Nostalgia cipher: can filtered LFSRs be secure again?

Nabil Chacal[1*], Pierrick Méaux[2] and AnonymousUntilAcceptance[3]

[1] Université Versailles Saint-Quentin-en-Yvelines, Versailles, France, nabil@murena.io
[2] Luxembourg University, Esch-sur-Alzette, Luxembourg, pierrick.meaux@uni.lu
[3] ANSSI, Paris, France, fisrt_name.name@ssi.gouv.fr

Abstract. Linear Feedback Shift Registers (LFSRs) combined with non linear filtering functions have long been a fundamental design for stream ciphers, offering a well-understood structure that remains easy to analyze. However, the introduction of algebraic attacks in 2003 shifted the focus toward more complex designs, as filtered LFSRs required larger registers to maintain security. While this was seen as a drawback at the time, it is no longer a limiting factor, and emerging cryptographic applications benefit from specialized designs—challenges that filtered LFSRs can effectively address. In this work, we propose a new filtered LFSR design, called Nostalgia, tailored for Hybrid Homomorphic Encryption (HHE). We use a weightwise quadratic function as filtering function, leveraging its efficiency in the HHE setting while ensuring security against classical attacks. We also discuss the parameter selection of our design and its efficiency in this context. By revisiting filtered LFSRs in light of modern security requirements, we aim to renew interest in their potential applications and stimulate further cryptanalysis efforts.

Keywords: LFSR · Stream Cipher · Weightwise quadratic functions · Hybrid Homomorphic Encryption

1 Introduction

1.1 Filtered LFSRs: a simple yet abandoned stream cipher paradigm

Filtered linear feedback shift registers are among the simplest stream cipher designs, relying on the well-understood structure of Linear Feedback Shift Registers (LFSRs) combined with a nonlinear filtering function. Their appeal lies in the deterministic nature of LFSRs, which generate sequences based on recurrence relations over finite fields. These sequences have been extensively studied in sequence theory, particularly in terms of periodicity, linear complexity, and correlation properties (*e.g.* [Gol82]). Maximum-length LFSR sequences (m-sequences) exhibit long periods and uniform statistical distributions, making them fundamental in cryptography and other fields. Additionally, LFSRs are well known for their implementation efficiency, as they require only simple shift operations and XORs, making them highly suited for hardware and lightweight cryptographic applications. By applying a nonlinear filtering function to the LFSR output, filtered LFSRs introduce additional complexity while preserving the efficiency and predictability of the underlying sequence [Sie84]. This simplicity, along with their well-developed theoretical foundation, makes filtered LFSRs a natural subject for security investigations.

However, despite their apparent robustness, filtered LFSRs have largely been abandoned due to the devastating impact of algebraic attacks. While early attacks, such as correlation attacks [Sie85], demonstrated statistical weaknesses in certain designs, it was the advent of algebraic attacks [CM03] and fast algebraic attacks [Cou03] that ultimately led to the paradigm shift away from filtered LFSRs. These attacks exploit the algebraic structure of the filtering function, constructing an overdefined system of equations that can be efficiently solved, leading to the recovery of the internal LFSR state. Critically, security is not determined solely by the nonlinearity or degree of the filtering function but also by its deeper algebraic properties, such as algebraic immunity and the complexity of expressing its output in terms of low-degree equations.

As a result, the cryptographic community moved toward more complex stream cipher designs, such as those incorporating nonlinear feedback shift registers (NFSRs) (*e.g.* Trivium [CP08]) or irregular clocking

*Most of this work was carried out during a master internship from April 2024 to September 2024 co-supervised at ANSSI and Luxembourg university.

mechanisms. However, these designs introduce components whose security justification is often more difficult to formalize, making it less clear whether they truly prevent known attacks or simply increase the complexity of analysis without offering provable resistance. In contrast, filtered LFSRs are based on a well-studied theory, where every component's impact on security can be analyzed in a structured way.

Despite the success of algebraic attacks against traditional filtered LFSRs, no fundamental result rules out the possibility of making them secure—provided that they are properly dimensioned to resist these attacks. By moving away from historically small register sizes and reconsidering filtering functions with improved algebraic properties, filtered LFSRs could once again be viable. This work revisits filtered LFSRs in light of modern security requirements, introducing a new instance designed to resist known attacks and explore its potential for recently developed cryptographic primitives.

1.2 Why revisit filtered LFSRs now?

Filtered LFSRs with significantly larger sizes than those considered in the 2000s are now worth revisiting because these sizes no longer present an efficiency bottleneck. Moreover, in several modern cryptographic contexts, the state size and number of variables have a minimal impact on performance.

Since the introduction of LowMC [ARS+15] in 2015, many advanced cryptographic primitives have benefited from symmetric schemes specifically designed for their intended applications, rather than relying on standardized block or stream ciphers. New symmetric encryption schemes have been developed for various domains, including multiparty computation [ARS+15, AGR+16, AGP+19, GLR+20, GØSW23, DKR+22, KHS+23, APRR24], Fully Homomorphic Encryption (FHE) [ARS+15, CCF+16, MJSC16, DEG+18, MCJS19, CHK+21, HKL+22, AMT22, CHMS22, DGH+23, HMS23, CCH+24], and zero-knowledge proofs [GKR+21, GHR+23, BBC+23]. Additionally, some symmetric primitives have been designed to facilitate masking and enhance resistance against side-channel attacks [MMMS23, GMM+24]. The recent rise of arithmetic-oriented ciphers and specialized symmetric primitives in general suggests that it may be worthwhile to revisit older designs—such as filtered LFSRs—that were abandoned for reasons that may no longer be relevant.

One particularly relevant use case is Hybrid Homomorphic Encryption (HHE) [NLV11], which combines an FHE scheme with a symmetric encryption scheme to significantly reduce both computation and communication costs on the client side in FHE-based protocols. In this context, some of the most efficient results have been achieved using stream ciphers inspired by the filtered LFSR paradigm. For example, FiLIP [MCJS19] achieves homomorphic evaluation with a latency below 3 ms per bit [CDPP22, MPP24], while Margrethe [HMS23] attains a throughput exceeding 1200 bits/s [AGHM24]. These results suggest that, with current techniques for homomorphic evaluation, the simplicity of filtered LFSRs could serve as a strong foundation for designing even more efficient HHE protocols.

1.3 Toward a secure filtered LFSR: contributions and roadmap

In this article, we propose to revisit filtered LFSRs as a viable basis for secure stream ciphers. We consider significantly larger parameter choices than those explored in previous decades, arguing that state-of-the-art attacks do not fundamentally rule out the possibility of achieving security with this primitive. By doing so, we aim to rekindle cryptanalysis efforts, inspire new designs, and encourage further study of this elegant and well-understood theoretical foundation. To achieve this, we first review the state of the art in attacks against filtered LFSRs, leading us to explore the impact of increasing both the LFSR length and the number of variables in the filtering function. As a filtering function, we employ a function from a recent generalization of the Hamming weight bit function [Car22, MO24, CS24, MST24], chosen from a class of functions that are both efficiently computable and possessing desirable cryptographic properties. We also estimate its relevant cryptographic parameters.

In Section 3, we introduce the Nostalgia instance and discuss its implementation. The main parameters of Nostalgia are set to $N = 1024$ for the LFSR state and $n = 128$ for the filtering function. The filtering function itself is a weightwise quadratic function, meaning that for each Hamming weight w of the input, it applies a quadratic function indexed by w. Our design choices prioritize simplicity and compatibility with HHE. The rationale behind these choices is developed in Section 4, where we also analyze the cryptographic properties of the filtering function and the taps' distribution.

A security analysis of Nostalgia is presented in Section 5. We first revisit the main attacks considered in the literature against filtered LFSRs and analyze their complexity to determine the minimal parameter sizes required to withstand them. We show that the chosen parameters of Nostalgia exceed these security thresholds. Additionally, we introduce and examine dedicated attacks that could apply to alternative filtered LFSR designs but do not compromise the security of Nostalgia.

In Section 6, we conduct a preliminary study on the evaluation of Nostalgia in the context of hybrid homomorphic encryption. We focus on the error growth incurred during the evaluation of the filtering function. Under reasonable assumptions, we show that the final noise growth remains quadratic in n, the number of variables in the filtering function. This low noise growth suggests that Nostalgia, or similar designs, could be particularly well-suited for HHE applications.

2 Preliminaries

For readability, we use the notation $+$ instead of $\oplus$ to denote the addition in $\mathbb{F}_2$, and $\sum$ instead of $\bigoplus$. In addition to classic notations, we denote by $[n, m]$ the subset of all integers between n and m: $\{n, n+1, \ldots, m\}$. For a vector $a \in \mathbb{F}_2^n$ we use $\mathsf{w_H}(a)$ to denote its Hamming weight $\mathsf{w_H}(a) = |\{i \in [1, n] \,|\, a_i = 1\}|$. For two vectors a and b in $\mathbb{F}_2^n$ we denote by $\mathsf{d_H}(a, b)$ the Hamming distance between a and b, that is, $\mathsf{d_H}(a, b) = \mathsf{w_H}(a + b)$. For two functions f and g we denote by $\mathsf{d_H}(f, g)$ the Hamming distance between their vectors of values.

2.1 Boolean Functions and cryptographic criteria

We give some preliminaries on Boolean functions and their cryptographic criteria, for a deeper introduction on these notions we refer to the book of Carlet [Car21].

2.1.1 Definitions and criteria

Definition 1 (Boolean Function)**.** A Boolean function f in n variables is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. The set of all Boolean functions in n variables is denoted by $\mathcal{B}_n$.

Definition 2 (Balancedness)**.** A Boolean function $f \in \mathcal{B}_n$ is said to be balanced if $|\{x \in \mathbb{F}_2^n \mid f(x) = 0\}| = |\{x \in \mathbb{F}_2^n \mid f(x) = 1\}| = 2^{n-1}$.

Definition 3 (Nonlinearity)**.** The nonlinearity $\mathsf{NL}(f)$ of a Boolean function $f \in \mathcal{B}_n$, where n is a positive integer, is the minimum Hamming distance (d_H) between f and all the affine functions in $\mathcal{B}_n$: $\mathsf{NL}(f) = \min_{g,\, \mathsf{deg}(g)\leq 1}\{\mathsf{d_H}(f, g)\}$, where $g(x) = a \cdot x + \varepsilon$, $a \in \mathbb{F}_2^n, \varepsilon \in \mathbb{F}_2$ (where $\cdot$ is an inner product in $\mathbb{F}_2^n$, any choice of inner product will give the same value of $\mathsf{NL}(f)$).

Definition 4 (Algebraic Normal Form (ANF), degree and monomials)**.** We call Algebraic Normal Form of a Boolean function f its n-variable polynomial representation over $\mathbb{F}_2$ (*i.e.* belonging to $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$): $f(x_1, \ldots, x_n) = \sum_{I \subseteq [1,n]} a_I (\prod_{i \in I} x_i)$ where $a_I \in \mathbb{F}_2$.

The (algebraic) degree of a non-zero function f is $\mathsf{deg}(f) = \max_{I \subseteq [1,n]}\{|I| \,|\, a_I = 1\}$. If $f = 0$, $\mathsf{deg}(f) = 0$. Each term $\prod_{i \in I} x_i$ is called a monomial.

Definition 5 (Algebraic Immunity [MPC04])**.** The algebraic immunity of a Boolean function $f \in \mathcal{B}_n$, denoted as $\mathsf{AI}(f)$, is defined as: $\mathsf{AI}(f) = \min_{g \neq 0}\{\mathsf{deg}(g) \mid fg = 0 \text{ or } (f + 1)g = 0\}$.

2.1.2 Weightwise quadratic functions

Weightwise quadratic functions are a subfamily of Boolean functions that can be seen as a generalization of symmetric functions.

The n-variable Boolean symmetric functions are those that are constant on each slice $\mathsf{E}_{k,n} = \{x \in \mathbb{F}_2^n \,|\, \mathsf{w_H}(x) = n\}$ for $k \in [0, n]$. This class has been assiduously studied in the context of cryptography, see *e.g.* [Car04, CV05, BP05, SM07, QFLW09, CL11, Méa19, Méa21, CM21]. A basis of the n-variable Boolean symmetric functions is given by the $n + 1$ functions called the slices indicators. For $k \in [0, n]$ the slice indicator $\varphi_{k,n}$ is such that:

$$\forall x \in \mathbb{F}_2^n,\ \varphi_{k,n}(x) = 1 \Leftrightarrow \mathsf{w_H}(x) = k.$$

The generalization from symmetric functions to weightwise quadratic functions comes from considering functions of bounded degree on each slice. It corresponds to the concept of weightwise degree-d function introduced in [GM22] for $d = 1$ and [MO24] for the general case.

Definition 6 (Weightwise degree-d functions ([MO24] Definition 12))**.** Let $n \in \mathbb{N}^*$. An n-variable Boolean function f, written as $f = \sum_{k=0}^{n} f_k \varphi_{k,n}$, is called weightwise degree-d if and only if for each $k \in [0, n]$ f_k coincide with a function of degree at most d over $\mathsf{E}_{k,n}$. The set of weightwise degree-d functions is denoted by $\mathcal{WD}_n^d$.

In this context, symmetric functions are simply weightwise degree-0 functions. The Hidden Weight Bit Function [Bry91] (HWBF) is a famous example of weightwise degree-1 function, which cryptographic properties are studied in [WCST14]. Recently, various generalization of the HWBF have been studied [Car22], various of them being WeightWise Quadratic (WWQ) [MO24, MST24] or weightwise degree-d with a small d [CS24].

2.2 Filtered linear feedback shift registers

A Linear Feedback Shift Register (LFSR) is a finite state machine updated by a linear function of its previous state. Formally,

Definition 7. (Linear Feedback Shift Register). An LFSR of length N over $\mathbb{F}_q$ is a finite state automaton which produces a semi-infinite sequence of elements of $\mathbb{F}_q$, s $= (s_i)_{i\geq 0}$ satisfying a linear recurrence relation of degree N over $\mathbb{F}_q$:

$$s_{t+N} = \sum_{i=1}^{N} c_i s_{t+N-i}.$$

The N coefficients $c_1, \ldots, c_N$ are elements of $\mathbb{F}_q$. They are called the feedback coefficients of the LFSR. We use the notation $(s_i^{(t)})_{0\leq i\leq N-1}$ for $(s_{t+i})_{0\leq i\leq N-1}$. In this article we will only consider binary LFSRs, that is with coefficients over $\mathbb{F}_2$.

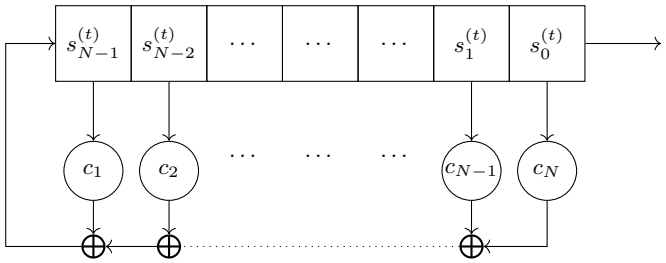

Figure 1: Representation of an LFSR of length N

Definition 8 (Feedback polynomial)**.** An LFSR can be represented by its feedback polynomial defined by:

$$P(X) = 1 + \sum_{i=1}^{N} c_i X^i.$$

Note that the sequence generated by an LFSR is uniquely determined by its feedback polynomial P and its initial state (values of $(s_0, \ldots, s_{N-1})$). A primitive feedback polynomial is sufficient to ensure the LFSR has maximal period.

LFSRs alone cannot generate cryptographically strong output sequences, as they lack any nonlinear component. The most straightforward way to use LFSRs in streamciphers is to combine them with a nonlinear filtering function f. This is typically done by selecting certain positions in the LFSR state, known as taps, and using their values as inputs to the nonlinear function. The output of this function then forms the keystream. Figure 2 provides an overview of a filtered LFSR.

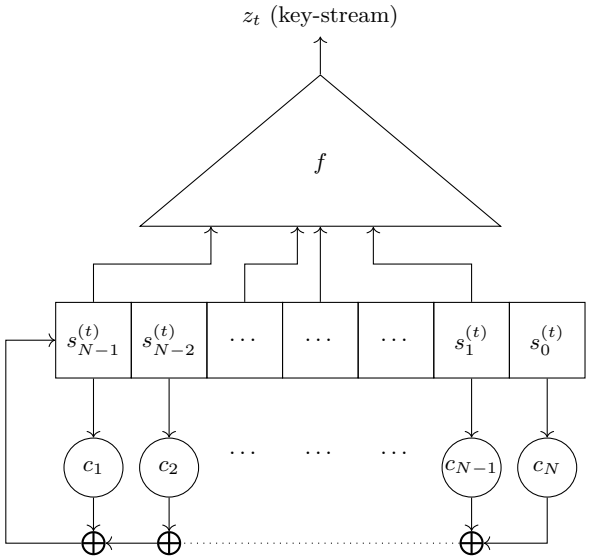

Figure 2: Filter generator

3 The Nostalgia instance

In this section, we present the chosen parameters to instanciate Nostalgia, along with a short description of its public reference implementation. We chose these parameters to resist to existing attacks, which will be detailed further in Section 5.

3.1 Instance

The main parameters of Nostalgia are summarized in Table 1. We set $N = 1024$ and $n = 128$ to achieve a security level of 128 bits, as further detailed in Section 5. The WWQ function f is designed to include a large portion of bits x_i on each slice, but not all. The rationale behind the choice of function and taps is discussed in Section 4.

Table 1: Main parameters of Nostalgia.

LFSR size N	LFSR polynomial	WWQ function size n	WWQ function f
1024	$1 + x^5 + x^{18} + x^{1023} + x^{1024}$	128	$x_k + \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} x_{k-i}x_{k+2i-1}$

Finally, we detail the chosen bit indexes from the LFSR state to the filtering function (*i.e.* the taps). To achieve a good property that will be further detailed, we choose the index 0 and then the first $n-1$ unique numbers that are the sum of the first consecutive primes modulo N. That is, taps $\mathcal{T} = \{0\} \cup \{x \mid x = \sum p_i \mod N\}$ where p_i the are all the first prime numbers, and such that $|\mathcal{T}| = n$. The complete list of tap indexes is given in Appendix A.

3.2 Implementation

We provide two C reference implementations of the Nostalgia instance along with the submission. It consists of two folders with a few files each. The first one, *nostalgia_clean*, considers a straightforward 32-bit implementation.

The filtering part significantly increases the time execution of this version, as it requires accessing 128 bits of the LFSR states, which is encoded in 32-bit values. The second implementation, *nostalgia_faster*, provides a faster time execution at the cost of more memory. In this case, each bit of the LFSR or the filter is encoded in a whole unsigned char, which speeds up the access to each single bit. Moreover, the LFSR state size is increased up to a user-chosen size *ADD_BUFF_SIZE* defined in *defines.h*. This allows to perform almost instant shift operation for the LFSR, as we only need to update the pointer to the initial

sequence and add the last element. We then only need to copy the whole state every *ADD_BUFF_SIZE* cycles, which is chosen by the user.

For all implementations, *defines.h* contains the parameters of Nostalgia and the structure of the filtered LFSR. *util.c/h* files include utility functions, and *nostalgia.c/h* files contain the core implementation of the instance. Finally, a usage example is given in *main.c*. The implementation does not rely on non-standard C libraries and can be simply compiled using gcc.

We ran our implementations on an Intel(R) Core(TM) i7-10510U CPU, compiled with -O3 optimization for speed. The first one achieved throughput of 284 KB/sec, while the second one achieved 1.3 MB/sec. Note that these implementations are not fully optimized and are primarily intended as references.

4 Design rationale

The main design principles behind Nostalgia are simplicity, targeting a clear understanding of its security, and high compatibility with hybrid homomorphic encryption.

The first principle of simplicity is based on the assumption that simpler building blocks can be more thoroughly analyzed both theoretically and across various studies. This approach contrasts with modern constructions, where designs are often complicated by numerous modifications aimed at countering potential attacks observed in other ciphers. Such complexities are introduced under the assumption of 'better safe than sorry,' yet they may ultimately compromise the security of the resulting cipher. The second principle focuses on identifying a function that combines operations efficient to evaluate with an homomorphic encryption scheme. The goal is to ensure that these operations remain efficient across the varying performance metrics of most fully homomorphic encryption schemes and their evolving designs.

4.1 Reviving filtered linear feedback shift registers

Filtered LFSRs are among the simplest stream cipher designs, combining the well-studied structure of linear feedback shift registers with a nonlinear filtering function. Their simplicity stems from the deterministic nature of LFSRs, which generate sequences based on recurrence relations over finite fields. These sequences, particularly maximum-length LFSRs (m-sequences), are well understood in terms of periodicity, linear complexity, and correlation properties (*e.g.* [Gol82]). For instance, selecting a primitive polynomial with a degree equal to the LFSR length ensures that the LFSR generates a maximum-length sequence. Applying a nonlinear filtering function to the LFSR output increases complexity while maintaining efficiency. The simplicity and well-developed theoretical foundation of filtered LFSRs provide a strong starting point for investigating the security of such designs.

Since the early 2000s, numerous attacks have been demonstrated against ciphers following the filtered LFSR paradigm.The (fast) correlation attack [Sie85] exploits the statistical dependence between the LFSR state and the keystream to recover the internal state when the filtering function lacks sufficient nonlinearity. Further advancements in cryptanalysis introduced algebraic attacks [CM03] and fast algebraic attacks [Cou03], revealing that security does not depend solely on the nonlinearity or degree of the filtering function but also on its broader algebraic properties, such as its algebraic immunity and the complexity of expressing its output as a system of low-degree equations. Beyond these, various time-memory tradeoff attacks have been applied to filtered LFSRs, leveraging precomputed tables to drastically reduce the complexity of recovering the internal state. Additionally, other attacks such as guess-and-determine and divide-and-conquer techniques exploit the relatively small internal states used in these designs, allowing attackers to reconstruct the keystream by progressively revealing bits of the LFSR state.

We note that all these attacks remain feasible with relatively low time or data complexity, primarily because the targeted LFSRs and filtering functions are typically of limited size—generally constrained to a few hundred bits for the LFSR and a filtering function acting on only around 20 bits. Due to the strength of these attacks, particularly the fast algebraic ones, filtered LFSRs have largely been abandoned in favor of more complex designs, such as those incorporating nonlinear feedback shift registers (NFSRs) (*e.g.* Trivium [CP08]) or irregular clocking mechanisms. However, our point here is that no known attack fundamentally rules out the possibility of a secure filtered LFSR, provided that it is appropriately dimensioned. By lifting the constraint of small register sizes, it is possible to design filtered LFSRs that

achieve the security levels currently required while offering greater efficiency than more complex stream ciphers for certain applications.

The LFSR is entirely defined by its size, which will be discussed in section 5, and its retroaction polynomial. The choice of this last one induced the period of the output. The only requirement is to ensure that the polynomial is primitive to have maximal period.

4.2 Choice of the filtering function

For the filtering function, we consider the family of weightwise quadratic functions, first studied in [MO24]. These functions are evaluated by first computing the Hamming weight of the input and then applying a quadratic function based on this weight. From a homomorphic evaluation perspective, computing the Hamming weight is relatively inexpensive (*e.g.* [HMR20]), and quadratic functions require only a single level of multiplication. As a result, these functions are well-suited for efficient evaluation across various FHE schemes.

From a symmetric security perspective, the idea of applying different low-degree functions based on the Hamming weight has already been explored in the hidden weight bit function (HWBF) [Bry91]. This function exhibits relatively good cryptographic properties, as analyzed in [WCST14], and recent works have proposed various generalizations [Car22, MO24, CS24, MST24]. In particular, generalizations incorporating addressing and quadratic functions, such as those in [MO24, CS24, MST24], achieve improved nonlinearity compared to the HWBF, which is often considered its Achilles' heel.

For the specific WWQ function used as Nostalgia's filter, we first present its relevant cryptographic parameters for up to $n = 17$ variables. Since the computation of algebraic immunity and nonlinearity is inherently exponential in n we then analyze these parameters for larger values of n. In particular, we establish a bound on the nonlinearity and propose a conjecture regarding the algebraic immunity, as exact computation remains impractical for values suitable for a secure streamcipher.

4.2.1 Parameters for small values of n

The concrete values (determined with Sage) for n up to 17 are presented in Table 2. These results show that the function exhibits high nonlinearity, quasi-optimal degree ($n - 1$ for a balanced function) and quasi-optimal algebraic immunity as it reaches or closely approaches the ceiling of $n/2$.

Table 2: Nonlinearity, degree and algebraic immunity of f for different n values.

n	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NL	4	10	22	48	104	218	456	922	1892	3874	7782	15740	31810	63730
deg	3	4	5	6	6	8	9	10	11	12	13	14	15	16
AI	2	2	3	3	4	5	5	5	6	6	7	7	8	9

4.2.2 Balancedness

Regarding the balancedness of f we can use directly the following result from [MO24]:

Property 1 ([MO24], Proposition 4)**.** *Let $n \in \mathbb{N}^*$, let P_i for $i \in [0, 1]$ be $n + 1$ permutations from the permutation group $\mathbb{S}_n$, the function $g \in \mathcal{B}_n$ is balanced if and only if the weightwise degree-d function h defined by $h_i = g(P_i(x))$ is balanced.*

On each slice the function f takes the value of $x_{P(1)} + \sum_{i=1}^{\lfloor \frac{n}{3} \rfloor} x_{P(2i)}x_{P(2i+1)}$ for a permutation $P \in \mathbb{S}_n$. Since this function is a direct sum of monomials containing a linear part it is balanced (*e.g.* [CM21]), and using Property 1, f is balanced.

4.2.3 Nonlinearity

For larger values of n, we can establish a lower bound on the nonlinearity of f using a result from [MST24]. Specifically, we first recall their theorem (Theorem 1), which bounds the absolute value of the Walsh

transform for any function constructed using quadratic terms in a direct sum over each slice. Then, using the relationship between the Walsh transform and nonlinearity, we derive a lower bound on the nonlinearity of f. Since nonlinearity values can be difficult to interpret directly, we express it in terms of the parameter r, defined by the relation $\mathsf{NL}(f) = 2^{n-1} - 2^r$. The parameter r is a real number belonging to the range $[n/2-1, n-1]$, where $n/2-1$ corresponds to the best possible value (achieved by a bent function, impossible here since the function is balanced), while $n-1$ represents the worst case (corresponding to an affine function).

Theorem 1 (Adapted from [MST24] Theorem 3)**.** *Let $n \geq 0$ be an even integer, let $t \in [0, n/2]$ and let f be a weightwise quadratic function with t quadratic terms in direct sum on each slice. We define $\lambda = t/n$, as well as:*

$$\mu = \mu(\lambda) = \begin{cases} \frac{\lambda+1}{2} + \frac{1}{2}\log_2\left(\frac{\left(-\lambda^2+2\lambda+\lambda\sqrt{\lambda^2-4\lambda+2}\right)^\lambda}{\left(1-\lambda+\sqrt{\lambda^2-4\lambda+2}\right)^{2\lambda-1}}\right) & \text{if } \lambda > \frac{1}{6}, \\ 1-\lambda & \text{if } \lambda \leq \frac{1}{6}. \end{cases}$$

Then, for all $a \in \mathbb{F}_2^n$, the following holds: $|\mathcal{W}_f(a)| \leq 1 + n \cdot 2^{\mu n}$, where $\mathcal{W}_f(a)$ denotes the value of the Walsh transform of f in a.

It allows to derive the following proposition for the chosen filtering function:

Proposition 1. *Let f be the weightwise quadratic function as defined in Section 3, its nonlinearity follows the following bound:*

$$r \leq \log_2\left(\frac{1 + n \cdot 2^{0.537n}}{2}\right),$$

and in particular for $n = 128$:

$$r \leq 74.753,$$

where $r = \log_2(2^{n-1} - \mathsf{NL}(f))$.

Proof. Since f is a weightwise quadratic function with $\lfloor n/3 \rfloor$ quadratic terms in direct sum on each slice we can apply Theorem 1. Taking $\lambda = 42/128$ we obtain $\mu(\lambda) \approx 0.537131725419237$. Using the relation between nonlinearity and Walsh transform:

$$\mathsf{NL}(f) = 2^{n-1} - \frac{1}{2}\max_{a \in \mathbb{F}_2^n} |\mathcal{W}_a(f)|,$$

we get:

$$r = \log_2(2^{n-1} - \mathsf{NL}(f)) = \log_2\left(\frac{\max_{a \in \mathbb{F}_2^n} |\mathcal{W}_a(f)|}{2}\right) \leq \log_2\left(\frac{1 + n \cdot 2^{0.537n}}{2}\right).$$

Replacing μ by 0.537131725419237 and n by 128 gives the particular result.

□

From Proposition 1, we obtain the bound $r \leq 0.537 \cdot n + \log_2(n/2)$, which represents a strong asymptotic value for a family of functions that are computationally efficient. In Section 5, we will see how this bound helps determine the appropriate choice of n to achieve the desired security level.

4.2.4 Algebraic immunity

Regarding algebraic immunity, asymptotic values are known for only a limited number of constructions, and determining it for arbitrary functions is generally considered difficult. Since WWQ functions can be seen as a generalization of the HWBF, whose algebraic immunity is proven to be at least $\lfloor n/3 \rfloor + 1$ in [WCST14], we expect a similar or better trend. Moreover, recalling that most balanced functions have their algebraic immunity between $\lceil n/2 \rceil$ and $n/2$ minus a logarithmic term (see [Did06]), we adopt the following conservative hypothesis for the algebraic immunity of f:

Conjecture 1. *Let $n \in \mathbb{N}^*$ and f be the weightwise quadratic function as defined in Section 3, we conjecture the following bound on its algebraic immunity:*

$$\mathsf{AI}(f) \geq \min(\lfloor n/3 \rfloor, 2 \cdot \sqrt{n}).$$

4.3 Choice of the taps

The choice of taps used as inputs to the filtering function can have a significant impact on the security of the scheme. In particular, one should avoid any patterns or regularities in the placement of the taps. Since the filtering function takes n variables as input and the LFSR consists of N registers, there are $\binom{N}{n}$ possible ways to select them. This number grows rapidly, making it impractical to analyze all possible distributions exhaustively. To assess this regularity, we measure the distances between each tap and the others to determine which distances appear most frequently. For a given tap distribution, the script iterates through all possible pairs of taps and computes the absolute difference between their positions. These distances are then used to update a frequency table, where each entry corresponds to the number of occurrences of a particular distance, ultimately producing a histogram of distances. The more frequently a particular distance occurs, the worse it is for the security of the scheme (see the example in Section 5.3).

Definition 9 (Sums of primes distribution)**.** Let $(p_i)_{0\leq i}$ be the increasing sequence of prime numbers, $N \in \mathbb{N}^*$ and $n \leq N$, we then define the sequence of the increasing sums of prime numbers $(sp_i)_{0\leq i}$ by:

$$\begin{cases} sp_0 = 0 \\ \forall i \in \mathbb{N}^*, sp_i = \sum_{k=1}^{i} p_i \end{cases}$$

Using that sequence, we choose the positions of the taps $(\gamma_i)_{0\leq i\leq n-1}$, by getting the sorted list of the n first distinct values of this sequence sp modulo N.

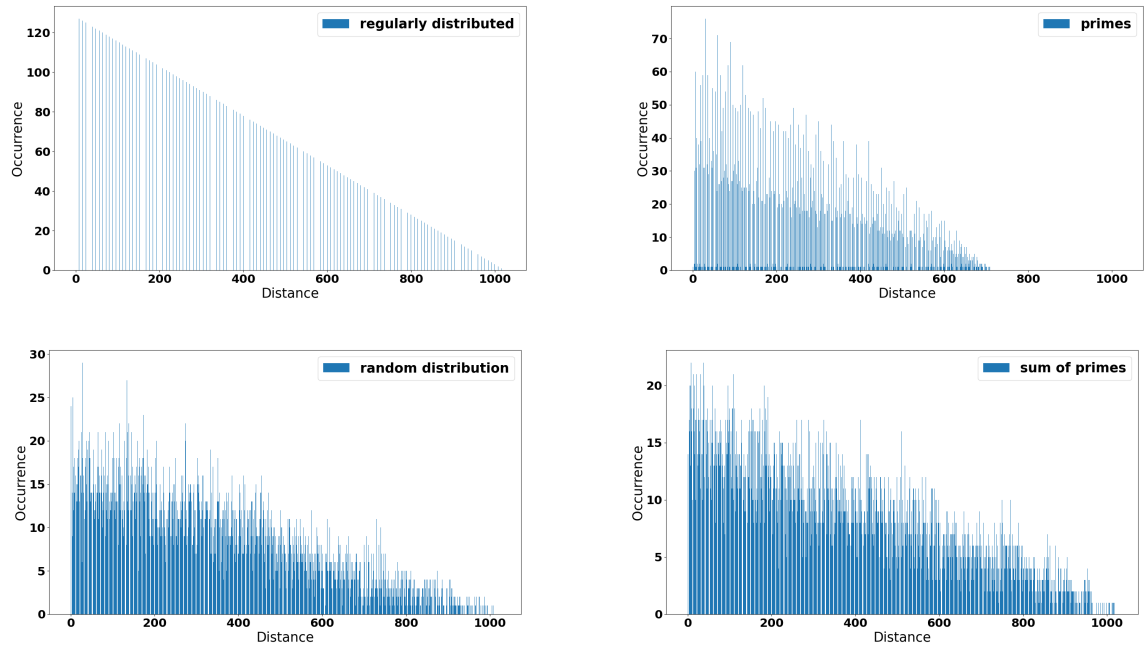

Figure 3: Occurrences of the distances between taps, for various distributions with $N = 1024$ and $n = 128$.

Figure 3 illustrates the distribution of distances between two taps for four different tap placement strategies. Figure 3.a shows the worst possible distribution, which is a regular placement. In this case, the most frequent distance, 8 positions, appears up to 128 times. Figure 3.b presents the distribution when taps are placed at the indices corresponding to the first 128 prime numbers. While this distribution is an improvement, it still exhibits a peak with more than 70 occurrences. Additionally, odd distances appear at most once. Figure 3.c provides an example of a random tap placement, serving as a baseline for comparison. Finally, Figure 3.d shows the distribution obtained using the sums of primes method, which achieves the best balance among all tested approaches, minimizing repetition in distances and providing a more uniform spread.

4.4 Alternative designs

We explore two possible modifications to the design of Nostalgia: the inclusion of an Initialization Vector (IV) and the addition of a whitening step.

Since hybrid homomorphic encryption does not require the use of IVs, no initialization process is specified for Nostalgia. However, if needed, an initialization step should be designed in such a way that the whole internal state depends non linearly of the whole initialization value and the key. One possible way to incorporate an α-bit IV into the state is to run the cipher at least $2n + \alpha$ times, feeding the state not with the LFSR output but with the filtering function output, XORed with the corresponding IV bit during the first α steps. The security analysis of this initialization method is beyond the scope of this article and remains an open research direction.

The linear nature of the LFSR can be altered by XORing a random vector with the input of f at each clock cycle. This process, known as whitening, randomizes the input of f. It requires either n bits of randomness per output bit or the use of a pseudorandom generator for efficiency. The addition of a whitening step in a stream cipher has already been implemented in FiLIP [MCJS19], as it incurs a negligible cost in the context of HHE. The main drawback of this alternative design is that it complicates the analysis of the filtered LFSR, whose simplicity is a key advantage.

5 Security analysis

In this section, we analyze the security of Nostalgia by applying known cryptanalysis. We focus on key recovery attacks in the known-plaintext model, where the adversary has access to the keystream. For a (bit)-security parameter λ, we set a time complexity limit of 2^λ basic operations and a data complexity limit of $2^{\lambda/2}$ keystream bits. The parameters of Nostalgia are chosen to target $\lambda = 128$.

In section 5.1 we recall the main attacks known on filtered LFSR and relevant cryptographic parameters for the filtering function. We argue that, differently from the more standards designs of block ciphers, we cannot study the security and its margin of the cipher by increasing or decreasing the number of rounds. The analog in our case consists in studying variations of the design. We thus analyze dedicated attacks on variants of the scheme and explain how the chosen instance mitigates these attacks. In section 5.2, we show how a simpler filtered function than f (from the same family) leads to an insecure instance. Then, we provide in Section 5.3 an instance where the choice of taps leads to an attack.

5.1 Filtered LFSR and secure parameters

Based on the book of Carlet [Car21] we survey the different attacks known to apply to filtered LFSR, and then determine the sufficient parameters a function should have to be out of reach for these attacks.

5.1.1 Algebraic attacks

Algebraic attacks and fast algebraic attacks were introduced on filtered LFSRs by Courtois and Meyer in [CM03] and fast algebraic attacks (respectively by Courtois in [Cou03]), leading to some of the most efficient attacks on these stream ciphers. The main idea behind algebraic attacks, as defined in [CM03], is to construct an overdefined system of equations where the unknowns correspond to the initial state of the LFSR. This system is then solved using Gaussian elimination. The core principle is to find a nonzero function g such that both g and $h = gf$ (where f is the filtering function) have a low algebraic degree, enabling the derivation of multiple equations of small degree d. The resulting degree-d algebraic system is then solved—either through linearization, using Gröbner basis methods (*e.g.* [Fau99, Fau02]), or SAT solvers. Notably, linearization is the only method for which complexity estimates are straightforward. In practice, the degree of g is at least $\mathsf{AI}(f)$, the algebraic immunity of f, and g is chosen as a nonzero annihilator of either f or $f + 1$ with minimal degree. This allows the adversary to derive a system of equations from the keystream, containing monomials of degree up to $\mathsf{AI}(f)$ in the key bit variables. After linearization, the adversary obtains a system of equations in $D^N_{\mathsf{AI}(f)} = \sum_{i=0}^{\mathsf{AI}(f)} \binom{N}{i}$ variables. Consequently, the time complexity

of the algebraic attack is $\mathcal{O}((D^N_{\mathsf{AI}(f)})^\omega)$, where ω is the complexity parameter for solving a linear system $(2 \leq \omega \leq 3)$.

Fast algebraic attacks [Cou03, HR04] are variant of standard algebraic attacks. Still relying on the relation $gf = h$, their objective is to find and use functions g of low algebraic degree e (possibly smaller than $\mathsf{AI}(f)$) and h of low but potentially higher degree d. The attacker then reduces the degree of the resulting equations by performing an offline elimination of monomials of degree greater than e, leveraging the feedback properties of the LFSR. This process requires multiple equations to generate each equation of degree at most e. Following the analysis in [ACG+06], the time complexity of fast algebraic attacks is given by:

$$\mathcal{O}(D^N_d \log_2^2(D^N_d) + E^N_e D^N_d \log_2(D^N_d) + (E^N_e)^\omega), \text{where } D^N_d = \sum_{i=0}^{d} \binom{N}{i}, \text{and } E^N_e = \sum_{i=0}^{e} \binom{N}{i}.$$

For simplicity, we use the facts that $e = \mathsf{deg}(g) \geq 1$, implying $E^N_e \geq N$, and that $d = \mathsf{deg}(fg) \geq \mathsf{AI}(f)$ when $0 < \mathsf{deg}(g) < \mathsf{AI}(f)$, which ensures $D^N_d \geq D^N_{\mathsf{AI}(f)}$. This allows us to bound the complexity of both the algebraic attack and the fast algebraic attack in terms of the algebraic immunity. For a security parameter λ, the function f must satisfy the following requirements:

$$\mathcal{O}((D^N_{\mathsf{AI}(f)})^\omega) > 2^\lambda \text{ and } \mathcal{O}(D^N_{\mathsf{AI}(f)} \log_2^2(D^N_{\mathsf{AI}(f)}) + N D^N_{\mathsf{AI}(f)} \log_2(D^N_{\mathsf{AI}(f)}) + N^\omega) > 2^\lambda.$$

Figure 4 shows the complexity of the algebraic attack (in blue) and the fast algebraic attack (in orange) on a logarithmic scale. Both complexities depend solely on the algebraic immunity, with N fixed at 1024 and a conservative $\omega = 2$ used in the calculations.

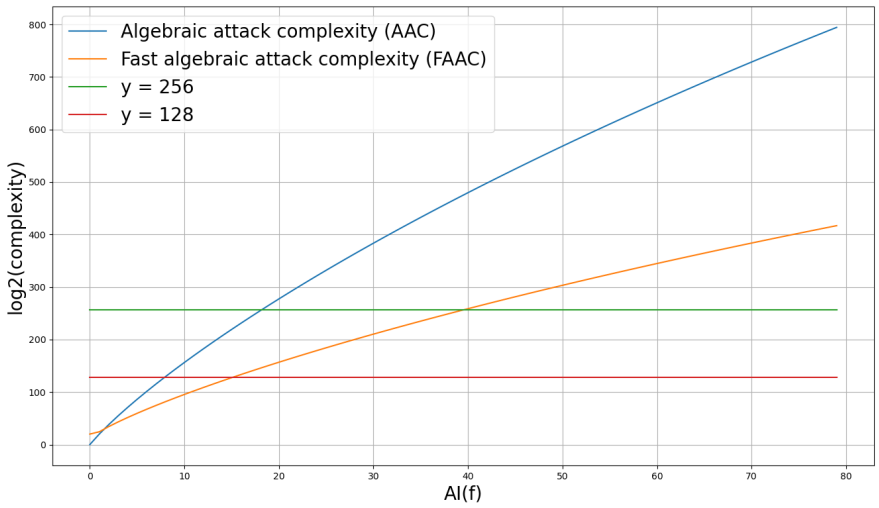

Figure 4: *Logarithmic complexities of the algebraic and fast algebraic attacks depending on the algebraic immunity, with* $N = 1024$.

Figure 4 shows that for $N = 1024$, a function with algebraic immunity of at least 16 is sufficiently resistant to (fast) algebraic attacks. From Section 4.2, we know that the 17-variable version of f already has an algebraic immunity of 9. Moreover, assuming Conjecture 1, f has an AI of at least 23.

5.1.2 Correlation attacks

(Fast) Correlation attacks cover a wide range of different attacks focusing on the correlation between the input (here the output of the LFSR) and the output of the filtering function. It was introduced in [Sie85] for the combiner model and later improved and generalized to any LFSR-based generators [MS88, CS91]. A pseudorandom sequence s generated by the cipher can be viewed as a noisy version of a linear sequence σ,

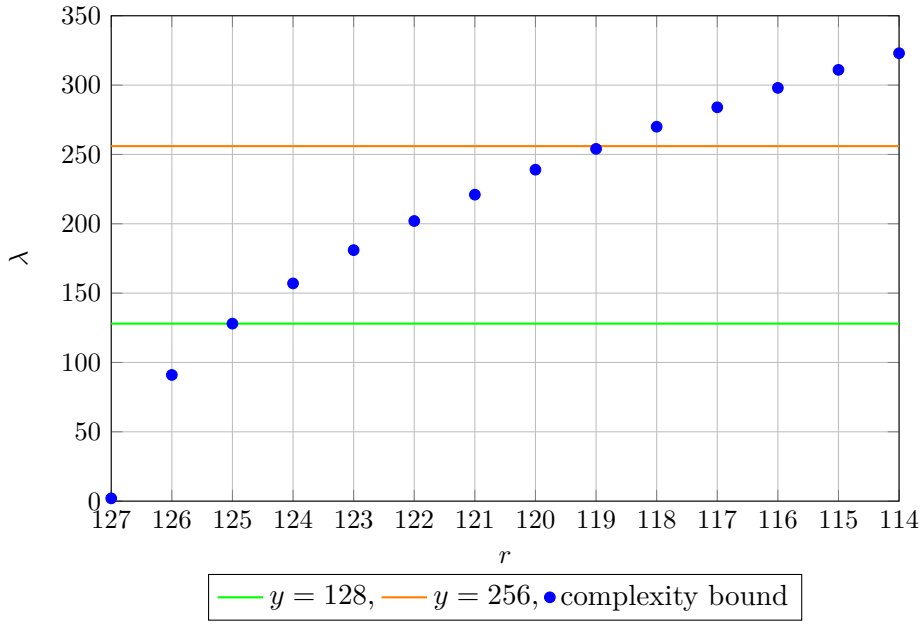

Figure 5: Correlation attack time complexity in function of r for $n = 128$, $N = 1024$

enabling error-correction methods to recover σ. This involves treating s as being transmitted over a noisy channel, where the probability of errors depends on the Hamming distance between the filtering function f and the affine approximation ℓ. By correcting these errors, the attacker can approximate the linear sequence σ, facilitating the recovery of the cipher's internal state. There are multiple decoding algorithms, as in [Car21] we consider the approach of [MS88, CS91] using parity check polynomials of bounded degree and at most $t \geq 3$ coefficients. From [CT00], we obtain the following complexities for this attack on a filtered LFSR:

$$D = \left(2^{\frac{N}{t-1}}\left(\frac{1}{2\varepsilon}\right)^{\frac{2(t-2)}{t-1}}\right), \quad P = \frac{D^{t-2}}{(t-2)!}, \quad \text{and } T = 2^{\frac{N}{t-1}}\left(\frac{1}{2\varepsilon}\right)^{\frac{2t(t-2)}{t-1}},$$

where D is the amount of data, $\varepsilon = \frac{2^{n-1}-\mathsf{NL}(f)}{2^n}$, P is the time complexity of the pre-computation step and T the time complexity of the online phase of the attack.

We study the value of the nonlinearity sufficient to get a time complexity T higher than 2^λ with this analysis. For simplicity we use the parameter r defined as $r = \log(2^{n-1} - \mathsf{NL}(f))$, which takes values from $n/2-1$ when f is bent to $n-1$ when the function is affine. In Figure 5 we represent the minimal complexity obtained for $n = 128$, $N = 1024$, varying r and $t \in [3, 1024]$. For these sizes, an r relatively close to $n-1$ is sufficient from the analysis of [CT00], such as $r = n-4$ from the figure.

We analyze the value of nonlinearity sufficient to achieve a time complexity T greater than 2^λ in this context. For simplicity, we use the parameter r, defined as $r = \log(2^{n-1} - \mathsf{NL}(f))$, which ranges from $n/2-1$ when f is bent to $n-1$ when the function is affine. In Figure 5, we represent the minimal complexity obtained for $n = 128$ and $N = 1024$, varying r and $t \in [3, 1024]$. For these parameters, an r value relatively close to $n-1$ is sufficient from the analysis in [CT00]. For instance, from the figure, $r = n-4$ is adequate.

Since other decoding techniques are possible, or alternative approaches to solving the Learning Parity with Noise (LPN) problem exist (*e.g.* [BV16]), we prefer functions with an r value that grows as a fraction of n. In particular, the filtering function considered here, achieving $r \leq 0.537 \cdot n + \log_2(n/2)$, guarantees a bias of 2^{-53} for $n = 128$. This bias already requires 2^{106} keystreams to distinguish the output from a truly random binary string.

5.1.3 Other attacks

We also considered other attacks on filtered LFSRs, as detailed in [Car21], but they result in higher complexities than the attack described above in this section. Below, we summarize the considered attacks

and refer to *e.g.* [Car21] for full details. The linearization attack always has a higher complexity than the algebraic attack since, for any Boolean function, $\mathsf{AI}(f) \leq \mathsf{deg}(f)$. The Rønjom-Helleseth attack [RH07], later improved in *e.g.* [GRHH11, Røn17], has a complexity of $\sum_{i=1}^{\mathsf{deg}(f)} \binom{N}{i}$. Since N is significantly larger than 128 for any filtering function with a degree close to $n = 128$, as considered in this work, the resulting complexity is far greater than 2^{128}. Time/memory/data trade-off attacks can also be applied to stream ciphers [BS00]. To prevent these attacks, it is recommended to select an LFSR length N at least twice the security parameter [BS00, HS05]. This requirement is met in the design of Nostalgia, where $N = 1024$ for a target security level of 128 bits. Recent attacks based on monomial mappings, such as those presented in [CR16], leverage the univariate representation of the LFSR, over $\mathbb{F}_2^N = \mathbb{F}_2^{1024}$. However, since the filtering function is described in its multivariate representation, we assume that a transformation, such as the discrete Fourier transform, would be required to obtain the polynomial coefficients in the univariate representation. This transformation would necessitate a precomputation step of complexity at least $2^n \log_2(2^n) = n2^n > 2^{128}$ in our context. This situation differs from previously studied filtered LFSRs, where both N and n were significantly smaller for the same security level.

5.2 Specific weightwise quadratic function weakness

The choice of the filtering function within the weightwise quadratic family has a significant impact on the overall security of the cipher. In particular, the number of second-degree terms and their periodicity within the index range can be an asset. To illustrate the importance of selecting a well-designed WWQ function, we present a practical attack against the WWQ function defined by $f_i(x) = x_i + x_{i+1}x_{i+2}$, where i represents the Hamming weight of the 128-bits input. For clarity, probabilities have been simplified in this exposition.

The probability that $f_i(x) = x_i$ is $3/4$, since the product of two bits is zero with probability $3/4$ Moreover, the probability of $i = 64$ is $\binom{128}{64}/2^{128} \approx 0.070$ while the probability of $i = 63$ is approximately 0.069. Let α denote the distance between the 64^{th} and the 63^{rd} taps. Suppose we have two inputs of weight 64 and 63, respectively, occurring at iterations t and $t + \alpha$ (with a combined probability of approximately 0.005). Then, f takes the same value on both inputs with probability $9/16$, since they correspond to x_{64} in the first iteration and x_{63} in the second. It follows that the probability of two bits separated by α positions in the output stream being equal is approximately $0.502 > 0.5$. This provides a distinguisher on the keystream.

In fact, this attack strategy is mitigated by the correlation attack analysis in Section 5.1.2. The nonlinearity of f represents the distance between f and its best affine approximation; hence, it already accounts for the fact that $f_{64}(x) = x_{64} + x_{65}x_{66}$ correlates with x_{64} on $3/4$ of the inputs in $\mathbb{F}_2^n$. Since f_{64} is only used on $\mathsf{E}_{64,128}$, the correlation may not be exactly $3/4$ on this subset. Moreover, the previous analysis does not consider that the correlation might be lower than $1/2$ on some other slices. Conversely, the nonlinearity already incorporates this information. The WWQ function $f_i(x) = x_i + x_{i+1}x_{i+2}$ corresponds to a function with $t = 1$ under Theorem 1. Applying the same reasoning as in Proposition 1 from Section 4.1, we obtain:

$$r \leq \log_2\left(\frac{1 + 128 \cdot 2^{127}}{2}\right) \approx 133,$$

where $r = \log_2(2^{n-1} - \mathsf{NL}(f))$. This bound on r provides no guarantee against correlation attacks, unlike the case studied in Proposition 1. For security, a value close to 64 is desirable, whereas a value close to 128 is unfavorable. Therefore, these results discourage the use of such function for the given choices of n and N.

5.3 Cryptanalysis on regular tap distribution

The most straightforward way to select the taps is to place them at regular intervals of distance δ, smaller than N/n, between consecutive positions. In this case, denoting γ_0 as the first index, the list of taps is defined by $\forall i \in [1, n-1], \quad \gamma_i = \gamma_0 + i \cdot \delta$. We show that such a distribution, when paired with certain weightwise degree-d functions, leads to a simple attack.

Let $f \in \mathcal{WD}_n^d$ be defined on each slice $\mathsf{E}_{k,n}$ by $f_k = g(x_k, \ldots, x_{k+r})$, where the indices are taken modulo n. Such functions correspond to cyclic weightwise degree-d functions as defined in [MO24], with some also analyzed in [CS24].

We denote by $x^{(t)}$ the n-bit state given by the taps at time t. The keystream bits are given by $z_t = f(x^{(t)})$ and $z_{t+\delta} = f(x^{(t+\delta)})$, where $z_{t+\delta}$ is obtained δ shifts of the LFSR later. Using the properties of the LFSR, for all $i \in [1, n-1]$, we have $x_i^{(t+\delta)} = x_{i+1}^{(t)}$. Accordingly, the Hamming weights of $x^{(t)}$ and $x^{(t+\delta)}$ differ by at most one.

Now, suppose that $2 \leq \mathsf{w_H}(x^{(t)}) = j \leq n - r$. If $x_1^{(t)} = 1$ and $x_n^{(t+\delta)} = 0$, then $\mathsf{w_H}(x^{(t)}) = j$ and $\mathsf{w_H}(x^{(t+\delta)}) = j - 1$, which gives:

$$z_t = f(x^{(t)}) = g(x_j^{(t)}, \ldots, x_{j+r}^{(t)}),$$

and

$$z_{t+\delta} = f(x^{(t+\delta)}) = g(x_{j-1}^{(t+\delta)}, \ldots, x_{j-1+r}^{(t+\delta)}) = g(x_j^{(t)}, \ldots, x_{j+r}^{(t)}) = z_t.$$

Thus, when $x^{(t)}$ has a Hamming weight in the range $[2, n-r]$, we have:

$$(x_1^{(t)}, x_n^{(t+\delta)}) = (1, 0) \Longrightarrow z_t = z_{t+\delta}.$$

Conversely, $z_t \neq z_{t+\delta}$ implies $(x_1^{(t)}, x_n^{(t+\delta)}) \neq (1, 0)$, which is satisfied if the quadratic equation $(x_n^{(t+\delta)} + 1)x_1^{(t)} = 0$ holds. Therefore, each time z_t and $z_{t+\delta}$ differ (which occurs with probability 0.5, as otherwise there would be a distinguisher on the keystream), we obtain quadratic equations in the key bits, provided the Hamming weight of $x^{(t)}$ belongs to a specific range. The probability that the weight of $x^{(t)}$ falls within $[2, n-r]$ is

$$p_r = 1 - \frac{1}{2^n} \sum_{i \in \{0,1\} \cup [n-r+1, n]} \binom{n}{i}.$$

For simplicity, we assume that each quadratic equation obtained from cases where $z_t \neq z_{t+\delta}$ is valid with probability p_r (this is an approximation, as the values of $x_1^{(t)}$ and $x_n^{(t+\delta)}$ are not entirely independent of the weight of $x^{(t)}$). Accordingly, an adversary can obtain degree-2 equations, linearize them, and solve the resulting linear system with probability

$$p = (p_r)^{1+N+\binom{N}{2}}.$$

For a cyclic WWQ function with small r, such as the example in Section 5.2 where $r = 2$, the attack is efficient since p^r is very close to 1. Conversely, this attack strategy is ineffective when $r \geq n/2$, as in this case, p_r is already below $1/2$.

6 Towards efficient homomorphic evaluation

As a first step towards the use of Nostalgia in the HHE framework, we conduct a detailed theoretical analysis to assess the behavior of the homomorphic evaluation, providing a strong foundation for future implementations. The choice of FHE scheme significantly impacts the set of efficient operations available, as some schemes support efficient small-domain bootstrapping, others have inefficient bootstrapping, and some offer only quasilinear growth for multiplication. Given this variability, we adopt a Boolean circuit-based approach to evaluate Nostalgia. There are two key reasons for this choice. First, if the noise in ciphertexts evaluated using this method remains sufficiently low, it indicates that relatively few complex operations are involved, which is generally a good starting point for efficient transciphering. Second, for each FHE scheme, there is considerable room for optimization beyond the circuit-based approach. Therefore, this approach serves as an initial reference point for expected efficiency.

First, we explain how an LFSR can be efficiently evaluated homomorphically, followed by a more in-depth analysis of the evaluation of the filtering function. To achieve this, we specify certain assumptions about the FHE scheme used, which allow us to derive a precise expression for noise evolution.

6.1 Evaluation of the LFSR

For each keystream bit produced, the new element in the LFSR is computed using XOR operations, which typically correspond to additions of homomorphic ciphertexts. This operation is generally considered to have the lowest impact on noise, as it only doubles the variance. A common oversimplification suggests that

'homomorphic addition is cheap, while multiplication is costly,' leading to the assumption that maintaining N homomorphic ciphertexts for the state and iterating on them would result in an efficient evaluation. However, in reality, doubling the variance with each iteration (or even every N iterations) causes noise accumulation that quickly becomes too significant, preventing the use of small FHE parameters. This is particularly critical given that we typically consider encrypting up to $2^{\lambda/2}$ bits—equivalent to 2^{64} for the security level we consider—which is far larger than N. The impact of addition on noise growth has been observed in practice, for example, in the homomorphic evaluation of LowMC using the HElib library. In this case, more layers than the number of multiplicative layers were necessary to perform the evaluation.

Accordingly, to prevent the noise parameter from depending on the number of encrypted bits, we can track which elements of the initial key contribute to each element of the current LFSR state. This is achieved by maintaining N lists, each containing at most N elements, which are updated at every LFSR iteration. For the elements entering the filtering function, an homomorphic addition is performed on each of them, resulting in ciphertexts with a noise level of at most N times the initial variance, with an average of $N/2$. This noise can be further reduced by employing techniques such as strategically combining additions and subtractions while preserving correctness, as observed in [MJSC16], Section 4.2.

This evaluation is well suited for FHE schemes often referred to as second-generation schemes. For those with quasi-additive error growth in multiplication, the norm of the plaintext appearing in the noise formula contributes to a polynomial increase in noise as the number of multiplications grows, when the plaintext norm exceeds 1. As noted in [Méa17], Section 4.2, this can lead to an error growth of $\mathcal{O}(N^d)$ when evaluating a filtered LFSR with a Boolean function of degree d, which becomes prohibitive for standard size of N. To address this, for FHE schemes exhibiting quasi-additive multiplicative error growth, we consider applying gate bootstrapping to the LFSR update. Gate bootstrapping enables us to obtain fresh ciphertexts with binary plaintexts and low noise before evaluating the filtering function. Gate or circuit bootstrapping is among the most efficient operations for FHE schemes with quasi-additive multiplicative error growth, as demonstrated in [DM15, CGGI16, CGGI20, LMK+23, WWL+24].

6.2 Evaluation of the WWQ function

To estimate the noise in an homomorphic ciphertext obtained through transciphering with Nostalgia, we analyze the variance of the noise distribution in the ciphertexts resulting from the evaluation of the function f. To do so, we assume that each ciphertext encoding a bit of the LFSR state encrypts either 0 or 1 and shares the same variance parameter. We then use an assumption on the noise propagation of fundamental operation:addition, subtraction, and multiplication, which are sufficient to derive the final noise parameter introduced by evaluating f.

We begin with the noise parameters rules encompassing various schemes in the line of GSW and TFHE ([GSW13, DM15, CGGI16]).

Assumption 1. *Noise evolution rules, adapted from [HMR20] Proposition 2 Let C_i an FHE encryption of the plaintext $\mu_i \in \{-1, 0, 1\}$ with $i \in [1, 3]$ with error noise variance V_i. We denote by V_+ the error variance of $C_1 + C_2$, $V_\times$ of $C_1 \times C_2$ and V_M of $\mathsf{MUX}(C_1, C_2, C_3) = C_1(C_3 - C_2) + C_2$.*

$$V_+ \leq V_1 + V_2, \quad V_\times \leq c_3 V_1 + |\mu_1|(c_4 + V_2), \quad \text{and } V_M \leq \max(V_2, V_3) + c_3 V_1 + c_4,$$

where c_3 and c_4 are constants depending on the FHE scheme and its parameters.

Next, we recall the error growth for an input-oblivious circuit that determines whether the Hamming weight of an input equals k. To illustrate this, Figure 6 presents the example from [HMR20] for $\mathbb{F}_2^7$, along with a summary of the main results concerning the homomorphic evaluation of such circuits.

Property 2. *Circuit correctness and homomorphic noise evaluation, adapted from [HMR20] Propositions 3 and 5. Let $n \in \mathbb{N}^*$, there exist an input-oblivious Boolean circuit of n inputs $x_1, \ldots, x_n$ and $n + 1$ output $y_0, \ldots, y_n$ such that for $i \in [0, n]$:*

$$y_i = 1 \Leftrightarrow \mathsf{w_H}(x_1, \ldots, x_n) = i.$$

Additionally, based on Assumption 1, such circuit can be evaluated with ciphertexts encrypting y_i with a variance V' following $V' \leq (n-1)(c_3 \cdot V + c_4) + V$, where V denotes the variance involved in the initial ciphertexts (encrypting x_i).

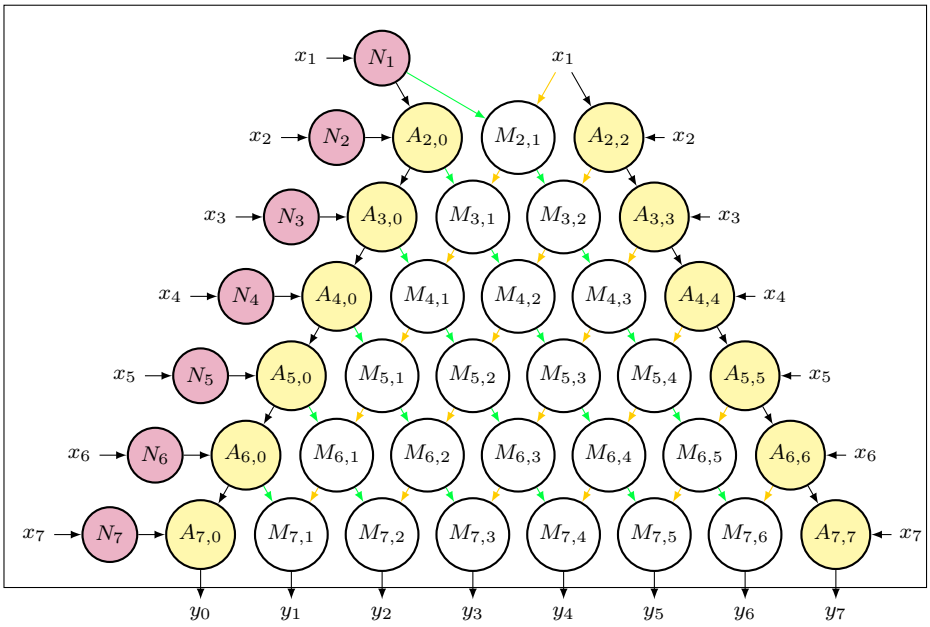

Figure 6: Boolean circuit to determine the Hamming weight of a binary vector of size 7, adapted from [HMR20]. The control bit of each level of MUX is the input x_i on the left side, the left arrow ($\rightarrow$) in input of a MUX corresponds to the input for a control bit equal to 0 and the right arrow ($\rightarrow$) for the value at 1.

Finally, it allows us to provide a bound for the evaluation of f:

Theorem 2 (Variance parameter involved in the homomorphic evaluation of f)**.** *Let $n \in \mathbb{N}^*$ and for $i \in [1, n]$ $\mathbf{C}_i$ be an homomorphic encryption of x_i satisfying Assumption 1 with variance V. An homomorphic encryption $\mathbf{C}_f$ of $f(x_1, \ldots, x_n)$ can be obtained with variance V_f such that:*

$$V_f \leq n \cdot \Big(c_3((n-1)(c_3V + c_4) + V) + c_4 + \lfloor \frac{n}{3} \rfloor (c_3 + 1)V + c_4) + V\Big).$$

Proof. To evaluate f, we first compute the circuit considered in Property 2 on the ciphertexts $\mathbf{C}_i$ for $i \in [1, n]$, obtaining $n + 1$ ciphertexts that encode the Hamming weight of the input. Specifically, each ciphertext $\mathbf{C}'_i$ encrypts 1 if the input has Hamming weight i, and encrypts 0 otherwise (*i.e.* it corresponds to the value of $\varphi_{i,n}$). According to Property 2, the variance V' of each ciphertext at this stage satisfies the bound $V' \leq (n-1)(c_3V + c_4) + V$.

Next, we consider the evaluation of each quadratic function f_k. This process involves $\lfloor \frac{n}{3} \rfloor$ products of two initial ciphertexts $\mathbf{C}_i$. Each product results in a variance bounded by $V_\times \leq c_3V_1 + |\mu_1|(c_4 + V_2)$, which simplifies to $(c_3 + 1)V + c_4$ in this context. The sum of these $\lfloor \frac{n}{3} \rfloor$ products, along with the ciphertext corresponding to x_k, produces the ciphertext $\mathbf{C}''_k$, whose variance is bounded by $V'' \leq \lfloor \frac{n}{3} \rfloor ((c_3+1)V + c_4) + V$.

Finally, the function f is obtained as the sum (over n) of the products of $\varphi_{k,n}$ with f_k. This result can be computed by evaluating $\sum_{k=0}^{n} \mathbf{C}'_k \times \mathbf{C}''_k$, where both the sum and the products refer to homomorphic addition and multiplication. Since $f(0_n) = 0$, we can omit the corresponding term in the evaluation. Indeed, if the Hamming weight is zero, all ciphertexts $\mathbf{C}'_i$ for $i \in [1, n]$ encrypt 0, ensuring that the final sum also encrypts 0. Conversely, if the weight is nonzero, $\mathbf{C}'_0$ encrypts 0 and thus does not affect the sum. Accordingly, each product contributes to a variance bounded by $c_3V' + c_4 + V''$, where we use $\mathbf{C}'_i$ as the first operand in the homomorphic multiplication to take advantage of its norm being bounded by 1. The final sum produces $\mathbf{C}_f$ with variance V_f, which satisfies the bound:

$$V_f \leq n(c_3V' + c_4 + V'') \leq n\Big(c_3((n-1)(c_3V + c_4) + V) + c_4 + \lfloor \frac{n}{3} \rfloor ((c_3 + 1)V + c_4) + V\Big).$$

□

From Theorem 2, the variance resulting from the evaluation of f grows only by a factor that is quadratic in n relative to the initial variance. This very low noise growth is particularly low for the full evaluation of a symmetric scheme. It represents an encouraging step toward the practical use of Nostalgia in the context of hybrid homomorphic encryption.

7 Conclusion

In this work, we introduced Nostalgia, a new filtered LFSR designed to provide 128-bit security. We analyzed state-of-the-art attacks on this class of constructions and proposed a design with a larger register size and an increased number of variables in the filtering function to enhance resistance against known attacks. By carefully selecting the filter function and increasing the register size, Nostalgia mitigates the weaknesses that led to the decline of filtered LFSRs following the introduction of (fast) algebraic attacks. We selected a weightwise quadratic function as the filtering function and demonstrated that its standard cryptographic properties are sufficient for the targeted security level. Additionally, we examined how alternative filtering functions could lead to dedicated attacks that do not apply to Nostalgia. Furthermore, we showed that the proposed stream cipher, particularly the chosen WWQ function, has the potential to be efficient for future applications in hybrid homomorphic encryption. The rise of arithmetic-oriented cryptographic primitives and specialized symmetric designs suggests that filtered LFSRs, once considered obsolete, may be reconsidered in modern settings where their structure and efficiency offer new advantages.

We hope that this work will revive interest in filtered LFSRs, encouraging further research into secure instances and fostering cryptanalysis efforts to refine and strengthen their security.

References

[ACG+06] Frederik Armknecht, Claude Carlet, Philippe Gaborit, Simon Künzli, Willi Meier, and Olivier Ruatta. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2006.

[AGHM24] Diego F. Aranha, Antonio Guimarães, Clément Hoffmann, and Pierrick Méaux. Secure and efficient transciphering for fhe-based MPC. *IACR Cryptol. ePrint Arch.*, 2024. https://eprint.iacr.org/2024/1702.

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for mpc, and more. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, 2019.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219, 2016.

[AMT22] Tomer Ashur, Mohammad Mahzoun, and Dilara Toprakhisar. Chaghri - A FHE-friendly block cipher. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *CCS 2022*, pages 139–150. ACM, 2022.

[APRR24] Navid Alamati, Guru-Vamsi Policharla, Srinivasan Raghuraman, and Peter Rindal. Improved alternating-moduli prfs and post-quantum signatures. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VIII*, volume 14927 of *LNCS*, pages 274–308. Springer, 2024.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 430–454. Springer, 2015.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions: Anemoi permutations and Jive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 507–539. Springer, 2023.

[BP05] An Braeken and Bart Preneel. On the algebraic immunity of symmetric boolean functions. In *Progress in Cryptology - INDOCRYPT 2005, 6th International Conference on Cryptology in India, Bangalore, India, December 10-12, 2005, Proceedings*, pages 35–48, 2005.

[Bry91] Randal E. Bryant. On the complexity of VLSI implementations and graph representations of boolean functions with application to integer multiplication. *IEEE Trans. Computers*, pages 205–213, 1991.

[BS00] Alex Biryukov and Adi Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 1–13, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[BV16] Sonia Bogos and Serge Vaudenay. Optimization of \mathsf LPN solving algorithms. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 703–728, 2016.

[Car04] Claude Carlet. On the degree, nonlinearity, algebraic thickness, and nonnormality of boolean functions, with developments on symmetric functions. *IEEE Trans. Information Theory*, pages 2178–2185, 2004.

[Car21] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2021.

[Car22] Claude Carlet. A wide class of boolean functions generalizing the hidden weight bit function. *IEEE Trans. Inf. Theory*, 68(2):1355–1368, 2022.

[CCF+16] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 313–333. Springer, 2016.

[CCH+24] Mingyu Cho, Woohyuk Chung, Jincheol Ha, Jooyoung Lee, Eun-Gyeol Oh, and Mincheol Son. FRAST: TFHE-friendly cipher based on random S-boxes. *IACR Trans. Symmetric Cryptol.*, 2024(3):1–43, 2024.

[CDPP22] Kelong Cong, Debajyoti Das, Jeongeun Park, and Hilder V. L. Pereira. Sortinghat: Efficient private decision tree evaluation via homomorphic encryption and transciphering. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *CCS 2022*, pages 563–577. ACM, 2022.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. *J. Cryptol.*, 33(1):34–91, 2020.

[CHK+21] Jihoon Cho, Jincheol Ha, Seongkwang Kim, ByeongHak Lee, Joohee Lee, Jooyoung Lee, Dukjae Moon, and Hyojin Yoon. Transciphering framework for approximate homomorphic encryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 640–669. Springer, 2021.

[CHMS22] Orel Cosseron, Clément Hoffmann, Pierrick Méaux, and François-Xavier Standaert. Towards case-optimized hybrid homomorphic encryption - featuring the Elisabeth stream cipher. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 32–67. Springer, 2022.

[CL11] Y. Chen and P. Lu. Two classes of symmetric boolean functions with optimum algebraic immunity: Construction and analysis. *IEEE Transactions on Information Theory*, 57(4):2522–2538, April 2011.

[CM03] Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2003.

[CM21] Claude Carlet and Pierrick Méaux. A complete study of two classes of boolean functions: direct sums of monomials and threshold functions. *IEEE Transactions on Information Theory*, pages 1–1, 2021.

[Cou03] Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer, 2003.

[CP08] Christophe De Cannière and Bart Preneel. Trivium. *LNCS, New Stream Cipher Designs - The eSTREAM Finalists*, page 244–266, 2008.

[CR16] Anne Canteaut and Yann Rotella. Attacks against filter generators exploiting monomial mappings. In Thomas Peyrin, editor, *Fast Software Encryption*, pages 78–98, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[CS91] Vladimir V. Chepyzhov and Ben J. M. Smeets. On A fast correlation attack on certain stream ciphers. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 176–185. Springer, 1991.

[CS24] Claude Carlet and Palash Sarkar. Constructions of efficiently implementable boolean functions possessing high nonlinearity and good resistance to algebraic attacks. *IACR Cryptol. ePrint Arch.*, page 1305, 2024.

[CT00] Anne Canteaut and Michaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer, 2000.

[CV05] Anne Canteaut and Marion Videau. Symmetric boolean functions. *IEEE Trans. Information Theory*, pages 2791–2811, 2005.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A cipher with low anddepth and few ands per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 662–692. Springer, 2018.

[DGH+23] Christoph Dobraunig, Lorenzo Grassi, Lukas Helminger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Pasta: A case for hybrid homomorphic encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):30–73, 2023.

[Did06] F. Didier. A new upper bound on the block error probability after decoding over the erasure channel. *IEEE Trans. Inf. Theory*, 52(10):4496–4503, 2006.

[DKR+22] Christoph Dobraunig, Daniel Kales, Christian Rechberger, Markus Schofnegger, and Greg Zaverucha. Shorter signatures based on tailor-made minimalist symmetric-key crypto. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *CCS 2022*, pages 843–857. ACM, 2022.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing groebner bases. *Journal of Pure and Applied Algebra*, 139:61–88, june 1999.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Grobner bases without reduction to zero. In *Workshop on application of Groebner Bases 2002*, Catania, Spain, 2002.

[GHR+23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst meets fluid-spn: Griffin for zero-knowledge applications. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 573–606. Springer, 2023.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 519–535. USENIX Association, 2021.

[GLR+20] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 674–704. Springer, 2020.

[GM22] Agnese Gini and Pierrick Méaux. On the weightwise nonlinearity of weightwise perfectly balanced functions. *Discret. Appl. Math.*, 322:320–341, 2022.

[GMM+24] Lorenzo Grassi, Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Generalized feistel ciphers for efficient prime field masking. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part III*, volume 14653 of *LNCS*, pages 188–220. Springer, 2024.

[Gol82] Solomon Golomb. *Shift register sequences*, volume 1. Laguna Hills, Calif., Aegean Park Press, 1982.

[GØSW23] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From farfalle to megafono via ciminion: The PRF hydra for MPC applications. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 255–286. Springer, 2023.

[GRHH11] Guang Gong, Sondre Rønjom, Tor Helleseth, and Honggang Hu. Fast discrete fourier spectra attacks on stream ciphers. *IEEE Transactions on Information Theory*, 57(8):5555–5565, 2011.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

[HKL+22] Jincheol Ha, Seongkwang Kim, ByeongHak Lee, Jooyoung Lee, and Mincheol Son. Rubato: Noisy ciphers for approximate homomorphic encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 581–610. Springer, 2022.

[HMR20] Clément Hoffmann, Pierrick Méaux, and Thomas Ricosset. Transciphering, using filip and TFHE for an efficient delegation of computation. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 39–61. Springer, 2020.

[HMS23] Clément Hoffmann, Pierrick Méaux, and François-Xavier Standaert. The patching landscape of elisabeth-4 and the mixed filter permutator paradigm. In Anupam Chattopadhyay, Shivam Bhasin, Stjepan Picek, and Chester Rebeiro, editors, *INDOCRYPT 2023, Part I*, volume 14459 of *LNCS*, pages 134–156. Springer, 2023.

[HR04] Philip Hawkes and Gregory G. Rose. Rewriting variables: The complexity of fast algebraic attacks on stream ciphers. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2004.

[HS05] Jin Hong and Palash Sarkar. New applications of time memory data tradeoffs. In Bimal Roy, editor, *Advances in Cryptology - ASIACRYPT 2005*, pages 353–372, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Jea16] Jérémy Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[KHS+23] Seongkwang Kim, Jincheol Ha, Mincheol Son, ByeongHak Lee, Dukjae Moon, Joohee Lee, Sangyup Lee, Jihoon Kwon, Jihoon Cho, Hyojin Yoon, and Jooyoung Lee. AIM: symmetric primitive for shorter signatures with stronger security. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *CCS 2023*, pages 401–415. ACM, 2023.

[LMK+23] Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo. Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 227–256. Springer, 2023.

[MCJS19] Pierrick Méaux, Claude Carlet, Anthony Journault, and François-Xavier Standaert. Improved filter permutators for efficient FHE: better instances and implementations. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT*, volume 11898 of *LNCS*, pages 68–91. Springer, 2019.

[Méa17] Pierrick Méaux. *Hybrid fully homomorphic framework*. PhD thesis, École Normale Supérieure, Paris, France, 2017.

[Méa19] Pierrick Méaux. On the fast algebraic immunity of majority functions. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology – LATINCRYPT 2019*, pages 86–105, Cham, 2019. Springer International Publishing.

[Méa21] Pierrick Méaux. On the fast algebraic immunity of threshold functions. *Cryptogr. Commun.*, 13(5):741–762, 2021.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016.

[MMMS23] Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Effective and efficient masking with low noise using small-mersenne-prime ciphers. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 596–627. Springer, 2023.

[MO24] Pierrick Méaux and Yassine Ozaim. On the cryptographic properties of weightwise affine and weightwise quadratic functions. *Discret. Appl. Math.*, 355:13–29, 2024.

[MPC04] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic attacks and decomposition of boolean functions. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 474–491, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[MPP24] Pierrick Méaux, Jeongeun Park, and Hilder V. L. Pereira. Towards practical transciphering for FHE with setup independent of the plaintext space. *IACR Commun. Cryptol.*, 1(1):20, 2024.

[MS88] Willi Meier and Othmar Staffelbach. Fast correlation attacks on stream ciphers (extended abstract). In Christoph G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314. Springer, 1988.

[MST24] Pierrick Méaux, Tim Seuré, and Deng Tang. The revisited hidden weight bit function. *IACR Cryptol. ePrint Arch.*, page 2022, 2024.

[NLV11] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, page 113–124, New York, NY, USA, 2011. Association for Computing Machinery.

[QFLW09] Longjiang Qu, Keqin Feng, Feng Liu, and Lei Wang. Constructing symmetric boolean functions with maximum algebraic immunity. *IEEE Trans. Information Theory*, pages 2406–2412, 2009.

[RH07] Sondre Røjom and Tor Helleseth. A new attack on the filter generator. *IEEE Transactions on Information Theory*, 53(5):1752–1758, 2007.

[Røn17] Sondre Rønjom. Improving algebraic attacks on stream ciphers based on linear feedback shift register over $\mathbb{F}_{2^k}$. *Des. Codes Cryptography*, 82(1–2):27–41, jan 2017.

[Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE*, IT-30(5):776–780, 1984.

[Sie85] Thomas Siegenthaler. Decrypting a Class of Stream Ciphers Using Ciphertext Only. *IEEE Trans. Computers*, 34(1):81–85, 1985.

[SM07] Palash Sarkar and Subhamoy Maitra. Balancedness and correlation immunity of symmetric boolean functions. *Discrete Mathematics*, pages 2351 – 2358, 2007.

[WCST14] Qichun Wang, Claude Carlet, Pantelimon Stanica, and Chik How Tan. Cryptographic properties of the hidden weighted bit function. *Discret. Appl. Math.*, 174:1–10, 2014.

[WWL+24] Ruida Wang, Yundi Wen, Zhihao Li, Xianhui Lu, Benqiang Wei, Kun Liu, and Kunpeng Wang. Circuit bootstrapping: Faster and smaller. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 342–372. Springer, 2024.

A taps

The selected bit indices from the LFSR state used as inputs to the filtering function (*i.e.* the taps) are as follows:

$$\begin{aligned} taps = (&0,\ 2,\ 5,\ 9,\ 10,\ 15,\ 17,\ 20,\ 24,\ 25,\ 28,\ 32,\ 36,\ 40,\ 41,\ 51,\ 58,\ 62,\ 66,\ 77,\ 79,\\ &83,\ 99,\ 100,\ 104,\ 124,\ 129,\ 131,\ 137,\ 160,\ 177,\ 178,\ 193,\ 194,\ 197,\ 200,\ 201,\\ &216,\ 218,\ 228,\ 230,\ 238,\ 240,\ 250,\ 281,\ 298,\ 307,\ 316,\ 324,\ 328,\ 335,\ 342,\ 347,\\ &350,\ 361,\ 375,\ 377,\ 378,\ 379,\ 381,\ 387,\ 390,\ 397,\ 430,\ 440,\ 456,\ 457,\ 469,\ 476,\\ &484,\ 501,\ 531,\ 535,\ 536,\ 541,\ 557,\ 565,\ 566,\ 568,\ 569,\ 575,\ 581,\ 603,\ 608,\ 609,\\ &624,\ 638,\ 639,\ 647,\ 651,\ 661,\ 681,\ 696,\ 697,\ 699,\ 700,\ 701,\ 702,\ 710,\ 712,\ 726,\\ &752,\ 759,\ 769,\ 780,\ 781,\ 791,\ 792,\ 808,\ 813,\ 814,\ 827,\ 857,\ 866,\ 874,\ 893,\ 899,\\ &932,\ 956,\ 961,\ 963,\ 964,\ 975,\ 976,\ 997,\ 1000,\ 1014,\ 1021) \end{aligned}$$