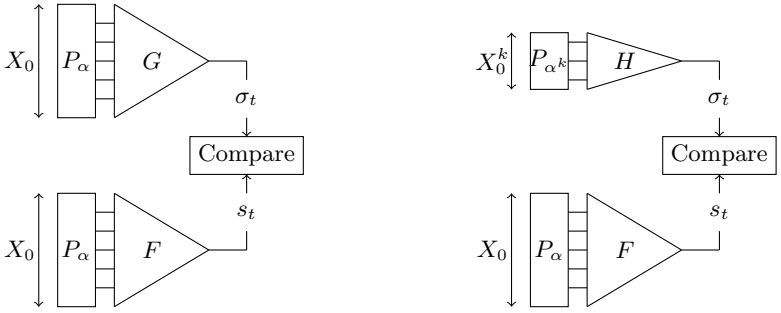

Fig. 2: Generalized correlation attack where $\gcd(k, 2^n - 1) > 1$.

internal state at time t is $X_0\alpha^t$, implying that the sequence produced by the target generator is $\sigma_t = G(X_0\alpha^t) = H(X_0^k\alpha^{kt})$ for all $t \geq 0$. On the other hand, the LFSR with characteristic polynomial P_{α^k} generates the successive internal states $(Y_0\alpha^{kt})_{t\geq 0}$, implying that σ can also be generated by the LFSR defined by α^k filtered by H. In other words, the two generators produce exactly the same sequence if the initial state of the LFSR defined by α^k satisfies $Y_0 = X_0^k$, as depicted on Figure 2. It is important to notice that the least period of the sequence generated by the LFSR defined by α^k is

$$\tau_k = \mathsf{ord}(\alpha^k) = \frac{2^n - 1}{\gcd(k, 2^n - 1)} \, .$$

We will see that this quantity plays a major role in the attack.

Firstly, the number of possible values for an initial state of the target LFSR of the form $Y_0 = X_0^k$ is τ_k. As previously mentioned, the classical correlation attack described by Siegenthaler is not relevant against filter generators because it requires an exhaustive search over all possible initial states of the constituent LFSR, leading to a time complexity higher than or equal to the cost of a brute-force attack. But, in our new setting, the attacker needs to perform an exhaustive search over a set of size $\tau_k < 2^n$, implying that this exhaustive search may be faster than the brute-force attack. More precisely, the data complexity required for applying the optimal hypothesis test (i.e., defined by the Neyman-Pearson

lemma) and determining the correct initialization out of τ_k possibilities is

$$N = \frac{2\ln(\tau_k)}{\varepsilon^2}$$

where ε is the correlation between F and G (see e.g. [18, Section 4.1]). The time complexity of Siegenthaler's algorithm is

$$\mathsf{Time} = O\left(\frac{\tau_k \ln(\tau_k)}{\varepsilon^2}\right) .$$

The counter-part of this attack compared to the case where k is coprime to $(2^n - 1)$ is that the knowledge of the quantity recovered in the attack, X_0^k, does not enable us to determine the whole initial state X_0 since k is no longer coprime to $(2^n - 1)$. However, we get some information on X_0.

Lemma 1. *The knowledge of X_0^k gives $\log_2(\tau_k)$ bits of information on X_0 where $\tau_k = (2^n - 1)/\gcd(k, 2^n - 1)$.*

Proof. Let X_0 be a non-zero element in the field $\mathbb{F}_{2^n}$ and α a primitive root. There is a unique $i \in [0, 2^n - 2]$ such that $X_0 = \alpha^i$. Then, $r = i \bmod \tau_k$ satisfies

$$X_0^k = \alpha^{qk\tau_k}\alpha^{rk} = \alpha^{rk}$$

by definition of τ_k. Moreover, r is the unique integer in $[0, \tau_k - 1]$ such that $X_0^k = \alpha^{rk}$. Indeed, if there exist r_1 and r_2, $r_1 > r_2$ such that $\alpha^{r_1 k} = \alpha^{r_2 k}$ then $\alpha^{(r_1 - r_2)k} = 1$. Then, $(r_1 - r_2)$ is a multiple of τ_k which is the order of α^k. This is impossible since $r_2 - r_1 \in [0, \tau_k - 1]$. Therefore, for $X_0 = \alpha^i$, the knowledge of X_0^k gives the value of the remainder of the Euclidean division of i by τ_k. It then provides $\log_2(\tau_k)$ bits of information on X_0. □

4.3 Recovering the remaining bits of the initial state

Once X_0^k has been recovered, the remaining $(n - \log(\tau_k))$ bits of X_0 can be found by an exhaustive search with time complexity proportional to

$$\frac{2^n - 1}{\tau_k} = \gcd(k, 2^n - 1) .$$

Another method consists in combining several correlation attacks in a divide-and-conquer approach, exactly as against combination generators. Suppose that there exist two integers k_1 and k_2 such that the two distinct correlation attacks can be performed in order to successively recover $X_0^{k_1}$ and $X_0^{k_2}$. This means that we have found

$$r_1 = i \mod \tau_{k_1} \text{ and } r_2 = i \mod \tau_{k_2} .$$

By the Chinese remainder theorem, this leads to the value of the remainder of the Euclidean division of i by $\mathsf{lcm}(\tau_{k_1}, \tau_{k_2})$. The best situation for the attacker is obviously the case where τ_{k_1} and τ_{k_2} are coprime, otherwise there is some redundancy between the information retrieved by the two distinct attacks.

4.4 Fast correlation attack when H is linear

In the correlation attack, the target generator is composed of the LFSR defined by α^k filtered by a Boolean function H, and it generates sequences σ with period $\tau_k < (2^n - 1)$. Then, as noticed in the pioneer work by Meier and Staffelbach [32], any N-bit portion of σ can be seen as a codeword in a code of length N and size τ_k. Therefore, recovering the initial state of the target generator boils down to decoding the corresponding n-bit keystream with respect to this code since the keystream can be identified with the result of the transmission of σ through a binary symmetric channel with error-probability $\frac{1}{2}(1-\varepsilon)$ where ε is the correlation between the two sequences.

In the specific case where the function H defining $G(x) = H(x^k)$ is linear, i.e., $H(x) = \mathsf{Tr}(\lambda x)$ for some $\lambda \in \mathbb{F}_{2^n}$, the involved code is a linear code. Some decoding algorithms dedicated to linear codes can then be used. These algorithms are faster than the exhaustive search (which corresponds to a maximum-likelihood decoding), at the price of a higher data complexity. The corresponding attack is then named *fast correlation attack* [32]. Obviously, a major parameter affecting the complexity of the decoding procedure is the dimension of the involved code. This dimension is the degree of the minimal polynomial of α^k, which may be smaller than n: it corresponds to the size n_k of the cyclotomic class of k. Equivalently, n_k is the smallest integer m such that $2^m \equiv 1 \bmod \tau_k$. In other words, if α^k belongs to a subfield $\mathbb{F}_{2^m}$ of $\mathbb{F}_{2^n}$, then the fast correlation attack consists in decoding a linear code of dimension m, instead of a code of dimension n. This may enable the attacker to recover $\log_2(\tau_k)$ bits of the initial state with a lower complexity than the fast correlation attack involving the original LFSR of length n. The optimal situation which maximizes the number of bits recovered by the attacker for a given complexity is then when $\tau_k = 2^m - 1$ for some divisor m of n, i.e., when k is such that $\gcd(k, 2^n - 1) = (2^n - 1)/(2^m - 1)$. Several decoding algorithms have been proposed in this context [32,21,6,7,22,33,8] which offer different trade-offs between the dimension of the code and the error probability (see [1] for a recent survey).

Example 1. Let us consider an LFSR of size 10 with primitive characteristic polynomial $P(X) = X^{10} + X^9 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + 1$. We then use as a filtering function a balanced function of 10 variables with a high nonlinearity obtained by Dobbertin's construction [12]. As described by Dobbertin, we start from a bent function which is constant on a subspace of dimension $\frac{n}{2}$ and replace this constant restriction by a balanced function in order to make the whole function balanced. Here we start from $\mathsf{Tr}(\alpha x^{33})$ where α is a root of P since this function is bent, and modify it as in [12]. It is worth noticing that this modification makes the function much more complex. In particular, it increases its degree and its linear complexity, at the price of a very small degradation of its nonlinearity. We construct this way a balanced function F of 10 variables with nonlinearity 481 and algebraic immunity 3. By computing its univariate representation, we get that the linear complexity of the keystream is equal to 992. Therefore, this filtering function meets all design criteria related to algebraic-like attacks and to fast correlation attacks. However, by construction, our filtered

function F is very close to the Boolean function $G(x) = \mathsf{Tr}(\alpha x^{33})$. This means that the keystream is highly correlated to the output of the LFSR defined by α^{33}. Indeed, the correlation between the two sequences equals $\varepsilon = 1 - 2^{-9}\mathsf{d}_H(F, G) = 0.96$. We can mount a fast correlation attack on an LFSR of size 5, and we recover almost 5 bits of the internal state of the generator. This attack is obviously much faster than the usual fast correlation attack: in our new setting, the involved correlation is $\varepsilon = 0.96$ and the code dimension is $n_{33} = 5$, while the usual fast correlation attack corresponds to a correlation $\varepsilon' = 1 - 481 \times 2^{-9} = 0.06$ and code dimension $n = 10$. The remaining 5 bits of the initial state can be determined by an exhaustive search over 33 possible values.

The previous example was rather specific since the filtering function is designed from a component of a monomial mapping x^k with k of the form $k = \frac{(2^n-1)}{(2^m-1)}$. However, a similar situation may happen for many other filtering functions which do not have any such specific structure. In order to quantify the advantage of this new setting, we first need a closer look at the complexity of fast correlation attacks. The decoding algorithms used in this context include some methods exploiting the existence of low-weight parity-check relations for the LFSR sequence [32,21,6,8]. These relations are derived from sparse multiples of the LFSR characteristic polynomial, implying that the data complexity which corresponds the degree of these multiples grows very fast with the LFSR length (unless the LFSR characteristic polynomial is very sparse). Once these relations have been found in a precomputation step, the attack consists in applying an iterative decoding algorithm. For instance, the complexity of the original attack based on parity-check relations with 3 terms is estimated by [6]:

$$\mathsf{Data} = \mathcal{O}\left(\frac{1}{\varepsilon} \times 2^{\frac{n}{2}}\right) \text{ and } \mathsf{Time} = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^3 \times 2^{\frac{n}{2}}\right) .$$

Using parity-check relations with a higher weight w decreases the influence of the LFSR length by replacing $2^{n/2}$ by $2^{n/(w-1)}$, at the price of a higher influence of the correlation, i.e., in the data complexity ε is replaced by $\varepsilon^{2(w-2)/(w-1)}$. The time complexity can be improved by different techniques, but the data complexity of most of these algorithms has a similar behaviour.

Example 2. Let us consider the same LFSR of size 10 as in Example 1, but now filtered by a Boolean function which is not constructed from a monomial function. We choose as a filtering function the following function of 6 variables:
$f(x_0, x_1, x_2, x_3, x_4, x_5) = x_0x_1x_2x_3x_4 + x_0x_1x_2x_3x_5 + x_0x_1x_2x_4x_5 + x_0x_1x_2x_4 + x_0x_1x_2 + x_0x_1x_3x_4 + x_0x_1x_3 + x_0x_1x_4 + x_0x_1x_5 + x_0x_1 + x_0x_2x_3x_4 + x_0x_2x_3x_5 + x_0x_2x_4x_5 + x_0x_2x_4 + x_0x_2 + x_0x_3x_4 + x_0x_4 + x_0 + x_1x_2x_3x_4x_5 + x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_3 + x_1x_2x_4 + x_1x_2 + x_1x_3x_5 + x_1x_3 + x_1x_4 + x_1x_5 + x_1 + x_2x_3x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_3 + x_2 + x_3x_4 + x_4x_5 + x_4$
and the inputs of f are given by the following tapping sequence $(\gamma_1, \ldots, \gamma_6) = (9, 8, 6, 3, 1, 0)$. The corresponding 10-variable function has nonlinearity 352, algebraic immunity 3 and the linear complexity of the generated keystream is 637.

But there exists a function G of the form $G(x) = \mathsf{Tr}(\lambda x^{33})$ at distance 456 from F. The correlation between the keystream and the output of a non-filtered LFSR of size $n_{33} = 5$ is then equal to $\varepsilon = 0.11$. A fast correlation attack in this setting appears to be more efficient than the usual fast correlation attack, which has parameters $n = 10$ and $\varepsilon' = 0.31$. For instance, if the iterative algorithm with parity-check relations of weight 3 is used, the ratio between the data complexities of the two attacks is given by

$$\frac{\mathsf{Data}}{\mathsf{Data}'} = \left(\frac{\varepsilon'}{\varepsilon}\right) \times 2^{\frac{n_{33}-n}{2}} = 0.498 \; .$$

4.5 Correlation attack using a Fast Fourier Transform when H is nonlinear

In the general case, i.e., when H is nonlinear, the correlation attack, as originally described in [40] corresponds to an exhaustive search over all initial states of the target generator of the form $Y_0 = X_0^k$. For each of these Y_0, the first N bits of the corresponding output sequence σ are generated and the correlation between σ and the keystream is computed, namely

$$\sum_{t=0}^{N-1} (-1)^{s_t + \sigma_t} \tag{1}$$

where N is the number of keystream bits we need to be able to detect the bias, i.e., $N = \frac{2\ln(\tau_k)}{\varepsilon^2}$ where ε is the expected correlation. The time complexity of this algorithm is therefore proportional to

$$\tau_k \times N = \frac{2\tau_k \ln(\tau_k)}{\varepsilon^2} \; .$$

We will now show that this time complexity can be improved by using a fast Fourier transform even when H is nonlinear[2]. A similar technique has been described in [34,5] but in an attack against combination generators. We now prove that it also applies in our context.

Let $\langle \alpha^k \rangle$ denote the multiplicative subgroup of $\mathbb{F}_{2^n}^*$ generated by α^k, i.e., the set with τ_k elements $\{1, \alpha^k, \alpha^{2k}, \cdots, \alpha^{(\tau_k-1)k}\}$. This set is composed of all possible internal states $Y_0 = X_0^k$ which must be examined in the attack. Then, the attacker aims at finding the initial state $Y_0 \in \langle \alpha^k \rangle$ which maximizes the correlation given by (1) where $\sigma_t = H(Y_0 \alpha^{kt})$. For any $Y_0 \in \langle \alpha^k \rangle$, we compute

$$\mathcal{Z}(Y_0) = \sum_{t=0}^{N-1} (s_t \oplus \sigma_t) = \sum_{r=0}^{\tau_k - 1} \sum_{q=0}^{\lceil \frac{N-r}{\tau_k} \rceil - 1} (s_{q\tau_k + r} \oplus \sigma_r)$$

[2] The use of a fast Fourier transform for computing the correlation in the linear case has been pointed out by several authors including [8,26].

since for any t, $\sigma_t = \sigma_{t+\tau_k}$. We then deduce

$$\mathcal{Z}(Y_0) = \sum_{r=0}^{\tau_k-1} (\sigma_r \oplus 1) \left(\sum_{q=0}^{\lceil \frac{N-r}{\tau_k} \rceil - 1} s_{q\tau_k+r} \right) + \sum_{r=0}^{\tau_k-1} \sigma_r \left(\left\lceil \frac{N-r}{\tau_k} \right\rceil - \sum_{q=0}^{\lceil \frac{N-r}{\tau_k} \rceil - 1} s_{q\tau_k+r} \right) .$$

For any $0 \leq r < \tau_k$, we set

$$\mathcal{S}(r) = \sum_{q=0}^{\lceil \frac{N-r}{\tau_k} \rceil - 1} s_{q\tau_k+r} .$$

Then, we have

$$\begin{aligned} \mathcal{Z}(Y_0) &= \sum_{r=0}^{\tau_k-1} (\sigma_r \oplus 1)\mathcal{S}(r) + \sum_{r=0}^{\tau_k-1} \sigma_r \left(\left\lceil \frac{N-r}{\tau_k} \right\rceil - \mathcal{S}(r) \right) \\ &= \sum_{r=0}^{\tau_k-1} (-1)^{\sigma_r} \left(\mathcal{S}(r) - \frac{1}{2} \left\lceil \frac{N-r}{\tau_k} \right\rceil \right) + \frac{N}{2} . \end{aligned}$$

It follows that

$$\sum_{t=0}^{N-1} (-1)^{s_t + \sigma_t(Y_0)} = N - 2\mathcal{Z}(Y_0) = \sum_{r=0}^{\tau_k-1} (-1)^{\sigma_r(Y_0)} \left(\left\lceil \frac{N-r}{\tau_k} \right\rceil - 2\mathcal{S}(r) \right) .$$

We need to compute this value for $Y_0 = \alpha^{ik}$ for every $0 \leq i < \tau_k$. But,

$$\sigma_t(\alpha^{ik}) = H(\alpha^{ik}\alpha^{tk}) = H(\alpha^{(t+i)k}) = \sigma_{t+i}(1) .$$

In other words, we search for the integer i, $0 \leq i < \tau_k$ which maximizes the value

$$\sum_{r=0}^{\tau_k-1} (-1)^{\sigma_{r+i \bmod \tau_k}(1)} \left(\left\lceil \frac{N-r}{\tau_k} \right\rceil - 2\mathcal{S}(r) \right) ,$$

which corresponds to the convolution product of two vectors of length τ_k, namely $(\sigma_t(1))_{0 \leq t < \tau_k}$ and $(\mathcal{S}(t))_{0 \leq t < \tau_k}$. This can be done efficiently with a fast Fourier transform with time complexity $\mathcal{O}(\tau_k \log \tau_k)$ (see e.g. [3] or [23, Page 299]). The memory complexity of the attack is then $\mathcal{O}(\tau_k)$ and the overall time complexity (including the computation of all $\mathcal{S}(t)$) is then roughly

$$\mathsf{Time} = \tau_k \log \tau_k + \frac{2\ln(\tau_k)}{\varepsilon^2} .$$

Example 3. Let us consider the LFSR of size 12 with characteristic polynomial $P(X) = X^{12} + X^{10} + X^9 + X^8 + X^7 + X^5 + X^4 + X^3 + X^2 + X + 1$ and filtered by the same 6-variable function as in Example 2, but where the inputs of F are now defined by the tapping sequence $(\gamma_1, \ldots, \gamma_6) = (11, 10, 7, 5, 2, 0)$.

Then, the correlation between F and any function of the form $G = \mathsf{Tr}(\lambda x^k)$ with $k = \ell\frac{2^n-1}{2^m-1}$ and $\gcd(\ell, 2^n-1) = 1$ is too low for improving on the classical correlation attack. However, we can use $k = 45$ which satisfies $\mathsf{ord}(\alpha^k) = 91$. In this case, we are able to get a higher correlation since we allow all possible functions H, not only the linear ones. Here, the best approximation by a function of the form $G(x) = H(x^k)$ gives us a correlation equal to 0.125. With an FFT, the attack requires roughly $(592 + 574) = 1166$ operations, and 574 keystream bits. The whole initial state can then be recovered by an exhaustive search.

4.6 Approximation of the filtering function by $H(x^k)$

All previous correlation attacks exploit the existence of a function G of the form $G(x) = H(x^k)$ for some k with $\gcd(k, 2^n-1) > 1$, which provides a good approximation of F. In particular, the fast correlation attacks involving a shorter LFSR point out that the notion of generalized nonlinearity as defined in [41] must be extended in order to capture these new attacks: it appears that the distance of the filtering function to all $\mathsf{Tr}(\lambda x^k)$ with $k = \ell \times \frac{2^n-1}{2^m-1}$ where m is a divisor of n and $\gcd(\ell, 2^n-1) = 1$ is a much more relevant quantity than its distance to the components of monomial permutations.

Moreover, even if such a fast correlation attack is not feasible, for instance if n is a prime, an efficient correlation attack may be possible based on the approximation of F by $G(x) = H(x^k)$ for some k with $\gcd(k, 2^n-1) > 1$. As observed in the previous example, the fact that H can be nonlinear usually yields a higher correlation. The best approximation of the form $G(x) = H(x^k)$ can be computed from F as follows. For the sake of simplicity, we now suppose that k is a divisor of (2^n-1), or equivalently that $\tau = (2^n-1)/k$ (otherwise, we get similar results by replacing k by $\gcd(k, 2^n-1)$). Let $\langle\alpha^\tau\rangle$ be the cyclic subgroup of $\mathbb{F}_{2^n}$ of order k. Then, by shifting this cyclic subgroup, we obtain the sets $E_i = \alpha^i\langle\alpha^\tau\rangle$, for $0 \le i < \tau$ which provide the partition

$$\mathbb{F}_{2^n}^* = \bigcup_{i=0}^{\tau-1} E_i$$

where all sets E_i, for $0 \le i < \tau$, are disjoint. It follows that G is constant on any set E_i since, for $x = \alpha^i \times \alpha^{j\tau}$, we have

$$G(x) = H((\alpha^i\alpha^{j\tau})^k) = H(\alpha^{ik}) \; .$$

The correlation between F and G can therefore be expressed as follows:

$$\begin{aligned}\sum_{x\in\mathbb{F}_{2^n}} (-1)^{F(x)+H(x^k)} &= 1 + \sum_{x\in\mathbb{F}_{2^n}^*} (-1)^{F(x)+H(x^k)} \\ &= 1 + \sum_{i=0}^{\tau-1} (-1)^{H(\alpha^{ik})} \left(\sum_{y\in E_i} (-1)^{F(y)}\right) \; . \qquad (2)\end{aligned}$$

If $\gcd(k,\tau)=1$, all values α^{ik}, for $0 \leq i < \tau$ belong to different sets E_j. Hence, the function H which maximizes this correlation is the function defined by

$$H(\alpha^{ik}) = \begin{cases} 0 & \text{if } \sum_{y\in E_i}(-1)^{F(y)} > 0 \\ 1 & \text{if } \sum_{y\in E_i}(-1)^{F(y)} < 0 \end{cases}$$

In other words, $H(\alpha^{ik})=1$ if and only if the Hamming weight of the restriction of F to E_i is strictly greater than $k/2$. It can be observed that H is uniquely determined because the weight of the restriction of F cannot be equal to $k/2$ since k is odd. This also implies that, for the optimal choice of H, we obtain

$$\sum_{x\in\mathbb{F}_{2^n}} (-1)^{F(x)+H(x^k)} = 1 + \sum_{i=0}^{\tau-1} \left| \sum_{y\in E_i} (-1)^{F(y)} \right| \geq 1+\tau$$

since each term in the sum is at least 1. Therefore, for any F, we can always find a function H such that the correlation between F and $G(x)=H(x^k)$ is at least $(1+\tau)2^{-n} \simeq k^{-1}$. It is worth noticing that this lower bound on the correlation does not decrease when the LFSR length n increases.

In the case where $\gcd(k,\tau)=d>1$, we have that α^{ik} and $\alpha^{(i+\frac{\tau}{d})k}$ belong to the same set E_j. Indeed, $\alpha^{\frac{k\tau}{d}} \in \langle \alpha^\tau \rangle$. Equation (2) can then be rewritten as

$$\sum_{x\in\mathbb{F}_{2^n}} (-1)^{F(x)+H(x^k)} = 1 + \sum_{i=0}^{\frac{\tau}{d}-1} (-1)^{H(\alpha^{ik})} \left(\sum_{j=0}^{d-1} \left(\sum_{y\in E_{i+j\frac{\tau}{d}}} (-1)^{F(y)} \right) \right) .$$

In this case, the value of H at point α^{ik} is defined by the weight of the restriction of F to the set $\bigcup_{j=0}^{d-1} E_{i+j\frac{\tau}{d}}$. Using again that this set has an odd cardinality, we get that the correlation between F and $G(x)=H(x^k)$ is at least $(1+\frac{\tau}{d})2^{-n}$.

While in usual (fast) correlation attacks, choosing a filtering function with a high nonlinearity guarantees that the attack will be infeasible, this is not the case here. For instance, some bent functions in the so-called class $\mathcal{PS}^-$ [11] are constant on all sets $\lambda\langle\alpha^\tau\rangle$ for $\tau = 2^{n/2}+1$, while they have the best nonlinearity.

The previous results enable us to find the best approximation of F by a function of the form $H(x^k)$. However, improving the complexity of this search when n grows and F depends on a few inputs only remains an open issue. Indeed, it seems difficult to use this property of F to simplify the search for the optimal H. Another open problem is to be able to find in an efficient way the best approximation of the form $G(x)=\mathsf{Tr}(\lambda x^k)$.

5 Conclusions

While the monomial equivalence introduced by Rønjom and Cid does not affect the security of filter generators regarding algebraic attacks, it usually allows to decrease the complexity of correlation attacks and their variants. Most importantly, considering a non-bijective monomial mapping enables the attacker to

mount a divide-and-conquer attack by decomposing the set of all nonzero initial states with respect to some multiplicative subgroup having a smaller order. If the LFSR length is not a prime, the involved subgroup may be a subfield and this divide-and-conquer attack can be further improved as in fast correlation attacks. A counter-measure to avoid these attacks then consists in choosing for the LFSR length a Mersenne prime, i.e. both n and $(2^n - 1)$ are prime.

References

1. Ågren, M., Löndahl, C., Hell, M., Johansson, T.: A survey on fast correlation attacks. Cryptography and Communications 4(3-4), 173–202 (2012)
2. Blahut, R.E.: Theory and practice of error control codes. Addison-Wesley (1983)
3. Blahut, R.E.: Fast algorithms for digital signal processing. Addison-Wesley (1985)
4. Canteaut, A.: Filter generator. In: Encyclopedia of Cryptography and Security, 2nd Ed., pp. 726–729. Springer (2011)
5. Canteaut, A., Naya-Plasencia, M.: Correlation attacks on combination generators. Cryptography and Communications 4(3-4), 147–171 (2012)
6. Canteaut, A., Trabbia, M.: Improved fast correlation attacks using parity-check equations of weight 4 and 5. In: Advances in Cryptology - EUROCRYPT'2000. LNCS, vol. 1807, pp. 573–588. Springer-Verlag (2000)
7. Chepyshov, V., Johansson, T., Smeets, B.: A simple algorithm for fast correlation attacks on stream ciphers. In: Fast Software Encryption - FSE 2000. LNCS, vol. 1978, pp. 181–195. Springer-Verlag (2000)
8. Chose, P., Joux, A., Mitton, M.: Fast correlation attacks: an algorithmic point of view. In: Advances in Cryptology - EUROCRYPT 2002. LNCS, vol. 2332, pp. 209–221. Springer-Verlag (2002)
9. Courtois, N.: Fast algebraic attacks on stream ciphers with linear feedback. In: Advances in Cryptology - CRYPTO 2003. LNCS, vol. 2729, pp. 176–194. Springer (2003)
10. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with linear feedback. In: Advances in Cryptology - EUROCRYPT 2003. LNCS, vol. 2656, pp. 345–359. Springer-Verlag (2003)
11. Dillon, J.: Elementary Hadamard difference sets. Ph.D. thesis, Univ Maryland (1974)
12. Dobbertin, H.: Construction of bent functions and balanced Boolean functions with high nonlinearity. In: Fast Software Encryption - FSE'94. LNCS, vol. 1008, pp. 61–74. Springer-Verlag (1994)
13. ECRYPT - European Network of Excellence in Cryptology: The eSTREAM Stream Cipher Project. `http://www.ecrypt.eu.org/stream/` (2005)
14. Golic, J.D.: On the security of nonlinear filter generators. In: Fast Software Encryption - FSE'96. LNCS, vol. 1039, pp. 173–188. Springer-Verlag (1996)
15. Golomb, S.W., Gong, G.: Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar. Cambridge University Press (2004)
16. Gong, G.: A closer look at selective DFT attacks. CACR report 2011-35, University of Waterloo (2011)
17. Gong, G., Rønjom, S., Helleseth, T., Hu, H.: Fast discrete Fourier spectra attacks on stream ciphers. IEEE Trans. Inform. Theory 57(8), 5555–5565 (2011)
18. Hell, M., Johansson, T., Brynielsson, L.: An overview of distinguishing attacks on stream ciphers. Cryptography and Communications 1(1), 71–94 (2009)

19. Helleseth, T.: Maximal-length sequences. In: Encyclopedia of Cryptography and Security, 2nd Ed., pp. 763–766. Springer (2011)
20. Helleseth, T., Rønjom, S.: Simplifying algebraic attacks with univariate analysis. In: Information Theory and Applications - ITA 2011. pp. 153–159. IEEE (2011)
21. Johansson, T., Jönsson, F.: Improved fast correlation attack on stream ciphers via convolutional codes. In: Advances in Cryptology - EUROCRYPT'99. LNCS, vol. 1592, pp. 347–362. Springer-Verlag (1999)
22. Johansson, T., Jönsson, F.: Fast correlation attacks through reconstruction of linear polynomials. In: Advances in Cryptology - CRYPTO'00. LNCS, vol. 1880, pp. 300–315. Springer-Verlag (2000)
23. Joux, A.: Algorithmic Cryptanalysis. Chapman & Hall/CRC (2009)
24. Key, E.L.: An analysis of the structure and complexity of nonlinear binary sequence generators. IEEE Trans. Inform. Theory 22, 732–736 (1976)
25. Lidl, R., Niederreiter, H.: Finite Fields. Cambridge University Press (1983)
26. Lu, Y., Vaudenay, S.: Faster correlation attack on Bluetooth keystream generator E0. In: Advances in Cryptology - CRYPTO 2004. LNCS, vol. 3152, pp. 407–425. Springer-Verlag (2004)
27. MacWilliams, F.J., Sloane, N.J.: The theory of error-correcting codes. North-Holland (1977)
28. Massey, J.L., Serconek, S.: A Fourier transform approach to the linear complexity of nonlinearly filtered sequences. In: Advances in Cryptology - CRYPTO '94. LNCS, vol. 839, pp. 332–340. Springer (1994)
29. Maximov, A., Johansson, T., Babbage, S.: An improved correlation attack on A5/1. In: Selected Areas in Cryptography - SAC 2004. LNCS, vol. 3357, pp. 1–18. Springer-Verlag (2004)
30. McEliece, R.J.: Finite Fields for Computer Scientists and Engineers. Kluwer (1987)
31. Meier, W., Pasalic, E., Carlet, C.: Algebraic attacks and decomposition of Boolean functions. In: Advances in Cryptology - EUROCRYPT 2004. LNCS, vol. 3027, pp. 474–491. Springer-Verlag (2004)
32. Meier, W., Staffelbach, O.: Fast correlation attack on certain stream ciphers. Journal of Cryptology pp. 159–176 (1989)
33. Mihaljevic, M.J., Fossorier, M.P., Imai, H.: A low-complexity and high performance algorithm for the fast correlation attack. In: Fast Software Encryption – FSE 2000. LNCS, vol. 1978. Springer-Verlag (2000)
34. Naya-Plasencia, M.: Cryptanalysis of Achterbahn-128/80. In: Fast Software Encryption - FSE 2007. LNCS, vol. 4593, pp. 73–86. Springer (2007)
35. Rønjom, S.: Powers of subfield polynomials and algebraic attacks on word-based stream ciphers. IACR Cryptology ePrint Archive 2015, 495 (2015)
36. Rønjom, S., Cid, C.: Nonlinear equivalence of stream ciphers. In: Fast Software Encryption – FSE 2010. LNCS, vol. 6147, pp. 40–54. Springer (2010)
37. Rønjom, S., Gong, G., Helleseth, T.: On attacks on filtering generators using linear subspace structures. In: Sequences, Subsequences, and Consequences - SSC 2007. LNCS, vol. 4893, pp. 204–217. Springer (2007)
38. Rønjom, S., Helleseth, T.: A new attack on the filter generator. IEEE Trans. Inform. Theory 53(5), 1752–1758 (2007)
39. Rueppel, R.A.: Analysis and Design of stream ciphers. Springer-Verlag (1986)
40. Siegenthaler, T.: Decrypting a class of stream ciphers using ciphertext only. IEEE Trans. Computers C-34(1), 81–84 (1985)
41. Youssef, A.M., Gong, G.: Hyper-bent functions. In: Advances in Cryptology - EUROCRYPT 2001. LNCS, vol. 2045, pp. 406–419. Springer (2001)
42. Zierler, N.: Linear recurring sequences. J. Soc. Indus. Appl. Math. 7, 31–48 (1959)