Revisiting Correlation-Immunity in Filter Generators

Aline Gouget[1] and Hervé Sibert[2]

[1] Gemalto, 6 rue de la Verrerie, F-92190 Meudon, France
aline.gouget@gemalto.com
[2] NXP Semiconductors, 9 rue Maurice Trintignant, F-72081 Le Mans Cedex 9, France
herve.sibert@nxp.com

Abstract. Correlation-immunity is a cryptographic criterion on Boolean functions arising from correlation attacks on combining functions. When it comes to filtering functions, the status of correlation-immunity lacks study in itself and, if it is commonly accepted as a requirement for nonlinear filter generators, this is for other concerns. We revisit the concept of correlation-immunity and clear up its meaning for filtering functions. We summarize existing criteria similar to correlation-immunity and attacks in two different models, showing that such criteria are not relevant in both models. We also derive a precise property to avoid correlations due to the filter function only, which appears to be a bit looser than correlation-immunity. We then propose new attacks based on whether this property is verified.

Keywords: Nonlinear filter generator, Boolean function, correlation-immunity, distinguishing attacks.

1 Introduction

Most stream ciphers proposed in the literature are built upon Linear Feedback Shift Registers (LFSR). One well-known proposal for destroying the linearity inherent to LFSRs is to use a nonlinear function to filter the contents of a single LFSR. All the components of a filter generator (i.e. the LFSR, the filtering function and the taps) must be chosen carefully to ensure the cryptographic security of the keystream generated by the generator. As often in symmetric cryptography, criteria on the filter generator components are mostly derived from known attacks.

The correlation-immunity property is a well-known cryptographic criterion for Boolean functions. Correlation-immunity is sometimes stated as a criterion dedicated to combining functions only, and sometimes as a requirement that also applies to filtering functions. In order to clear up the role of correlation-immunity for filtering functions, we investigate known distinguishing attacks on filter generators that consist in finding correlation relations between the keystream bits by using properties of the filter function only.

C. Adams, A. Miri, and M. Wiener (Eds.): SAC 2007, LNCS 4876, pp. 378–395, 2007.

1.1 Related Work

The *nonlinear filter model* is a classical model of synchronous stream ciphers that involves a nonlinear Boolean function to filter the contents of a single shift register.

The *correlation-immunity* criterion has been introduced by Siegenthaler [15] for combining functions, in order to protect them from a "divide and conquer" attack well-known under the name *(fast) correlation attack* [17,11,4,5]. These attacks also apply to nonlinear filter generators [16,7]. Notice that such attacks require that the internal state memory of the generator is updated in a deterministic way. The only criterion on the filtering function involved in this attack is the nonlinearity of the Boolean function, not the correlation-immunity. Canteaut and Filiol [3] studied the fast correlation attack given in [5] for filter generators and they showed that the keystream length which guarantees a successful attack does not depend on the filtering function, except for functions which are very close to affine functions. Then, they suggest that the choice of the Boolean function in the design of a filter generator should be mostly conditioned by other types of attacks. Thus, fast correlations attacks are out of the scope of this paper.

Anderson [1] found other correlations in nonlinear filter generators and proposed an *optimum correlation attack*. This attack is based on the (un)balancedness of the *augmented filter function*. The update of the internal state memory of the generator is assumed to be probabilistic. Hence, this attack does not take advantage of a deterministic update, and it targets correlation relations between the keystream bits that arise from properties of the filter function only. Golic [8] studied a different definition of the augmented filter function and derived a construction of Boolean functions that resist the optimum correlation attack. Still in [8], Golic recommended to use in practice only filtering functions coming from his construction (with additional criteria on the filtering function including correlation-immunity). However, it is unclear to what extent this construction captures all the filtering functions that are immune to this attack, as Dichtl [6] showed by exhibiting such a filtering function that does not follow Golic's construction.

The relevance of the correlation-immunity criterion for filtering functions has been partially studied by Ding *et al.* [7]. Many Boolean functions which are not correlation-immune can be transformed into correlation-immune functions by performing a linear transform on the input variables and adding a linear function. Indeed, Ding *et al.* gave a general method to construct, from a correlation immune function f that filters an LFSR, an equivalent filter generator which differs from the original one only by its initial state vector and by its filter function g, which is not correlation immune. Even if there is no efficient method known to construct such an equivalent generator, stream ciphers with correlation immune filter functions are theoretically vulnerable provided that those with non-correlation-immune functions are. In [7], the authors concluded that using correlation-immune filter functions may not get any advantage in the case when the filter function and the feedback polynomial of the LFSR are known.

Thus, from the state of the art on the application of the correlation-immunity criterion to filtering functions, it is still unclear to what extent one must or not choose a correlation-immune function when designing a filter generator.

1.2 Our Contribution

In this paper, we give in-depth analysis of correlation-related criteria in filter generators. We investigate known distinguishing attacks on filter generators that take advantage of correlation relations between the keystream bits that arise from properties of the filter function only. So as to better understand the attacks, we introduce two security models for filter generators depending on the memory update procedure: the *probabilistic* nonlinear filter model and the *deterministic* nonlinear filter model. We show that considering separately these two models helps to shed light on the design criterion for filtering function, while there is no interest to do the same for combining generators.

We revisit the *optimal correlation attack* [1,8] that targets correlation due to the filtering function. We precisely study the criteria to resist this attack depending on whether it is performed in the probabilistic or in the deterministic model. We show that the relevance of this criterion in the deterministic model is questionable, and that it does not target the initial attack in this model.

Next, we reconsider the original observation of Anderson and give a practical criterion on the filter to avoid the optimal correlation attack in both models. This criterion also thwarts a recent distinguishing attack focusing on a filtering function [19]. We call this new criterion *quasi-immunity*, since it appears to be a bit looser than correlation-immunity. This criterion embeds previous criteria, and it turns out to be the criterion most directly related to correlations of the filtering function.

We then provide the complexity of different types of attack against filtering function that do or do not meet the quasi-immunity requirement. We show that if the filtering function f does not fulfil the quasi-immunity criterion (of order 1), then there always exists a distinguisher between random sequences and keystream outputted by the filter generator even when considering the probabilistic filter generator model. We next evaluate the cost of state recovery attack depending on whether the filtering function fulfils the quasi-immunity criterion. Finally, we discuss the construction of equivalent filter generators that are potentially weaker against such attacks.

1.3 Organization of the Paper

In Section 2, we give the main cryptographic properties of Boolean functions, we briefly describe the components of filter generators and update procedure, and we summarize well-known criteria on the filter generator components. In Section 3, we study correlation attacks targeted at the filtering function in filter generators, and next we derive a new criterion called “quasi-immunity” criterion. In Section 4, we study the complexity of general attacks for filters that do or do not meet the new criterion. At last, we give directions for future work and we conclude.

2 Preliminaries

In this section, we briefly recall the main properties of Boolean functions. Next, we describe the components of a filter generator and give the main known design criteria.

2.1 Boolean Functions

Every n-variable Boolean function f can be uniquely represented by its *algebraic normal form*, $f(x_1, \ldots, x_n) = \sum_{I \subseteq \{1,\ldots,n\}} a_I \prod_{i \in I} x_i$, where the a_I's are in $\mathbb{F}_2$. The terms $\prod_{i \in I} x_i$ are called *monomials*. For any Boolean function f of n variables, we denote by $\mathcal{F}(f)$ the quantity $\mathcal{F}(f) = \sum_{x \in GF(2)^n} (-1)^{f(x)} = 2^n - 2w_H(f)$, where $w_H(f)$ is the Hamming weight of f, related to the Fourier transform of f. In the following, we denote by $e_1, \ldots, e_n$, the n coordinate vectors of the vector space $GF(2)^n$ with Hamming weight 1. For $u \in GF(2)^n$, we denote by φ_u the linear Boolean function $x \mapsto x \cdot u$ where $\cdot$ denotes the inner product.

A Boolean function f is called *balanced* if 0 and 1 have the same number of preimages by f. The *nonlinear order* of a Boolean function f equals the maximum degree of those monomials whose coefficients are nonzero in its algebraic normal form. The nonlinearity of an n-variable Boolean function f is the minimum Hamming distance between f and the set of affine functions.

An n-variable Boolean function f is *correlation-immune of order* m with $1 \leq m \leq n$ if the output of f and any m input variables are statistically independent. The correlation-immunity criterion can be characterized by means of Walsh coefficients:

Proposition 1. **[20]** *A Boolean function $f : GF(2)^n \rightarrow GF(2)$ is correlation-immune of order m if, and only if, $\mathcal{F}(f + \varphi_u) = \sum_{x \in GF(2)^n} (-1)^{f(x)+u \cdot x} = 0$ for all u with $1 \leq w_H(u) \leq m$.*

The nonlinear order and the nonlinearity of a Boolean function are both affine invariant whereas the correlation-immunity is not [12].

2.2 Nonlinear Filter Generators

A nonlinear filter generator is defined by a finite memory, a filtering function, a tapping sequence defining the input stages to the filter function and a procedure to update the memory.

Finite memory. We assume that every nonlinear filter has a finite input memory of r bits. The value of the initial state of the memory is assumed to be random. At each time t, the $r - 1$ first bits of the memory are shifted right by one position and the leftmost bit is a new bit, that is either random, or determined by the current bits in the register. The indexes in the register are numbered from right to left, starting at 1. We denote by $s = (s_t)_{t=-r}^{\infty}$ the binary sequence of the state memory. Then, the finite sequence $(s_t)_{t=-r}^{-1}$ is the initial state of the memory.

It is recommended to choose $r \geq 2L$ where 2^L is the target security level to avoid time-memory tradeoff attacks [2,9]. More precisely, the number of possible initial states before keystream generation should be at least 2^{2L}.

Filtering function. Let f be a Boolean function of n non-degenerate input variables with $1 \leq n \leq r$. The inputs of the filtering function f are some values $s_{t-\gamma_1}, s_{t-\gamma_2}, \ldots, s_{t-\gamma_n}$ of the finite memory, where $\gamma = (\gamma_i)_{i=1}^n$ is an increasing sequence of positive integers such that $\gamma_1 = 1$, and $\gamma_n \leq r$. The output sequence $z = (z_t)_{t=0}^\infty$ of f is called the keystream sequence.

The function f must be balanced since the output sequence is expected to be balanced. The nonlinear order of f must be high enough and f should include many terms of each order up to the nonlinear order of f [13]. Indeed, filter generators can be vulnerable to the Berlekamp-Massey algorithm if the linear complexity of the output sequence is too small. Also, the Boolean function f must not be close to affine functions in order to avoid fast correlation attacks [3].

Taps. The sequence $\gamma = (\gamma_i)_{i=1}^n$ defining the indexes of the input to the filtering function is called the *tapping sequence*, and the corresponding output sequence $z = (z_t)_{t=0}^\infty$ is defined by $z_t = f(s_{t-\gamma_1}, \ldots, s_{t-\gamma_n}), \quad t \geq 0$. The choice of the tapping sequence defining the input stages to the filter function f must be done as indicated in [8]: the input memory size should be close to its maximum value $r - 1$, and the set of the tap positions should be a full positive difference set.

Update of the leftmost bit. In the literature, depending on the context, authors either consider that the leftmost bit is a random bit, or that it is determined by the current bits in the register. Nevertheless, these two points of view and their impact in terms of security model have not been studied or even underlined. We call these two models respectively the *probabilistic* nonlinear filter model and the *deterministic* nonlinear filter model.

Probabilistic nonlinear filter model. At each time t, the leftmost bit b is the output of an unbiased random bit source. In this case, the input sequence is perfectly random and then $s = (s_t)_{t=-r}^\infty$ is a random sequence. In this model, the aim of an attack is not to recover the key since the knowledge of $(s_t)_{t=-r}^{i-1}$ does not reveal anything about s_i. Here, the aim is to distinguish the keystream sequence z from a random sequence. Thus, an attack on the nonlinear filter generator in the probabilistic model reveals weaknesses of the filter.

Deterministic nonlinear filter model. At each time t, the leftmost bit b is computed from the current memory state, *e.g.* by using a linear feedback of the register. The best-known criterion on the feedback polynomial is that it should be a primitive polynomial of degree r to ensure that the LFSR sequence $s = (s_t)_{t=-r}^\infty$ is a binary maximum-length sequence of period $2^r - 1$ [14]. In this model, the aim of an attack can be either to recover the initial state or to distinguish the keystream from a random sequence.

A successful attack in the probabilistic nonlinear filter model can be adapted to the deterministic model, whereas the converse is not true. However, a criterion to prevent an attack in the probabilistic model does not always translate to the deterministic model.

3 Correlation Attacks on the Filtering Function

In this section, we first review the *optimal correlation attack* presented by Anderson [1] that targets correlations due to the filtering function, before studying criteria to resist this attack in the probabilistic and deterministic models. Next, we consider a distinguishing attack on a filter generator that targets exactly the *optimal correlation* of Anderson. At last, we deduce the *quasi-immunity* criterion for filtering functions.

In the sequel, we assume the filtering function f to be balanced.

3.1 The Optimal Correlation Attack

The *optimal correlation attack* proposed by Anderson [1] is the first attack on filter generators that exploits correlations due to the filtering function only. This attack relies on the fact that each bit going along the register is input to the filtering function at each one of its taps. This results in correlations between the internal register state and the keystream produced. These correlations are avoided if an *augmented filter function* defined accordingly is balanced.

This *augmented filter function* is constructed as follows: consider a single bit b moving along the register. Each time this bit is at a tap location, the filter combines it with other register bits to form a keystream bit. The augmented function is the vectorial function that maps all these (independent) register bits to the n-bit-vector consisting of the n values that involve bit b. One can then distinguish the generator from a random sequence by studying the distribution of the n-tuples in the output sequence that correspond to the output of the augmented filter function.

Anderson provides an example of a filter whose taps are consecutive entries of the register:

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 + x_2 + (x_1 + x_3)(x_2 + x_4 + x_5) + (x_1 + x_4)(x_2 + x_3)x_5.$$

This Boolean function is balanced, correlation-immune of order 2 and of nonlinear order 3. However, the augmented function that maps 9-tuples of the shift register sequence to 5-tuples of the keystream output is not balanced, which yields an attack. Notice that here, as the attacks takes place in the probabilistic model, we assume that all 9-tuples are equiprobable.

3.2 Analysis of the Optimal Correlation Attack - Probabilistic Model

Both in [1] and in [8], the authors consider a probabilistic model in which the input sequence $s = (s_t)_{t=-r}^{\infty}$ is regarded as a sequence of balanced and independent bits. The output sequence $z = (z_t)_{t=0}^{\infty}$ is a sequence of balanced bits if and only if the filter function f is balanced. The aim of the attacker is to distinguish the keystream outputted by the filter from a random sequence.

Augmented filter function. The augmented filter function $\bar{h}$ constructed by Anderson in [1] makes it possible to find an *optimal correlation* between the output keystream bits and the internal state of the register. The keystream bit produced at time t is equal to

$$z_t = f(s_{t-\gamma_1}, \ldots, s_{t-\gamma_n}) \ .$$

The function $\bar{h}$ is defined as follows. Consider the n^2 (not necessarily distinct) variables involved in the n values of the filter function at time $t+\gamma_1, \ldots, t+\gamma_n$, which all involve the bit s_t, and denote by G the set of all independent variables among those n^2 variables. The function $\bar{h}$ maps every element of G to the corresponding n-tuple of keystream bits $(z_{t+\gamma_i})_{i=1\ldots n}$.

In [8], Golic studied the randomness of the keystream in the probabilistic model. Assuming that the input sequence $s = (s_t)_{t=-r}^{\infty}$ is a sequence of balanced and independent bits, the output sequence $z = (z_t)_{t=0}^{\infty}$ is a sequence of balanced bits if and only if the filter function f is balanced. The output sequence z is *purely random* if and only if for each $t \geq 0$, the output bit z_t is balanced for any fixed value of the previous output bits $(z_i)_{i=0}^{t-1}$.

For a finite nonlinear filter generator with input memory size r, z_t depends only on the current input bit s_t and on the r preceding input bits $(s_i)_{i=t-r}^{t-1}$. Golic showed that the output sequence is purely random given that the input sequence is such if and only if the vectorial Boolean function F_{M+1} that maps $2M+1$ consecutive input bits to the $M+1$ corresponding consecutive output bits is balanced, where $M = \gamma_n - \gamma_1$.

It appears that Golic's construction generalizes the augmented filter function $\bar{h}$ and the corresponding attack to an arbitrary choice of taps for the filter. The criterion for the keystream to be *purely random* and thus to resist the optimum correlation attack in the probabilistic model is the balancedness of this *new* augmented filter function.

We now precisely establish the link between the augmented functions of Anderson and Golic.

Proposition 2. *If the augmented function of Golic F_{M+1} is balanced, then the augmented function of Anderson $\bar{h}$ is balanced.*

Proof. The functional graph in Figure 1 links $\bar{h}$ and F_{M+1} augmented functions, with P and Q being projections respectively from the $2M+1$ bit variables onto those involved in $\bar{h}$, and from the $M+1$ consecutive output bits to the subset

$$\begin{tikzcd}
\mathbb{F}_2^{2M+1} \arrow[r, "F_{M+1}"] \arrow[d, "P"'] & \mathbb{F}_2^{M+1} \arrow[d, "Q"] \\
G \arrow[r, "\bar{h}"] & \mathbb{F}_2^n
\end{tikzcd}$$

Fig. 1. Commutative diagram of augmented functions of Anderson and Golic

of n output bits observed at $t+\gamma_n, \ldots, t+\gamma_1$. Using the commutative diagram in Figure 1, the proof is straightforward. □

Remark 1. The augmented function $\bar{h}$ is a restriction of the augmented function F_{M+1}, and both functions $\bar{h}$ and F_{M+1} coincide if all the filter taps are consecutive. Thus, $\bar{h}$ being balanced does not imply that F_{M+1} also is. Indeed, for the register with output $z_t = s_{t-3} + s_{t-6} \cdot s_{t-1}$, the function $\bar{h}$ is balanced, whereas F_{M+1} is not.

Golic's formulation in the same framework as Anderson is thus a generalization that enables finding optimal so-called correlations, as it involves the whole memory of the generator. Thus, a nonlinear filter generator is immune to the optimum correlation attack in the probabilistic model if, and only if, Golic's augmented filter function is balanced.

Unfortunately, straightforward study of the balancedness of F_{M+1} is too complex when the taps of the function are located at both ends of the register as recommended in [8].

Criterion on the filter function. We now study the criterion on the filter function for the augmented filter function F_{M+1} to be balanced, which is equivalent to the output being purely random. Golic in [8] gave a characterization in terms of the filter function f and the tapping sequence γ in the following theorem, for which only the sufficiency of the conditions was proven:

Theorem 1. **[8]** *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) $f(x_1, \ldots, x_n)$ is balanced for each value of $(x_2, \ldots, x_n)$, that is, if*

$$f(x_1, \ldots, x_n) = x_1 + g(x_2, \ldots, x_n), \tag{1}$$

or if $f(x_1, \ldots, x_n)$ is balanced for each value of $(x_1, \ldots, x_{n-1})$, that is, if

$$f(x_1, \ldots, x_n) = x_n + g(x_1, \ldots, x_{n-1}), \tag{2}$$

Function F_{M+1} depends on the choice of the taps, while Theorem 1 gives a characterization independent from the tap sequence. However, filtering functions that yield a purely random output for a specific choice of the taps exist, thus contradicting Theorem 1. Indeed, Sumarokov in [18] had already defined perfectly balanced Boolean functions as those functions whose augmented function is balanced when the taps are consecutive, and had given an example that is not of the form (1) or (2). Dichtl [6] also found a similar filtering function. More recently, Logachev [10] gave a general construction to obtain new such functions.

Then, it appears that perfect balancedness of filter functions was not properly defined by Golic, and the definition should enclose the choice of the taps. The filter function to consider is thus the $M+1$-variable Boolean function constructed from f and $\gamma = (\gamma_i)_{i=1}^{n}$ by adding $M+1-n$ mute variables. However, filtering functions of the form (1) or (2) have the particularity that the associated augmented functions are balanced regardless of the choice of the taps.

To summarize, the set of filters that thwart the optimum correlation attack in the probabilistic model includes not only the functions from [8], but also functions whose suitability may depend on the choice of the taps.

3.3 Analysis of the Optimal Correlation Attack - Deterministic Model

We now consider a deterministic model such that the memory is updated using a deterministic linear relation. At each clock, the new leftmost bit is a linear combination of the memory state bits. Then, the input sequence $s = (s_t)_{t=-r}^{\infty}$ is regarded as a sequence of balanced bits which are dependent. The output sequence $z = (z_t)_{t=0}^{\infty}$ is a sequence of balanced bits if and only if the filter function f is balanced. The aim of the attacker is to distinguish the keystream, *i.e.* the output of the filtering function, from a random sequence.

In this case, the approach of [1] and [8] is not valid anymore. Indeed, a very simple counterexample shows that correlation may appear even in the case of functions of the form (1) or (2).

Proposition 3. *Consider the filter generator consisting of a* 4*-bit register with:*

$$\begin{cases} z_t = s_{t-2} + s_{t-4} \cdot s_{t-3} \\ s_t = s_{t-4} + s_{t-3} \end{cases}$$

The deterministic counterparts of the augmented functions of Anderson and Golic are unbalanced.

Proof. Anderson's augmented function $\bar{h}$ is defined as follows:

$$\begin{array}{llcl} \bar{h}: & \mathbb{F}_2^4 & \to & \mathbb{F}_2^3 \\ & s_{t-4}, s_{t-3}, s_{t-2}, s_{t-1} & \mapsto & (s_{t-2} + s_{t-4} \cdot s_{t-3}, s_{t-1} + s_{t-3} \cdot s_{t-2}, s_t + s_{t-2} \cdot s_{t-1}) \end{array}$$

Taking the correlation into account yields $s_t + s_{t-2} \cdot s_{t-1} = s_{t-4} + s_{t-3} + s_{t-2} \cdot s_{t-1}$. Thus, the edge random variable x_4 (in the random input model) which had a balancing role disappears, and, whenever pattern 101 appears in the keystream, the register content is 0101, hence the result. □

The reason for this observation is that, as feedback bits are produced by bits that have already passed through the register and mixed in previous values of the filter function, the criterion in Theorem 1 is less relevant. Indeed, there is no reason to consider the edge bits as being "more random" than the others, and to consider filtering functions of the form (1) or (2) only.

We now study the augmented function of Golic with respect to the deterministic model in general. Remember that the augmented function F_{M+1} maps $2M+1$ consecutive input bits to the $M+1$ corresponding consecutive output bits. A proper choice of the taps implies maximizing the size of the range of the inputs to the filter [8], so that the length of the register is equal to $M+1$. Therefore, among the $2M+1$ input bits of F_{M+1}, the last M bits are uniquely determined by the first $M+1$ input bits. Therefore, we have

Proposition 4. *Consider a register with length $M+1$, filtered by a Boolean function f whose distance between the extremity taps is M. In the deterministic model, the augmented function F_{M+1} maps the internal state of size $M+1$ to the first $M+1$ output bits.*

In the deterministic model, the balancedness of the original augmented filter function is not relevant, as not all inputs of the function are possible. Therefore, instead of studying the augmented function F_{M+1} itself, it is necessary to study its restriction to its possible inputs. This amounts to study the balancedness of the first $M+1$ output bits of the nonlinear filter, which is related to well-known distinguishing attacks consisting in studying the distribution of the first output bits, and also to algebraic attacks.

3.4 A Practical Criterion to Avoid Optimal Correlations

As we have seen, in the deterministic model, not only cannot we assume that the leftmost bit is perfectly random, but also the definition of the augmented filter function is no longer sound. Instead of studying the augmented function, it is necessary to take the feedback function into account and to study the output sequence itself.

Therefore, in this section, we refer to the probabilistic model, and we consider a distinguishing attack on a filter generator that attempts to exploit a weakness of the filtering function only to distinguish the output of the filtering function from a random sequence.

The study of the balancedness of Golic's augmented filter function F_{M+1} captures related biases, but the complexity is too high when the length between extreme taps is maximal: in this case, F_{M+1} maps $2r-1$-bit-vectors to r-bit-vectors, which makes finding a bias as hard as an exhaustive search.

We thus come back to the original idea of Anderson in [1] to derive a criterion that prevents optimal correlations from appearing in the output, by considering only the n output bits that share an equal bit in the input to the filter.

The aim of the attack is to correlate n keystream bits that are output within intervals equal to each difference between two consecutive tap positions having at least one bit in common.

We denote by $x(t)$ the input of the filtering function at time t, *i.e.* , $x(t) = [s_{t-\gamma_1}, \ldots, s_{t-\gamma_n}]$. At time t, the value of the i-th variable x_i which is $s_{t-\gamma_i}$ is denoted by $x_i(t)$.

Proposition 5. *Consider a nonlinear filter generator with filter f, where f is an n-variable Boolean function. Assume that the input sequence $s = (s_t)_{t=-r}^{\infty}$ is purely random, and that the tapping sequence γ is a full positive difference set. For $1 \leq i \leq n$, let $\delta_i = \gamma_i - \gamma_1$.*

Then, for every $t > 0$, the n-tuple $(z_{t+\delta_i})_{1 \leq i \leq n}$ is unbiased, if and only if,

$$\mathcal{F}(f + \varphi_{e_i}) = \sum_{x \in GF(2)^n} (-1)^{f(x)+x_i} = 0 \tag{3}$$

for at least $n-1$ integers i, $1 \leq i \leq n$.

Proof. First, notice that the bit $s_{t-\gamma_1}$ is at tap x_i at time $t + \delta_i$ for each i, $1 \leq i \leq n$. For $1 \leq i \leq n$, let p_i be the probability defined by

$$p_i = \mathrm{Prob}\left(f(x(t+\delta_i)) = 0 \mid x_i(t+\delta_i) = 1\right).$$

The LFSR sequence being balanced, we have

$$\begin{aligned}\mathrm{Prob}(f(x(t)) = 0) = \frac{1}{2} = \frac{1}{2}(&\mathrm{Prob}(f(x(t)) = 0 \mid x_i(t) = 1) \\ &+ \mathrm{Prob}(f(x(t)) = 0 \mid x_i(t) = 0)).\end{aligned}$$

We deduce

$$\begin{aligned} p_i &= \mathrm{Prob}\left(f(x(t+\delta_i)) = 0 \mid x_i(t+\delta_i) = 1\right) \\ &= \mathrm{Prob}\left(f(x(t+\delta_i)) = 1 \mid x_i(t+\delta_i) = 0\right) \end{aligned}$$

$$\begin{aligned} 1 - p_i &= \mathrm{Prob}\left(f(x(t+\delta_i)) = 0 | x_i(t+\delta_i) = 0\right) \\ &= \mathrm{Prob}\left(f(x(t+\delta_i)) = 1 | x_i(t+\delta_i) = 1\right). \end{aligned}$$

Thus, the probability that $f(x(t+\delta_i))$ is equal to a given bit b_i given $x_i(t+\delta_i) = s_{t-\gamma_1} = 0$ is equal to $(1-b_i)(1-p_i)+b_i p_i$, and it is equal to $(1-b_i)p_i+b_i(1-p_i)$ given $x_i(t+\delta_i) = s_{t-\gamma_1} = 1$.

As the choice of the taps is a full positive difference set, two n-tuples of bits input to the filter share at most one bit in common, and their other bits are supposed to be independent. Therefore, the n-tuple $(z_t, z_{t+\delta_2}, \ldots, z_{t+\delta_n})$ is equal to a given n-tuple $(b_1, \ldots, b_n)$ of bits with probability $\frac{1}{2}\prod_{i=1}^{n}((1-b_i)(1-p_i)+b_i p_i) + \frac{1}{2}\prod_{i=1}^{n}(b_i(1-p_i)+p_i(1-b_i))$. In order to have no bias in $(z_t, z_{t+\delta_2}, \ldots, z_{t+\delta_n})$, it is thus necessary and sufficient that the equality $\frac{1}{2}\prod_{i=1}^{n}((1-b_i)(1-p_i)+b_i p_i) + \frac{1}{2}\prod_{i=1}^{n}(b_i(1-p_i)+p_i(1-b_i)) = \frac{1}{2^n}$ holds for all choices of b_i's. This is equivalent to all the p_i's being equal to $\frac{1}{2}$, apart from at most one p_i. This is true if and only if Equation 3 holds for at least $n-1$ integers i, $1 \leq i \leq n$. □

The attack we considered also generalizes the attack against the stream cipher Decim presented by Wu and Preneel in [19] where a bias in the probability that two output bits with a common input bit were equal was taken advantage of. Therefore, the criterion in Proposition 5 thwarts this attack, as it encompasses all the biases arising from the fact that several outputs of the function can share a common input bit.

Remark 2. Notice that the condition stated in Proposition 5 is close to the correlation-immunity of order 1, as introduced in Proposition 1. Indeed, this new criterion allows for at most one unbalanced 1-variable restriction, instead of none.

Definition 1. *We say that a Boolean function satisfying the property in Proposition 5 is* quasi-immune to correlations of order 1*.*

Quasi-immunity of order 1 is not only close to correlation-immunity of order 1, but it is also close to the perfect balancedness definition from Golic. Indeed, it is also a criterion on the filter function only, and a function that is not quasi-immune has a bias, as shown in the proof of Proposition 5, so its output for a random input cannot be random. Moreover, functions satisfying the criterion given by Golic in Theorem 1 are quasi-immune of order 1.

More precisely, quasi-immunity of order 1 is exactly equivalent to the balancedness of the augmented filter function $\bar{h}$ of Anderson in the setting of Proposition 5. Unlike the balancedness of F_{M+1}, the balancedness of $\bar{h}$ is thus easy to check, which makes quasi-immunity a practical criterion to avoid optimal correlation attacks. However, this criterion should be completed to avoid key recovery attack based on a weakness of the filtering function. We will see in the next section that this amounts to bound the bias of the only possible unbalanced 1-variable restriction of a quasi-immune function.

4 Attack Complexity and Quasi-immunity

In this section, we compare different types of attacks targeting filtering functions that are quasi-immune to correlations of order 1, and functions that are not.

4.1 Distinguishing Attack

The scope of this attack is to distinguish the output sequence from a random sequence.

Case of a quasi-immune filtering function. In the probabilistic model, the input sequence is assumed to be random. In this case, if f is perfectly balanced, then the output is also random. Therefore, the output cannot be distinguished from a random sequence.

However, as we have shown, this is not always the case in the deterministic model. On the contrary, in this model, some quasi-immune functions which are not perfectly balanced, might result in balanced augmented functions with a properly chosen feedback polynomial. Recall that a function f that is quasi-immune to correlations of order 1 has at most one restriction e_i, $1 \leq i \leq n$, such that $x_1, \ldots, x_n \mapsto f(x_1, \ldots, x_n) \oplus \varphi_{e_i}(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \oplus x_i$ is unbalanced.

Case of a non quasi-immune filtering function. When a function is not quasi-immune to correlations of order 1, then there exist two unbalanced restrictions e_i and e_j, with two associated probabilities both distinct from $\frac{1}{2}$:

$$p = \text{Prob}\left(f(x(t)) = b_1 \mid x_i(t) = 1\right) \text{ and}$$

$$q = \text{Prob}\left(f(x(t+\gamma)) = b_2 \mid x_j(t+\gamma) = 1\right)$$

Without loss of generality (by exchanging b_i and $\bar{b}_i$ if necessary), we assume $p < \frac{1}{2}$ and $q < \frac{1}{2}$. Then, the output bits pair $(z_t, z_{t+\gamma})$ related to the two inputs $x(t)$ and $x(t+\gamma)$ is equal to (b_1, b_2) or $(\bar{b}_1, \bar{b}_2)$ with probability $pq + (1-p)(1-q) > \frac{1}{2}$. Therefore, in order to distinguish between the output and a random sequence, it is sufficient to consider pairs of output bits distant from one another by γ, and to check that pairs (b_1, b_2) and $(\bar{b}_1, \bar{b}_2)$ appear with probability $pq + (1-p)(1-q)$. Thus, if the filtering function f is not quasi-immune to correlations of order 1, then there always exists a distinguisher between random sequences and keystream output by the filter generator (even when considering the probabilistic filter generator model).

4.2 State Recovery Attack

A standard aim of an attack against an LFSR-based cipher is to retrieve the internal content of the register. This attack takes place necessarily in the deterministic model.

Case of a quasi-immune filtering function. In the case of a quasi-immune function f, if there is one unbalanced restriction e_i, it is possible to guess the internal state of the cipher as the output bit is correlated to the bit with unbalanced restriction. For instance, suppose

$$p = \text{Prob}\left(f(x(t)) = b \mid x_i(t) = 1\right) \neq \tfrac{1}{2},$$

with $p < \frac{1}{2}$ for instance (otherwise exchange b and $\bar{b}$). Then, for each bit in the output, we guess the input bit with probability $1-p$. The complexity of the related attack is $\left(\frac{1}{1-p}\right)^r$.

Remark 3. Even if f is perfectly balanced, it can have unbalanced restrictions, so perfect balancedness is not sufficient to avoid such correlation attacks. Here, we need to choose f and r such that $(\frac{1}{1-p})^r \geq 2^k$ where k is the security parameter.

Case of a non quasi-immune filtering function. Suppose now that the function is not quasi-immune to correlations of order 1. Then, we have:

Proposition 6. *Let (x_i, x_j) be a pair of variables whose relative restrictions are unbalanced, and let*

$$p = \mathit{Prob}\left(f(x(t)) = b_1 \mid x_i(t) = 1\right),$$

$$q = \mathit{Prob}\left(f(x(t+\gamma)) = b_2 \mid x_j(t) = 1\right),$$

with b_1 and b_2 such that $p < \frac{1}{2}$ and $q < \frac{1}{2}$. Then, the nonlinear filter generator with filter f and internal state of length r is vulnerable to a state recovery attack of complexity $\mathcal{O}\left(P(r)\left(1+\frac{pq}{(1-p)(1-q)}\right)^r\right)$, with P a polynomial corresponding to the resolution of a linear system.

The proof is given in Appendix A.

4.3 Building of a Weaker Equivalent Filter Generator

From the attacker side, the first step to attack a filter generator by focusing on the filtering function is to look for an equivalent filter generator with a weaker filtering function. Indeed, correlation-immunity is not an affine invariant, and neither is quasi-immunity. Indeed, the quasi-immunity of the filtering function of a given filter generator does not guarantee the quasi-immunity of the filtering functions of equivalent generators.

We consider an LFSR of length r with feedback polynomial $C(x) = 1 + c_1x + c_2x^2 + \cdots + c_{r-1}x^{r-1} + x^r$. The sequence generated by the LFSR with feedback polynomial C and initial value $[s_{-r}, \ldots, s_{-1}]$ is denoted by $s = (s_t)_{t=-r}^{\infty}$. The filtering function f_0 is an n-variable Boolean function where $0 < n \leq r$. Let $\gamma = (\gamma_i)_{i=1}^{n}$ be an increasing sequence of positive integers such that $\gamma_1 = 1$, and $\gamma_n \leq r$.

We denote by $\widetilde{f_0}$ the r-variable Boolean function constructed from f_0 and $\gamma = (\gamma_i)_{i=1}^{n}$ by adding $r - n$ mute variables. The function $\widetilde{f_0}$ is defined by $\widetilde{f_0}(x_1, \ldots, x_r) = f_0(x_{\gamma_1}, x_{\gamma_2}, \ldots, x_{\gamma_n})$. The keystream sequence $z = (z_t)_{t=0}^{\infty}$ is the output sequence of $\widetilde{f_0}$, *i.e.* $z_t = \widetilde{f_0}\,(s_{t-1}, \ldots, s_{t-r})$, $t \geq 0$. We consider in the following the filter generator $\mathcal{FG}_0 = \left(C, \widetilde{f_0}, [s_{-r}, \ldots, s_{-1}]; z = (z_t)_{t=0}^{\infty}\right)$.

For every $i > 0$, it is possible to construct an equivalent generator $\mathcal{FG}_i$ with the same feedback polynomial and output sequence, but with different initial state and filtering function: $\mathcal{FG}_i = \left(C, \widetilde{f_i}, [s_{-r+i}, \ldots, s_{-1+i}]; z = (z_t)_{t=0}^{\infty}\right)$.

We now show how to construct $\widetilde{f_i}$. Given an LFSR state $[x_1, \ldots, x_r]$, the previous state is computed using the transformation

$$\begin{array}{rccc} A: & \{0,1\}^r & \rightarrow & \{0,1\}^r \\ & x_1, \ldots, x_r & \mapsto & (x_r + c_{r-1}x_1 + c_{r-2}x_2 + \cdots + c_1x_{r-1}, x_1, \ldots, x_{r-1}), \end{array}$$

For every $i \geq 1$, we have: $\widetilde{f_i}(x_1, \ldots, x_r) = \widetilde{f_{i-1}} \circ A(x_1, \ldots, x_r)$. We deduce that $\widetilde{f_i}(x_1, \ldots, x_r) = \widetilde{f_0} \circ A^i(x_1, \ldots, x_r)$, where $A^i(x_1, \ldots, x_r)$ denotes the iteration of i times the transformation A.

Proposition 7. *Consider a filter generator with a balanced and quasi-immune of order* 1 *filtering function f_0. All the functions $\widetilde{f_i}$ are quasi-immune of order* 1 *for every $i \geq 0$ if, and only if, for every $i > 0$, one of the following properties is satisfied:*

1. *the function* $x_1, \ldots, x_r \mapsto \widetilde{f_i} \circ A(x_1, \ldots, x_r) \oplus x_r$ *is balanced,*
2. *the restrictions of* $\widetilde{f_i}$ *following* x_j*, for all* $2 \leq j \leq r$*, are all balanced.*

The proof is given in Appendix B.

Remark 4. Balancedness is invariant under linear transformations. Hence, Condition 1 of Proposition 7 is fulfilled if and only if the function $x_1, \ldots, x_r \mapsto (\widetilde{f_i} \circ A + \varphi_{e_r}) \circ A^{-1}(x_1, \ldots, x_r)$ is balanced, *i.e.* , if and only if $x_1, \ldots, x_r \mapsto \widetilde{f_i}(x_1, \ldots, x_r) \oplus x_1 \oplus c_{r-1}x_2 \oplus \cdots \oplus c_2 x_{r-1} \oplus c_1 x_r$ is balanced.

As we have seen, the quasi-immunity criterion is not affine-invariant, so it should be satisfied not only by the filtering function of a given filter generator, but also by the filtering functions of equivalent generators. Thus, the filtering function f_0 should be chosen such that $\widetilde{f_i}$ is quasi-immune of order 1 for every $i \geq 0$. Note that this requirement is clearly a consequence of taking the linear feedback into consideration, and it is therefore related to the notion of an extended augmented function as mentioned in Section 3.3.

4.4 Summary of Our Results on Attacks Complexity

Recall that if the filtering function of a filter generator is balanced then all the filtering functions $\tilde{f_i}$, $i \geq 0$, of equivalent generators are balanced since the balancedness is an affine invariant. We summarize our complexity attack results by taking into account, given a filter generator, all the filtering functions of equivalent generators.

Proposition 8. *Let* f *be the filtering function of a filter generator, and let* $\tilde{f_i}$*,* $i \geq 0$*, be the filtering functions of the equivalent generators. Assuming that* f *is balanced, we have:*

1. *if* $\tilde{f_i}$ *is quasi-immune and has a unique unbalanced restriction* x_j*, then the filter generator is vulnerable to a state recovery attack that exploits this restriction, with time and space complexity* $\mathcal{O}\left(\left(\frac{1}{\max(p,1-p)}\right)^r\right)$*, where* p *is the probability that the value of the restriction of* $\tilde{f_i}$ *in* x_j *is equal to* 0 *(c.f. subsection 4.2);*
2. *if* f_i *is not quasi-immune, then the filter generator is vulnerable to a straightforward distinguishing attack based on a bias of* $pq + (1-p)(1-q) - \frac{1}{2}$*, with* p *and* q *being the probabilities relative to two distinct unbalanced restrictions of* $\tilde{f_i}$ *(c.f. subsection 4.1);*
3. *if* $\tilde{f_i}$ *is not quasi-immune, then the filter generator is vulnerable to a state recovery attack of time and space complexity* $\mathcal{O}\left(\left(1 + \frac{pq}{(1-p)(1-q)}\right)^r\right)$ *(c.f. subsection 4.2).*

Thus, when designing a filter generator, the filtering function must be chosen quasi-immune of order 1 to avoid distinguishing attacks focusing on the filtering function. Furthermore, the at most unbalanced 1-variable restriction must be chosen such that $\mathcal{O}\left(\left(\frac{1}{\max(p,1-p)}\right)^r\right) \geq 2^k$ where k is the security parameter to avoid key reconstruction attack focusing on the filtering function.

5 Conclusion

In the case of nonlinear filter generators, correlation-based attacks and the criteria to avoid them depend heavily on the considered security model. We have shown that perfect balancedness prevents the optimal correlation attack in the probabilistic model, but that it does not apply to the deterministic model. In the deterministic model, perfect balancedness is equivalent to the absence of bias in the output of the system.

We also extracted a precise criterion on filtering Boolean functions, related to correlation between the output bits as in the optimal correlation attack, based on the fact that input bits at different stages may be correlated in case of nonlinear filter generators. This is a major difference with combiners, and pointing this out clears up the status of correlation-based attacks against nonlinear filter generators. We also provided the complexity of different types of attacks against filtering function that do or do not satisfy this new criterion.

Still, several criteria related to correlation exist, but their relevance is now clear. This should provide a convenient basis for designers. Moreover, we believe that the distinction between two security models is also promising, and new attacks should refer to one model or the other in order to precise their relevance.

References

1. Anderson, R.J.: Searching for the Optimum Correlation Attack. In: Preneel, B. (ed.) Fast Software Encryption. LNCS, vol. 1008, pp. 137–143. Springer, Heidelberg (1995)
2. Biryukov, A., Shamir, A.: Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 1–13. Springer, Heidelberg (2000)
3. Canteaut, A., Filiol, E.: On the influence of the filtering function on the performance of fast correlation attacks on filter generators. In: Proceedings of 23rd Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgique, pp. 299–306 (2002)
4. Canteaut, A., Trabbia, M.: Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 573–588. Springer, Heidelberg (2000)
5. Chepyzhov, V., Johansson, T., Smeets, B.J.M.: A Simple Algorithm for Fast Correlation Attacks on Stream Ciphers. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 181–195. Springer, Heidelberg (2001)
6. Dichtl, M.: On Nonlinear Filter Generators. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 103–106. Springer, Heidelberg (1997)
7. Ding, C., Xiao, G., Shan, W.: The Stability Theory of Stream Ciphers, vol. 561. Springer, Berlin (1991)
8. Dj.Golic, J.: On the Security of Nonlinear Filter Generators. In: Gollmann, D. (ed.) Proceedings of Fast Software Encryption 1996. LNCS, vol. 1039, pp. 173–188. Springer, Heidelberg (1996)
9. Hong, J., Sarkar, P.: New Applications of Time Memory Data Tradeoffs. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 353–372. Springer, Heidelberg (2005)

10. Logachev, O.A.: On Perfectly Balanced Boolean Functions. Cryptology ePrint Archive, Report 2007/022 (2007), http://eprint.iacr.org/
11. Meier, W., Staffelbach, O.: Fast Correlation Attacks on Certain Stream Ciphers. Journal of Cryptology 1(3), 159–176 (1989)
12. Meier, W., Staffelbach, O.: Nonlinearity Criteria for Cryptographic Functions. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 549–562. Springer, Heidelberg (1990)
13. Menezes, A.J., Vanstone, S.A., Van Oorschot, P.C.: Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA (1996)
14. Rueppel, R.A.: Analysis and design of stream ciphers. Springer, New York (1986)
15. Siegenthaler, T.: Correlation-immunity of nonlinear combining functions for cryptographic applications. IEEE Transactions on Information Theory 30(5), 776–780 (1984)
16. Siegenthaler, T.: Cryptanalysts Representation of Nonlinearly Filtered ML-Sequences. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 103–110. Springer, Heidelberg (1986)
17. Siegenthaler, T.: Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Trans. Computers 34(1), 81–85 (1985)
18. Sumarokov, S.N.: Zaprety dvoichnyx funkcii i obratimost' dlya odnogo klassa kodiruyushchix ustrojstv (Defects of Boolean functions and invertibility of a class of coding circuits, in Russian). Obozrenie prikladnoj i promyshlennoj matematiki 1(1), 33–55 (1994)
19. Wu, H., Preneel, B.: Cryptanalysis of the Stream Cipher DECIM. In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 30–40. Springer, Heidelberg (2006)
20. Xiao, G., Massey, J.L.: A spectral characterization of correlation immune combining functions. IEEE Transactions on Information Theory IT-34(3), 569–571 (1988)

A Proof of Proposition 6

Proof. Every bit in the input sequence $(s_t)_{t=0}^{\infty}$ is a linear combination of the initial state bits of the register, that is, in the variables $(s_t)_{t=-r}^{-1}$. Therefore, in order to reconstruct the initial state, one can proceed as follows: first, guess $R \geq r$ bits of the input sequence, write the R equations in the r initial state bits, solve the system to find the initial state, and at last check that the guess is correct by comparing the keystream it generates with the actual keystream. In practice, R is chosen to be equal to r, and, if the system solving leads to multiple solutions, there are two solutions: either we add one (or more) equation(s) by guessing some more input bits, or we drop this system and construct another from r new input bits.

In order to guess R bits of the input sequence, we parse the keystream into pairs of bits distant from one another by γ, and guess the value of the corresponding input bit $x_k(t) = x_j(t+\gamma)$. When the pair belongs to $B = \{(b_1, b_2), (\bar{b}_1, \bar{b}_2)\}$, then we guess the input bit - 0 when (b_1, b_2) is observed, 1 for $(\bar{b}_1, \bar{b}_2)$ - with probability $\frac{(1-p)(1-q)}{pq+(1-p)(1-q)}$.

If the pair belongs to $B' = \{(b_1, \bar{b}_2), (\bar{b}_1, b_2)\}$, then we guess it with probability $\frac{\max(p(1-q),q(1-p))}{p+q-2pq}$. However, it is easy to show that $\frac{\max(p(1-q),q(1-p))}{p+q-2pq} < \frac{(1-p)(1-q)}{pq+(1-p)(1-q)}$, so the R bits we guess are those producing pairs of B.

We notice that knowing the output pair $(z_t, z_{t+\gamma})$ does not impact the probability that the pair $(z_{t+\gamma}, z_{t+2\gamma})$ belongs to B or not, as the bit $z(t+\gamma)$ is the first bit of exactly one pair of bits in B and in B'. Therefore, the probability that a pair of bits is or is not in B does not depend on previous output, and it is equal to $pq + (1-p)(1-q)$. This value being greater than $\frac{1}{2}$, finding such pairs of bits is easy.

Let us now assume that we know R pairs of output bits distant from one another by γ, and that all these pairs belong to B. Then, the success probability of reconstruction is

$$\left(\frac{(1-p)(1-q)}{pq + (1-p)(1-q)}\right)^R .$$

In practice, we have $R = r$, and the reconstruction complexity (both in time and space) is thus $\mathcal{O}(P(r)\left(1 + \frac{pq}{(1-p)(1-q)}\right)^r)$, with P the polynomial corresponding to solving the system to retrieve the r bits of the initial state. □

B Proof of Proposition 7

Proof. For $\widetilde{f}_0$, if the filtering function f_0 fulfils the quasi-immunity criterion, then so does the entire function $\widetilde{f}_0$. Indeed, f_0 is balanced and thus $x_1, \ldots, x_r \mapsto \widetilde{f}_0(x_1, \ldots, x_r) \oplus \varphi_{e_j}(x_1, \ldots, x_r)$ is balanced for every mute variable x_k. Therefore, $\widetilde{f}_0$ is quasi-immune.

Suppose now that $\widetilde{f}_i$ is a r-variable quasi-immune function such that $x_1, \ldots, x_r \mapsto \widetilde{f}_i(x_1, \ldots, x_r) + \varphi_{e_j}(x_1, \ldots, x_r)$ is unbalanced for every j such that $1 \leq j \leq r$, apart for at most one value j_0 of j.

Due to the special form of A, we have:

$$\begin{cases} (\widetilde{f}_i + \varphi_{e_j}) \circ A(x) = \widetilde{f}_{i+1}(x) \oplus x_{j-1} \\ (\widetilde{f}_i + \varphi_{e_1}) \circ A(x) = \widetilde{f}_{i+1}(x) \oplus x_r \oplus c_{r-1}x_1 \oplus c_{r-2}x_2 \oplus \cdots \oplus c_1 x_{r-1} \end{cases}$$

If $j_0 > 1$, then $x_1, \ldots, x_r \mapsto \widetilde{f}_{i+1}(x_1, \ldots, x_r) \oplus x_j$ is balanced for every $1 \leq j \leq r-1$, apart from $j = j_0 - 1$. As $\widetilde{f}_{i+1}$ is quasi-immune if, and only if, it is unbalanced for at most one 1-variable restriction, then it is quasi-immune if, and only if, $x_1, \ldots, x_r \mapsto \widetilde{f}_{i+1}(x_1, \ldots, x_r) \oplus x_r$ is also balanced, which is equivalent to $x_1, \ldots, x_r \mapsto \widetilde{f}_i \circ A(x_1, \ldots, x_r) \oplus x_r$ being balanced.

If $j_0 = 1$, then $x_1, \ldots, x_r \mapsto \widetilde{f}_{i+1}(x_1, \ldots, x_r) \oplus x_j$ is balanced for every $1 \leq j \leq r-1$, so $\widetilde{f}_{i+1}$ is quasi-immune. □