The Patching Landscape of Elisabeth-4 and the Mixed Filter Permutator Paradigm

Clément Hoffmann[1], Pierrick Méaux[2], François-Xavier Standaert[1]

[1] UCLouvain, ICTEAM/ELEN/Crypto Group, Belgium
[2] University of Luxembourg, Luxembourg

Abstract. Filter permutators are a family of stream cipher designs that are aimed for hybrid homomorphic encryption. While originally operating on bits, they have been generalized to groups at Asiacrypt 2022, and instantiated for evaluation with the TFHE scheme which favors a filter based on (negacyclic) Look Up Tables (LUTs). A recent work of Gilbert et al., to appear at Asiacrypt 2023, exhibited (algebraic) weaknesses in the Elisabeth-4 instance, exploiting the combination of the 4-bit negacyclic LUTs it uses as filter. In this article, we explore the landscape of patches that can be used to restore the security of such designs while maintaining their good properties for hybrid homomorphic encryption. Starting with minimum changes, we observe that just updating the filter function (still with small negacyclic LUTs) is conceptually feasible, and propose the resulting Elisabeth-b4 design with three levels of NLUTs. We then show that a group permutator combining two different functions in the filter can simplify the analysis and improve performances. We specify the Gabriel instance to illustrate this claim. We finally propose to modify the group filter permutator paradigm into a mixed filter permutator, which considers the permutation of the key with elements in a group and a filter outputting elements in a different group. We specify the Margrethe instance as a first example of mixed filter permutator, with key elements in $\mathbb{F}_2$ and output in $\mathbb{Z}_{16}$, that we believe well-suited for recent fully homomorphic encryption schemes that can efficiently evaluate larger (not negacyclic) LUTs.

1 Introduction

In recent years, symmetric designs purposed for hybrid homomorphic encryption have been a topic of considerable research attention. They enable a client requiring homomorphic computations to exclusively rely on symmetric cryptography, delegating the computationally intensive homomorphic tasks to the server. Examples include LowMC [ARS+15], Kreyvium [CCF+16], FLIP [MJSC16], FiLIP [MCJS19], Rasta [DEG+18], Hera [CHK+21], Rubato [HKL+22], Chagri [AMT22] or Pasta [DGH+23]. In this paper we are concerned with the Improved Filter Permutator (IFP) paradigm [MCJS19], which is a type of stream ciphers leveraging the possibility to perform a part of its computations in a non homomorphic manner. The IFP paradigm has been extended to groups at Asiacrypt 2022 – leading to the Group Filter Permutator (GFP) paradigm – and the Elisabeth family of such ciphers was introduced as a first example by Cosseron et al. in [CHMS22].

In a work to appear at Asiacrypt 2023, Gilbert et al. demonstrated that Elisabeth-4, an instance of the Elisabeth family, has less than the 128 bits of security claimed. They show that the algebraic system derived from the last significant bit of Elisabeth-4's ciphertexts has an insufficient number of monomials over $\mathbb{F}_2$ that leads to a linearization attack which, after various optimizations, is able to retrieve the secret key in 2^{88} operations. There are two main factors allowing this vulnerability. The first one is the size of Elisabeth-4's inner function, which only takes 5 elements of $\mathbb{Z}_{16}$ as input. The second one is the combination of Negacyclic Look-Up Tables (NLUTs) and the modular addition over $\mathbb{Z}_{16}$ which results in the least significant bit of the output having a weak algebraic structure.

In this article, we present three methods for modifying the Elisabeth-4 instance, necessitating increasingly substantial alterations to the original scheme. The first method, Elisabeth-b, essentially involves expanding the parameters' size. It entails an increase in both the number of inputs to the inner function and the number of inner functions that are summed. In the second method, Gabriel, we combine different inner functions in the sum in order to harness the benefits of each, thus enabling us to mitigate performance overheads. Finally, we introduce the Mixed Filter Permutator (MFP) paradigm, which combines different groups. We then present Margrethe, a family of stream ciphers that displays how the MFP avoids the weaknesses highlighted in [GBJR23].

2 Preliminaries

Notations. For two integers a and b such that $a \leq b$ we denote by $[a, b]$ the set $\{a, a+1, \ldots, b-1, b\}$ and $[b]$ for $[1, b]$.

2.1 Boolean functions and cryptography

We introduce some key concepts of Boolean functions and relevant cryptographic criteria, and their generalizations for functions from $\mathbb{G}^n$ to $\mathbb{G}$ where $\mathbb{G}$ is a group (with operation denoted "+"). We refer to [Car21] for the notions on Boolean functions, and [CHMS22] for the generalizations we consider.

Definition 1 ((Vectorial) Boolean function). *An (n, m) vectorial Boolean function F is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. When $m = 1$ the $(n, 1)$ vectorial Boolean function is referred to as a Boolean function and the space of n-variable Boolean function is denoted by $\mathcal{B}_n$. The coordinate functions of F are the m Boolean functions f_i which associate for each $x \in \mathbb{F}_2^n$ the i-th binary output of $F(x)$. The component functions of F are the $2^m - 1$ non-trivial linear combinations of the coordinate functions of F.*

Definition 2 (Algebraic Normal Form (ANF) and degree). *The Algebraic Normal Form of a Boolean function f is its n-variable polynomial representation over $\mathbb{F}_2$ (belonging to $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$):*

$$f(x) = \sum_{I \subseteq [n]} a_I \left(\prod_{i \in I} x_i \right) = \sum_{I \subseteq [n]} a_I x^I,$$

where $a_I \in \mathbb{F}_2$. The algebraic degree of f is defined as: $\mathsf{deg}(f) = \max_{\{I \mid a_I = 1\}} |I|$ (with the convention that $\mathsf{deg}(0) = 0$).

Definition 3 (Algebraic Immunity). *The algebraic immunity of a Boolean function $f \in \mathcal{B}_n$, denoted by $\mathsf{AI}(f)$, is defined as:*

$$\mathsf{AI}(f) = \min_{g \neq 0} \{\mathsf{deg}(g) \mid fg = 0 \text{ or } (f+1)g = 0\}.$$

The Boolean function g is called an annihilator of f (or $f + 1$).

Definition 4 (Balancedness and Resiliency). *A Boolean function $f \in \mathcal{B}_n$ is balanced if $|\{x \mid f(x) = 0\}| = |\{x \mid f(x) = 1\}|$. The function f is m-resilient if any of its restrictions obtained by fixing at most m of its variables is balanced. We denote by $\mathsf{res}(f)$ the maximum resiliency (also called resiliency order) m of f and set $\mathsf{res}(f) = -1$ if f is unbalanced.*

Definition 5 (Nonlinearity). *For $f \in \mathcal{B}_n$ the nonlinearity NL of f is the minimum Hamming distance between f and all the affine functions in $\mathcal{B}_n$:*

$$\mathsf{NL}(f) = \min_{g,\, \mathsf{deg}(g) \leq 1} \{d_H(f, g)\},$$

where $d_H(f, g) = \#\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}$ is the Hamming distance between f and g; and with $g(x) = a \cdot x + \varepsilon$, $a \in \mathbb{F}_2^n, \varepsilon \in \mathbb{F}_2$ where $\cdot$ denotes the inner product.

We denote by NL^d the order-d nonlinearity of f, the minimum Hamming distance between f, and all functions of degree at most d.

We recall the notion of direct sum and some properties of the functions obtained with this construction.

Definition 6 (Direct Sum). *Let $f \in \mathcal{B}_n$ and $g \in \mathcal{B}_m$, f and g operating on different variables, the direct sum h of f and g is defined by:*

$$h(x, y) = \mathsf{DS}(f, g) = f(x) + g(y), \quad \text{where } x \in \mathbb{F}_2^n \text{ and } y \in \mathbb{F}_2^m.$$

Lemma 1 (Boolean direct sum properties (e.g. [MJSC16] Lemma 3)). *Let $h = \mathsf{DS}(f, g)$ be the direct sum of f and g n and m-variable Boolean functions respectively. Then $\mathsf{DS}(f, g)$ has the following cryptographic properties:*

1. *Degree:* $\mathsf{deg}(h) = \max(\mathsf{deg}(f), \mathsf{deg}(g))$.
2. *Algebraic immunity:* $\max(\mathsf{AI}(f), \mathsf{AI}(g)) \leq \mathsf{AI}(h) \leq \mathsf{AI}(f) + \mathsf{AI}(g)$.

3. *Resiliency:* $\mathsf{res}(h) = \mathsf{res}(f) + \mathsf{res}(g) + 1$.
4. *Nonlinearity:* $\mathsf{NL}(h) = 2^m\mathsf{NL}(f) + 2^n\mathsf{NL}(g) - 2\mathsf{NL}(f)\mathsf{NL}(g)$.

Lemma 2 ([Méa22] Lemma 5). *Let $n, m \in \mathbb{N}^*$, f and g be Boolean functions in n and m. If $\mathsf{AI}(f) < \mathsf{deg}(g)$ then $\mathsf{AI}(\mathsf{DS}(f, g)) > \mathsf{AI}(f)$.*

Lemma 3 ([Méa22] Lemma 6). *Let $t \in \mathbb{N}^*$, and $f_1, \ldots, f_t$ be t Boolean functions, if for $r \in [t]$ there exists r different indexes $i_1, \cdots, i_r$ of $[t]$ such that $\forall j \in [r], \mathsf{deg}(f_{i_j}) \geq j$ then $\mathsf{AI}(\mathsf{DS}(f_1, \ldots, f_t)) \geq r$.*

The notions defined above on Boolean functions can easily be extended to functions from $\mathbb{G}^n$ to $\mathbb{G}$, we denote these extended notions by the subscript $\mathbb{G}$.

Definition 7 (Cryptographic criteria over $\mathbb{G}$). *For a function f from $\mathbb{G}^n$ to $\mathbb{G}$ we denote:*

- $\mathsf{deg}_{\mathbb{G}}(f)$ *the degree over $\mathbb{G}$. It corresponds to the minimum degree over the polynomial representations of f in the polynomial ring $(\mathbb{G}, \cdot)[x_1, \ldots, x_n]$ when such representations exist.*
- $\mathsf{res}_{\mathbb{G}}(f)$ *the resiliency order over $\mathbb{G}$. f is balanced if and only if:*

$$\forall a \in \mathbb{G} : |\{x \mid f(x) = a\}| = |\mathbb{G}|^{n-1}.$$

  f is m-resilient if all the sub-functions obtained by fixing up to m variables are balanced.
- $\mathsf{NL}_{\mathbb{G}}(f)$ *the nonlinearity over $\mathbb{G}$. The nonlinearity is taken as the minimum Hamming distance between f and the affine functions: $a_0 + \sum_{i=1}^{n} a_i x_i$ where the a_i describe $\mathbb{G}^{n+1}$. $\mathsf{NL}^d_{\mathbb{G}}(f)$ denotes the order-d nonlinearity over $\mathbb{G}$.*

We also denote $\mathsf{DS}_{\mathbb{G}}(f, g)$ the direct sum $f(x) + g(y)$ where $x \in \mathbb{G}^n$ and $y \in \mathbb{G}^m$ and f and g are two n-variable and m-variable functions defined on distinct variables, the "+" here designs the operation of $\mathbb{G}$.

Functions from $\mathbb{G}^n$ to $\mathbb{G}$ (when $\mathbb{G}$ is only a group and not a field) are referred to as polyfunctions when they admit a polynomial representation. This formalism has been used lately in cryptography in the context of bootstrapping for fully homomorphic encryption [GIKV23]. In the following, we recall a result on the number of polyfunctions:

Property 1 (Number of polyfunctions, [SHW23], Proposition 26). The number of d-variable polynomial with coefficients in $\mathbb{Z}_{p^m}$ is:

$$\psi_d(p^m) = \prod_{\substack{\boldsymbol{k} \in \mathbb{N}^d \\ e_p(\boldsymbol{k}) < m}} p^{m - e_p(\boldsymbol{k})},$$

where $e_p(\boldsymbol{k}) := \max\{x \in \mathbb{N} : p^x | \boldsymbol{k}!\}$ and $\boldsymbol{k}! := \prod_{i=1}^{d} k_i!$.

2.2 Group Filter Permutator paradigm

Paradigm The Group Filter Permutator has been introduced in [CHMS22], it generalized the improved filter permutator model [MCJS19] to groups. It is illustrated in Figure 1, is defined by a group $\mathbb{G}$ with operation noted $+$, a forward secure PRNG, a key size N, a subset size n, and a filtering function f from $\mathbb{G}^n$ to $\mathbb{G}$. To encrypt m elements of $\mathbb{G}$ under a secret key $K \in \mathbb{G}^N$, the public parameters of the PRNG are chosen and then the following process is executed for each key stream s_i (for $i \in [m]$):

- The PRNG is updated, its output determines a subset, a permutation, and a length-n vector of $\mathbb{G}$.
- the n-element subset S_i is chosen as a subset of the N key elements,
- the n-to-n permutation P_i is chosen,
- a $\mathbb{G}^n$ vector is uniformly sampled. This vector is named whitening and denoted by w_i,
- the key stream element s_i is computed as $s_i = f(P_i(S_i(K)) + w_i)$, where $+$ denotes the element-wise addition of $\mathbb{G}$.

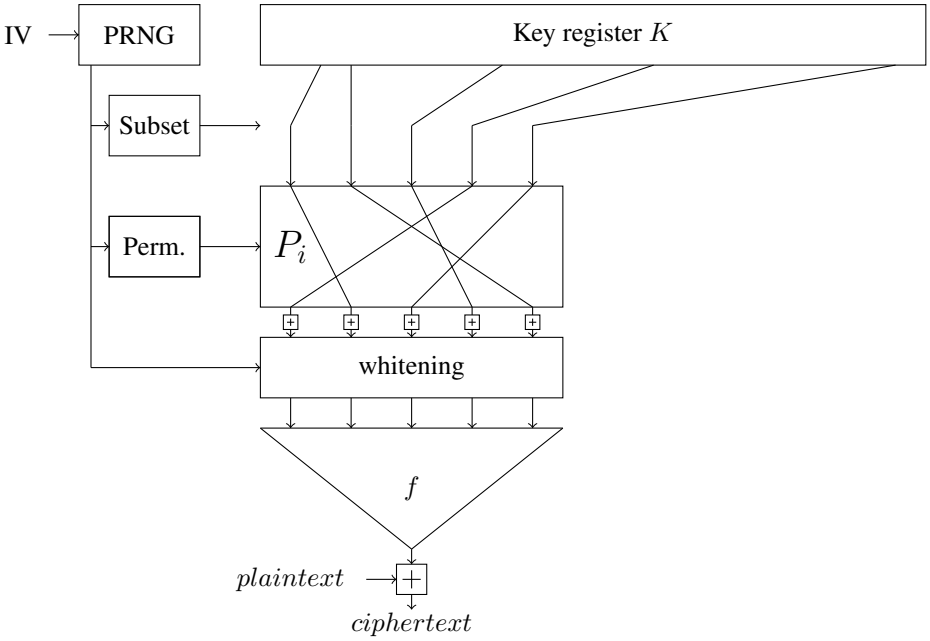

Fig. 1. The Group Filter Permutator design

Precision on the forward secure PRNG We instantiate the forward secure PRNG following the Bellare Yee construction [BY03]. The high-level idea of the construction is to use an IV as an AES key. With this key, two AES encryptions are performed, one producing pseudo-random bits, the other providing a fresh AES key to iterate the process. More precisely, the IV provides two constants C_0, C_1, and for each K_i, $\mathsf{AES}_{K_i}(C_0) = K_{i+1}$ and $\mathsf{AES}_{K_i}(C_1)$ provides 128 bits of randomness.

In practice the subset and permutation are computed at the same time using an aborted Knuth shuffling. To create a size-n permuted subset from the N-element key, we shuffle n elements of the key using Knuth shuffling [FY53, Knu97], and stop the algorithm after the first n iterations.[3]. Finally, n last random nibbles are used to generate the whitening. If the number of requested ciphertexts $m < 2^\lambda$ requires more pseudo-random bits than the PRNG can provide in a secure manner, a new instance is used with new constants.

Security model In the paradigms we consider in this article, the subsets, permutations and whitening are chosen from a forward secure PRNG to prevent malleability, and as before for the GFP paradigm we assume that no weaknesses come from the randomness generation and the analysis can focus on the properties of the filter. As a consequence, the IV of the PRNG is considered known by the adversary, but not chosen. In the context of HHE, where only an honest but curious model is considered for the server, it is classical to assume that the client sets the public values. The honest but curious model is justified by the fact that FHE in not IND-CCA (INDistinguishability against Chosen Ciphertext Attack), therefore we consider that the same server that does not temper with the homomorphic protocol will not temper with the symmetric encryption part. In this context, we do not consider chosen-IV attacks.

We consider the security in the known plaintext/ciphertext pairs setting. In order to assess the security of the schemes presented in this article, we verify that attacks that apply against the paradigms used have a cost of more than $2^\lambda = 2^{128}$ in either memory or operations and a data cost of more than $2^{\frac{\lambda}{2}} = 2^{64}$.

Elisabeth-4 Elisabeth has been introduced in [CHMS22]. It is a stream cipher family following the GFP paradigm that has been designed to take advantage of the most native functions of the TFHE [CGGI16] fully homomorphic encryption scheme. Accordingly, it relies on modular additions and negacyclic LUTs, where for a LUT of size n even negacyclic means that for $x \in [0, n/2-1]$ $f(i) = -f(i+n/2) \mod n$. For efficiency reasons, the modulo chosen for Elisabeth is $2^4 = 16$ according to the state of the art on TFHE, which gives the instance Elisabeth-4.

[3] See the Elisabeth implementation for an example

Elisabeth-4 is the instance of the GFP with the following particularities: $\mathbb{G} = \mathbb{Z}_{16}$, $N = 256$, $n = 60$, the filtering function f is the direct sum of 12 times the 5-to-1 function g described in Figure 2. The 8 particular negacyclic LUTs of Elisabeth-4 have been chosen at random, they are explicitly given in [CHMS22] Appendix B and the generation protocol is available at https://github.com/princess-elisabeth/sboxes_generation.

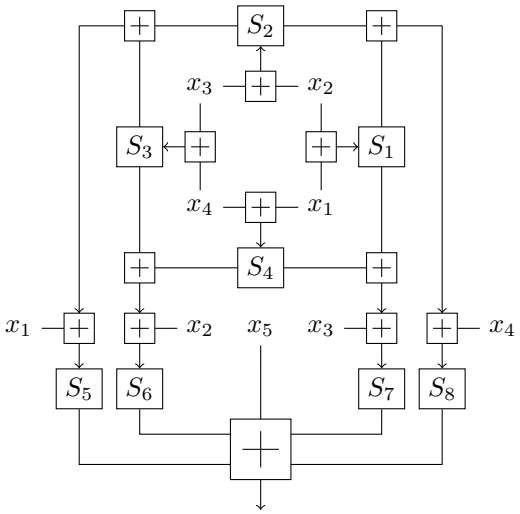

Fig. 2. Elisabeth-4's 5-to-1 inner function.

2.3 GFP and security analysis

We summarize the attacks considered on the GFP and in particular on Elisabeth-4 in [CHMS22], for more details we refer to Section 4 of the same reference.

The GFP generalizes the Improved Filter Permutator (IFP) paradigm from [MCJS19] to any group, therefore the main attacks considered are the ones applying to the IFP and adaptations to the particular group used. These attacks are regrouped in [MCJS19] as algebraic-like attacks and correlation-like attacks that can be combined with guess and determine strategies. They are built on top of former analyses on similar paradigms, filter permutator [MJSC16] further analyzed in [DLR16, CMR17] and Goldreich's pseudorandom generator [Gol00]. The complexity of the different attacks on IFP is derived from parameters of the filtering function f, seen over $\mathbb{G}$ or over another group $\mathbb{G}'$ if this representation is favorable to the attacker. In particular for Elisabeth-4, $\mathbb{G} = \mathbb{Z}_{16}$ and various attacks take advantage of the representation over $\mathbb{F}_2$.

We now give a list of the considered attacks and their complexity. We do not give a detailed description of these attacks nor the methods used to compute their complexities and refer to [CHMS22] for these explanations.

- **Algebraic attack** [CM03] aims at solving a system of equations in which variables are the key elements. Its complexity is $\mathcal{O}(D)^\omega$ where $D = \sum_{i=1}^{\mathsf{AI}(f)} \binom{N}{i}$, where N is the number of variables of the system, f is the filter function and ω is the exponent appearing in the complexity for solving a linear system. We use $\omega = \log(7)$.
- **Fast algebraic attack** [Cou03] was introduced as an attack against filtered Linear feedback shift registers. While this attack does not apply to the GFP, we make the conservative choice of considering it to provide a lower bound on the algebraic-like attack complexity. This covers the fact that algebraic-like attacks may be more efficient than the complexity given above, depending on the structure of the system (i.e. [Fau99,Fau02]). Its complexity is $\mathcal{O}(D\log^2(D) + N \cdot D\log(D))$ where $D = \sum_{i=1}^{\mathsf{AI}(f)+1} \binom{N}{i}$.

- **Correlation-like attacks** approximate the filter function by a degree-d one and solve the resulting degree-d system. Their complexity is $\mathcal{O}\left(((|\mathbb{G}|^n)/\mathsf{NL}_{\mathbb{G}}^d) \cdot N^{d\omega}\right)$, and in particular $\mathcal{O}\left(((|\mathbb{G}|^n)/\mathsf{NL}_{\mathbb{G}}^1) \cdot N^{\omega}\right)$ when $d = 1$.
- **Guess and determine strategies** consist of guessing some key elements and applying the previous attack on the resulting subfunctions. This strategy is only worth it if the cost of guessing key elements is compensated by the simplicity of the remaining system. Performing ℓ guesses brings an overhead of $|\mathbb{G}|^\ell \binom{N}{\ell}$, and we consider the complexity of the simpler remaining system (which is computed by exhausting all subfunctions).

3 Linearization attack from [GBJR23]

The key observation of [GBJR23] is that for Elisabeth-4, at the binary level, the coordinate function of f giving the lowest bit is the direct sum of 12 functions that depends only on 5 chunks of 4 bits [4], with 4 of them interacting in a nonlinear way. Therefore, calling this Boolean function g, it holds that monomials in its ANF cannot have monomials with variables coming from more than 4 different chunks. The number of monomials appearing in the ANF of g is therefore at most:

$$E = \binom{N}{c} 2^{c \cdot t},$$

where N is the number of ($\mathbb{Z}_{16}$) key elements, c the number of chunks, and t the number of bits per chunk. Moreover, since the term $2^{c \cdot t}$ corresponds to the number of monomials in $c \cdot t$ variables, it can be reduced if the function has a lower degree or an extra structure. Accordingly, a linearization attack in complexity E^ω is doable instead of the usual bound D^ω where $D = \sum_{i=0}^{\deg(g)} \binom{Nt}{i}$ is the number of all monomials of degree up to $\deg(g)$ in Nt binary variables. As shown in [GBJR23], filtering the equations allows us to improve the time complexity of the attack. Taking only the equations such that $N' < N$ key variables appear leads to a smaller value of E, at the cost of a higher data complexity.

Since the number of monomials is consequently smaller than the numbers of monomials in Nt variables of degree up to $\deg(g)$, the complexity of this attack by linearization is lower than in the general case, and attacks using the sparsity of the system improve upon this bound. For Elisabeth-4, it gives the following bound on E:

$$E \leq \binom{256}{4} 2^{4 \cdot 4} \approx 2^{43.38},$$

which leads to a linearization attack in complexity $E^\omega \approx 2^{122}$ considering $\omega = \log 7$, already giving a successful attack in the security model of [CHMS22].

The smaller-than-expected number of monomials comes from the small number of chunks mixed in a nonlinear fashion, the use of NLUTs, and the linearity of $\mathbb{Z}_{16}$ addition over $\mathbb{F}_2$. More precisely, we will show that a bigger value of c, the number of chunks, quickly increases the number of monomials and makes the attack impracticable. Then, the ANF of the coordinate function of the lowest bit produced by an NLUT does not contain monomials with the highest input bit (Proposition 2 [GBJR23]). Finally, note that due to the addition modulo 2^t, the next coordinate Boolean function already has a more complex equation. The linear addition of the m outputs is combined with the m choose 2 quadratic terms from these outputs for a direct sum of m components (12 in the case of Elisabeth-4) brought by the carry value. From these three remarks, we can deduce that the attack based on the low number of monomials comes from the specific choice of parameters for Elisabeth-4, and does not apply to the GFP model in general.

Exact number of monomials Looking at the Boolean function of the lowest bit in the particular 5-to-1 function used in [CHMS22], we can count experimentally the number of monomials over $\mathbb{F}_2$ that can be generated from 4 different chunks (looking at the 4! possible permutations). In practice, we obtain that 32535 such monomials can be obtained from the chosen function, compared to 64839 if we consider the number of monomials of degree at most 12 in 16 variables. This number is a lower bound on the number of monomials appearing in the algebraic system obtained by an adversary since the addition of the whitening can create more monomials, but not of a higher degree nor involving variables from different chunks. Accordingly, we consider $E' = E/2$ to estimate a more accurate number of monomials, and display in Table 3 the minimal size of N necessary to have enough

[4] each chunk corresponding to a $\mathbb{Z}_{16}$ element of the key

monomials to avoid the linearization attack, based on the number of chunks c combined in the nonlinear (over $\mathbb{Z}_{16}$) part of the filter and ω.

Table 3 appears to indicate that small modifications of Elisabeth-4's parameter would be sufficient to avoid the attack (while keeping the same Gaussian elimination exponent ω and the same number of chunks $c = 4$, doubling the key-size would give a high enough complexity). However, the filtering function of Elisabeth relies on a direct sum (over $\mathbb{Z}_2^t$) which gives a sparse system of equations. Therefore lower complexity algorithms may apply and considering a smaller value of ω is recommended. For instance, as exhibited by the analysis of [GBJR23] on Elisabeth-4, using block Wiedemann algorithm [Cop94] to take advantage of the sparsity of the system allows to decrease the attack complexity from 2^{122} to 2^{88}. Accordingly, we consider the conservative choice of $\omega = 2$ for the linearization attack, which leads to considering functions mixing 6 chunks instead of 4 for a number of variables lower than 500.

c	$\omega = 2$	$\omega = \log(7)$
4	10783	446
5	1336	106
6	344	44

Table 1. Minimal value of N such that $E'^{\omega} > 2^{128}$.

Accordingly, we present alternative instances of Elisabeth-4 in the following. First, starting with minimal changes we modify the inner function of the filter to mix more chunks nonlinearly, which gives the cipher Elisabeth-b. Then, we consider a filter combining two different functions, in order to improve performance and use each function to counter particular attacks. We specify the Gabriel instance to illustrate this claim. Finally, we propose to extend the group filter permutator paradigm into a mixed filter permutator, which considers the key elements in one group, and the output of the filter in another group. We propose the Margrethe instance as a first example of mixed filter permutator, where the key elements are bits, and the outputs are elements of $\mathbb{Z}_{16}$. We analyze the security of these schemes first following the strategy of [CHMS22] summarized in Section 2.3. Then, we study their security relative to the attack of of [GBJR23], providing a bound on the number of monomials and considering the conservative bound of $\omega = 2$ for the algebraic attack and the linearization attack from [GBJR23].

4 Elisabeth-b

Design motivation and description. The design approach behind Elisabeth-b consists of using a filter based on the direct sum of the same base function (as for Elisabeth-4) but with a base function mixing more chunks. Mixing enough chunks in a nonlinear part allows to prevent a too-low number of monomials in one of the Boolean functions obtained when the keystream is written as equations over $\mathbb{F}_2$. Based on Table 3 a base function mixing 6 chunks (in a nonlinear manner) is sufficient to reach $\lambda = 128$ bits of security.

Contrarily to the 5-to-1 function of Elisabeth-4, 2 levels of NLUTs are not sufficient to mix 6 chunks of 4 bits since a 4-variable Boolean function has degree at most 4 in its inputs. Then, at least 3 levels of NLUTs are necessary. Since we are not aware of designs combining 6 entries, or 3 layers of Sboxes in the literature, we propose the 7-to-1 function over $\mathbb{Z}_{16}$ for Elisabeth-b, that mixes relations modulo 2 and modulo 3 to mix the 6 chunks over the three levels. Similarly to Elisabeth, another variable is simply added, to ensure the balancedness of the output, independently of the choice of the NLUTs.

It gives the following instance, Elisabeth-b4 is the GFP paradigm instantiated with:

- $\mathbb{G} = \mathbb{Z}_{16}$,
- $N = 512$,
- the filter function $f(x_1, \cdots, x_{98})$ the direct sum of 14 times the 7-to-1 function g,
- the 7-to-1 function g described in Algorithm 1,

NLUTs generation. We generate the NLUTs for all the schemes with the same guidelines. In Python, the protocol starts by hashing a character string (for instance, in the case of Elisabeth-b4, the string "Welcome to Gabriel" is used) with SHA256 (in the scripts, using the hashlib module). This hash is then used as the seed of a PRNG (randint from Python, seeded with random.seed), which is then used to generate hex numbers (using

random.randrange(16)). Each number encodes an image of an NLUT, so in order to generate n NLUTS we need $8n$ hex numbers (only the first half of the NLUT is random, the other half can be determined from the previous values).

The script used for generating the NLUTs can be found in the git repository.

Algorithm 1: Elisabeth-b4 7-to-1 function.

] **input :** $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in \mathbb{Z}_{16}^7$
output: $z \in \mathbb{Z}_{16}$
begin
  for i in range(3) **do**
    $x_{2i+2} = x_{2i+2} + x_{2i+1}$
  for i in range(6) **do**
    `// First round of NLUTs`
    $y_{i+1} = S_{i+1}(x_{i+1})$
  for i in range(3) **do**
    $z_{2i+1} = y_{2i \mod 6} + y_{2i+1}$
    $z_{2i+2} = y_{2i+5 \mod 6} + y_{2i+2}$
  for i in range(6) **do**
    $z_{i+1} = z_{i+1} + x_{i+3 \mod 6}$
    `// Second round of NLUTs`
    $z_{i+1} = S_{i+7}(z_{i+1})$
  for i in range(2) **do**
    $t_{3i+1} = z_{3i+1} + z_{3i+2} + z_{3i+3}$
    $t_{3i+2} = z_{3i+2} + z_{3i+4 \mod 6}$
    $t_{3i+3} = z_{3i+3} + z_{3i+4 \mod 6} + y_{3i+1}$
  $t_1 = t_1 + x_6$
  $t_2 = t_2 + x_5$
  $t_3 = t_3 + x_4$
  $t_4 = t_4 + x_2$
  $t_5 = t_5 + x_1$
  $t_6 = t_6 + x_3$
  $z = x_7$
  for i in range(6) **do**
    `// Third round of NLUTs`
    $u_{i+1} = S_{i+13}(t_{i+1})$
    $z = z + u_{i+1}$
  return z

Security analysis. Using the security analysis from [CHMS22] and the analysis on the number of monomials to avoid the attack from [GBJR23], we verify that the filter satisfies different properties over $\mathbb{Z}_{16}$ and $\mathbb{F}_2$:

- g is not a polyfunction,
- the nonlinearity of g is sufficiently high, even after guessing $\mathbb{Z}_{16}$ variables,
- g is balanced,
- the 15 component Boolean functions of g have a sufficient algebraic immunity, even after guessing $\mathbb{F}_2$ variables,
- the 15 component Boolean functions of g have a sufficient nonlinearity, even after guessing $\mathbb{F}_2$ variables,
- the 15 component Boolean functions of g are balanced,
- the 15 component Boolean functions of g have enough monomials in their ANF, even after guessing $\mathbb{F}_2$ variables.

Since Elisabeth-b4 uses a 7-to-1 inner function it leads to different challenges than Elisabeth-4 where the inner function is 5-to-1. Both for the analysis over $\mathbb{Z}_{16}$ and $\mathbb{F}_2$ the computations needed to determine the cryptographic

parameters are possible in reasonable time for Elisabeth-4 (looking at the properties of function from $\mathbb{G}^4$ to $\mathbb{G}$ or Boolean functions in 16 variables), whereas standard algorithms require a consequent amount of time or data to perform the computations on functions from $\mathbb{G}^6$ to $\mathbb{G}$ or 24-variable Boolean functions.

Tools and methods for computing the Boolean functions properties. In order to compute the Boolean properties of Elisabeth-b4's filter function, we used the Sagemath Boolean function module for the nonlinearity, and the DahuHunting Python library (from [DMR23]) to compute the degree[5].

While those libraries are the fastest we found to perform those computations, the timings are not good enough to compute the degree for each of the $4 \times \binom{24}{2}$ 22-variable subfunctions after 2 variable guesses. To speed up the computation, we use a trick to not compute the complete algebraic normal form of every subfunction, and perform a lazy evaluation. The main idea of this technique is to first sort the ANF of the n-variable functions in $n+1$ lists, each one containing the indexes of all the monomials with a given degree. We can focus solely on the list that corresponds to the monomials with the highest degree, and then use the guess and determine approach exclusively for those highest-degree monomials. Finding a high-degree monomial is sufficient to determine the degree of the subfunctions.

All the scripts used for generating Elisabeth-b4's negacyclic look-up tables, computing Elisabeth-b4's security parameters as well as an implementation of the stream-ciphers filter inner function can be found in the following repository:
https://anonymous.4open.science/r/elisabeth_patch_scriptsB27F.

Degree and nonlinearity of the component Boolean functions. We perform the analysis over $\mathbb{F}_2$ using a similar method as in [CHMS22] to analyze Elisabeth-4 through the analysis of Beth-b4, the scheme obtained by replacing some of the modulo 16 addition with bitwise XOR. The addition of the independent variable in the 7-to-1 function (5-to-1 in Elisabeth-4) is replaced, and the 13 additions (11 in Elisabeth-4) inside the direct sum are replaced. This simplification allows us to use the results on direct sums to bound the properties of the Boolean functions on the filter in 98×4 variables from 24-variable functions. Since the addition over $\mathbb{Z}_{16}$ seen as a vectorial Boolean function corresponds to the (length-4) bitwise XOR plus functions of higher degree (corresponding to the carry values), it is natural to think that the cryptographic properties of the simplified version are weaker than the original one. Accordingly, we introduce Beth-b4, the variation of Elisabeth-b4 in which some $\mathbb{Z}_{16}$ additions are replaced by a bitwise XOR, we conjecture Elisabeth-b4, is at least as secure as Beth-b4.

Formally, the filter function of Beth-b4 can be written as the function ϕ from $\mathbb{G}^{98}$ to $\mathbb{F}_2^4$ defined as:

$$\phi(x_1, \ldots, x_{98}) = \bigoplus_{i=1}^{14} g(x_{7i-6}, x_{7i-5}, x_{7i-4}, x_{7i-3}, x_{7i-2}, x_{7i-1}, 0) \bigoplus_{i=1}^{14} x_{7i},$$

where $\bigoplus$ denotes the bitwise XOR on length-4 vectors.

First, we determine the cryptographic properties of the 15 component Boolean functions of the nonlinear part of g, and the properties of their subfunctions up to guessing two variables, in Table 2. Additionally, we compute the number of monomials of the 15 component functions. Then, we derive lower bounds on the properties of ϕ using the direct sum properties (using Lemma 1 various times for the resiliency and nonlinearity, and Lemma 2 and Lemma 3 for the algebraic immunity from the degree). Finally, we derive the complexity bounds on the different attacks, following the analysis of Section 2.3. The results on ϕ on the resiliency and nonlinearity are displayed in Table 3. For the algebraic immunity, applying Lemma 3 we get that with no guesses the AI is at least 14, and the same bound applies when none of the 14 parts receive more than two guesses. When there are more than 3 guessed variables in one of the 14 functions we cannot conclude with the same arguments since Table 2 stops at two guesses (for computational reasons since computing the degree of all the subfunctions or the AI of the 24-variable functions is time and resources consuming) so this function could be null and therefore not used in the chain of Lemma 3. In this case, we can still obtain $\mathsf{AI} \geq 14 - \lfloor \ell/3 \rfloor$ by applying directly the lemma, but we expect this bound to be too conservative. Instead, we rely on the assumption that the 24-variable functions have AI at least 8, that is $n/3$ (note that the AI of a random function is close to $n/2$: for all $a < 1$ when n tends to infinity $\mathsf{AI}(f)$ is almost surely larger than $n/2 - \sqrt{(n/2)\ln(n/2a\ln 2)}$ [Did06, CM13]). In this case, if one function did not receive guesses and 6 others received less than 3 guesses, applying Lemma 2 already guarantees an AI of at

[5] The degree is not directly implemented in DahuHunting, but can be computed from the Algebraic Normal Form

least 14 (the function that did not receive a guess gives an AI of at least 8, and then the degree of the 6 others allow to increase one by one). We note also that both the complexity of the algebraic attack and fast algebraic attack as described in Section 2.3 (considering the worst possible case for the fast algebraic attack, taking $\mathsf{deg}(g) = 1$ and $\mathsf{deg}(h) = \mathsf{AI}(f) + 1$) is already larger than 128-bits of security for an AI value of 12 and up to 28 variables guessed. The scripts used to compute the complexities given in Table 3 can be found in the git repository.

ℓ	0	1	2
deg	23	21	20
res	0	-1	-1
NL	8355114	4172710	2083058

Table 2. Minimum cryptographic parameters of the nonlinear part of the inner function g seen as a (24,4)-vectorial Boolean function after fixing up to ℓ binary inputs.

Number of guesses ℓ	0	$[1, 13]$	$[15, 27]$
Resiliency	13	$13 - \ell$	-1
$\mathsf{NL}/2^{4n-\ell}$	$\geq 0.5 - 10^{-33}$	$\geq 0.5 - 10^{-32}$	$\geq 0.5 - 10^{-30}$
Correlation attack complexity (bits)	$>> 128$	$>> 128$	$>> 128$

Table 3. Beth-b4 minimal parameter bounds up to ℓ fixed binary inputs and complexity estimations (in bits). The complexity estimations come from Section 2.3, when ℓ lies in an interval the complexity given is the minimal one for this interval of guesses.

Bound on the number of monomials of the component Boolean functions. To avoid the attack from [GBJR23], we counted the number of monomials in the ANF of the component of the inner function. The minimum number of monomials among all of the component Boolean functions is at least $22^{22.95}$. Note that a random polynomial would have 22^{23} monomials, the number of monomials then appears close to optimal.

Applying the analysis from Subsection 3, for the given instance Elisabeth-b4, the number of monomials is bounded by $E \geq \binom{512}{6} 2^{22.95}$. With the assumption that the exponent from the Gaussian elimination is only $\omega = 2$ (the assumption is conservative, the complexity of used algorithms being usually closer to $\log(7)$), the linearization attack complexity is already $E^\omega \geq 2^{134}$.

Distance to polyfunctions. To begin with, we prove that g is not a polyfunction. We then use a code-based argument to prove that the probability that g is close to a polyfunction (in a sense that is formalized below) is negligible. Having g close to a low degree polynomial would expose the scheme to correlation attacks. Since the negacyclic look-up tables were chosen using a hash function, the argument also asserts that the authors could not have put a backdoor by choosing a function g close to a low-degree polynomial.

Proposition 1. *Let f a function from $\mathbb{G}^6$ to $\mathbb{G}$. If $f(8,8,8,8,8,8) - f(0,0,0,0,0,0) \notin \{0, 8\}$ then f is not a polynomial of $(\mathbb{G}, \cdot)[x_1, \ldots, x_6]$.*

Proof. All monomials of $(\mathbb{G}, \cdot)[x_1, \ldots, x_6]$ have the form $x_1^{i_1}\, x_2^{i_2}\, x_3^{i_3}\, x_4^{i_4}, x_5^{i_5}, x_6^{i_6}$ where i_1 to i_6 are positive integers. In $(0,0,0,0,0,0)$ all non-trivial monomials give 0. In $(8,8,8,8,8,8)$ if $\sum_{k=1}^{6} i_k > 1$ the monomial gives 0, else the remaining non trivial monomials give 8. Let p be a polynomial of $(\mathbb{G}, \cdot)[x_1, \ldots, x_6]$, its value in $(0,0,0,0,0,0)$ will be the constant coefficient $a_{\mathbf{0}}$. In $(8,8,8,8,8,8)$, since $2 \times 8 = 0$, every non-trivial monomial multiplied by its constant also gives 0 or 8, and so does their sum. Therefore $p(8,8,8,8,8,8)$ is either equal to $a_{\mathbf{0}}$ or $a_{\mathbf{0}} + 8$, allowing to conclude. □

For the instance of Elisabeth-b4 we have $g(8,8,8,8,8,8) - g(0,0,0,0,0,0) = 13 \neq 0, 8$, therefore by Proposition 1 g is not a polyfunction therefore no classical algebraic attack on $\mathbb{G}$ can be applied.

With the parameters of the scheme, using Proposition 1, the number of 6-variable polyfunctions in $\mathbb{G}$ is 2^{4992} (the script used for computing this number can also be found in the public git while the total number of functions is $16^{16^6} = 2^{262144}$). It allows us to derive the following result to thwart the attacks based on low degree approximation of the filter:

Proposition 2. *Let $\mathcal{G}$ be the space of functions from $\mathbb{G}^6$ to $\mathbb{G}$, the probability that a function f taken uniformly at random in $\mathcal{G}$ agrees with a polynomial of $(\mathbb{G}, \cdot)[x_1, x_2, x_3, x_4, x_5, x_6]$ on at least half of the inputs is lower than 2^{-128}.*

Proof. For this proof, we use the formalism of error-correcting codes. Each function in $\mathcal{G}$ can be uniquely represented by its vector of outputs in $\mathbb{G}^{16^6}$, and the polynomials of $(\mathbb{G}, \cdot)[x_1, x_2, x_3, x_4]$ form a linear code of the vector-space $(\mathbb{G}, \cdot)^{16^6}$ with parameter $[n, k, d]$. The code dimension k is given by Proposition 1 while the length n is 2^{24}. We then determine an upper bound on the number of elements of $\mathbb{G}^{16^4}$ at Hamming distance lower than $n/2 + 1$, and compare this bound with the number of functions in $\mathcal{G}$ to conclude.

We use the standard packing/covering bound to determine the maximum number of elements covered by the union of Hamming balls of radius $n/2$ centered around the code elements (here the polynomials). Each ball contains $B = \sum_{i=0}^{n/2}(|\mathbb{G}| - 1)^i \binom{n}{i}$ elements, giving a total number of $2^{4992}B$ elements: the upper bound reached if no ball intersect.

Finally, since there are $(16)^{16^6} = 2^{2^{26}}$ elements in $\mathcal{G}$ we can compute the proportion p of functions coinciding on at least half of their inputs with a polynomial:

$$p \leq \frac{2^{4992}B}{2^{2^{26}}} = \frac{\sum_{i=0}^{2^{23}} 15^i \binom{2^{24}}{i}}{2^{2^{26}-4992}} \leq \frac{15^{2^{23}} \cdot 2^{2^{24}}}{2^{2^{26}-4992}}$$
$$\leq \frac{16^{2^{23}} \cdot 2^{2^{24}}}{2^{2^{26}-4992}} \leq 2^{2^{25}+2^{24}-2^{26}+4992} = 2^{-2^{24}+4992} < 2^{-128}.$$

□

This property asserts that the amount of functions from $\mathbb{G}^6$ to $\mathbb{G}$ that can be well approximated by a polynomial is negligible, which rules out algebraic attacks that would try to solve a noisy polynomial system.

To sum up, Elisabeth-b4 provides a patch while simply increasing the filter inner function of Elisabeth-4. Since this scheme is purposed to be ran homomorphically on a server, it is relevant to consider a parallel implementation and the time cost of the keystream will be dominated by the evaluation time of the NLUTs. In the end, the performance overhead for this secure instance is limited to one extra layer of (negacyclic) look-up tables.

5 Gabriel

Design motivation and description The design approach for Gabriel consists of having two different branches for the filter function, which we call the left and right parts. The left part uses functions mixing a relatively low number of chunks, like Elisabeth-b4, and is designed to tackle the different attacks described in [CHMS22]. The right part is aimed to mix more chunks to provide sufficiently enough monomials in the algebraic system, to tackle the attack presented in [GBJR23], or variants. Therefore, the two parts consist of direct sums of a function g_L and a function g_R (for Left and Right) and are combined in a direct sum to give the filter f.

Differently to Elisabeth-b's design where all base functions mix 7 chunks, this approach gives a trade-off, allowing some base functions cheaper to compute. For security, the number of inner functions mixing more chunks needs to be sufficient to provide enough monomials in all equations. It can be guaranteed if this number is high enough even after taking into consideration that the adversary can guess a bounded number of variables (fixing the values of the monomials containing it). This number has to be high enough considering also that an adversary can build a system where some key variables never appear, and therefore no monomials in these variables either (by doing preprocessing, since the IV is given by the user and the selected subsets are publicly derived from the IV). The latter corresponds to the filtering technique used in [GBJR23].

We could propose an instance of this design strategy only using the 5-to-1 function of Elisabeth-4 and its compositions. We detail why this natural idea is not the best option, in our opinion. In this case, g_L is only the 5-to-1 function, in direct sum 12 times in order to use the results on the properties over $\mathbb{Z}_{16}$ and $\mathbb{F}_2$ already proven or computed in [CHMS22]. We consider g_R as (direct sums of) the composition of the 5-to-1 function g, that

is $g(y_1, y_2, y_3, y_4, x_5)$ where y_1, y_2, y_3, y_4 are the output of g applied on different key entries. Consequently, it gives a 21-to-1 function over $\mathbb{Z}_{16}$ admitting g as a subfunction. This alternative has the advantage of limiting the number of different NLUTs used in the full scheme and allows an easier implementation from Elisabeth-4. On the downside, very few results are known relatively to the cryptographic parameters of the composition of functions (even for Boolean functions), making it difficult to obtain meaningful bounds on the properties of g_R. Using experimental approaches to determine the parameters of such function is also beyond the computational limit. In particular, the nonlinear part gives Boolean functions in 80 variables making it challenging to store the truth tables of size 2^{80} and impossible to compute the parameters (these algorithms are polynomial in the truth table size, therefore exponential in the number of variables).

Accordingly, for simplicity of analysis, we consider instantiating the left part with the inner function of Elisabeth and the right part with the inner function of Elisabeth-b. Based on the analysis of [CHMS22], the attacks based on the nonlinearity, algebraic immunity and degree over $\mathbb{F}_2$, and the attacks based on the polynomial representation and distance to low degree polyfunctions over $\mathbb{Z}_{16}$ take a complexity higher than the threshold of 2^{128}. Then, based on the analysis of Section 4, the number of monomials produced by the 7-to-1 function of Gabriel-4 is sufficient to prevent the linearization attack. It gives the following proposed instance, Gabriel-4 is the GFP paradigm instantiated with:

- $\mathbb{G} = \mathbb{Z}_{16}$,
- $N = 512$,
- the filter function $f(x_1, \cdots, x_{110})$ the direct sum of 8 times the 5-to-1 function g_L and 10 times the 7-to-1 function g_R,
- the 5-to-1 function is the inner function of Elisabeth-4, described in Figure 2, with the same NLUTs,
- the 7-to-1 function is the function g described in Algorithm 1, with the same NLUTs as Elisabeth-b4.

Security analysis Since Gabriel-4 uses two branches to tackle both the attacks studied in [CHMS22] and [GBJR23], the analysis is straightforward form the one of Elisabeth-4 and Elisabeth-b4. The filter function f is a direct sum containing 8 times the inner function of Elisabeth-4, accordingly the properties of f are at least the ones of Elisabeth-b4's inner function, studied in [CHMS22]. Using the properties on the direct sum of these 8 functions and 6 from the part taken from Elisabeth-b4 gives better cryptographic parameters than the ones of the filter of Elisabeth-4. Thereafter, no attack described in Section 2.3 from [CHMS22] leads to a complexity lower than 2^{128}. Regarding the attack from [GBJR23], the right part mixes non-linearly 6 chunks, which is sufficient to thwart the attack from the analysis of Section 3. The 10 7-to-1 functions in the direct sum prevent further guess and determine techniques to guess $\mathbb{Z}_{16}$ variables to get only equations where only 4 chunks are mixed.

Mixing two different kinds of branches leads to better performances than Elisabeth-b4. While the number of NLUT layers remains 3 because of the Elisabeth-b4 branches, the total number of NLUT to evaluate decreases.

6 Margrethe and Mixed Filter Permutators

The small number of monomials in the ANF of the Boolean functions of Elisabeth-4's filter is strongly linked to the constraints of using only 4-bits NLUTs and the addition modulo 16. The inner function mixes a fixed number of $\mathbb{G}$-variables (c), therefore, in the representation over $\mathbb{F}_2$, no monomials with variables from more than c chunks can appear in the ANF (due to the linearity of the Least Significant Bit (LSB) of the addition modulo 16 when considered over $\mathbb{F}_2$). On the contrary, if the filter acts at the bit level it breaks this structure and the bound of binary variables from at most c different chunks. Moreover, as proven in [GBJR23] the negacyclic property implies that the Boolean function corresponding to the LSB cannot have monomials depending on the Most Significant Bit of the input [6].

Accordingly, we consider a different setting with two differences, the LUTs do not need to be negacyclic and the function can operate at the bit level, that is the (binary) key variables are not in the same chunks for each application of the filter. On the HHE side, removing these restrictions is sound since recent techniques allow the evaluation of larger LUTs, as witnessed by the line of works on full domain functional bootstrapping [KS23, CZB+22, MHWW23]. Furthermore, diverse techniques using changes of representation are getting more common (e.g. [CDPP22, MPP23]).

[6] Denoting f the Boolean function corresponding to the LSB of an n-variable NLUT, the negacyclicity implies $f(x) = f(x + (1, 0_{n-1}))$, hence the sensitivity of f in the MSB is null so no monomial containing the MSB appear in the ANF of f.

Since the group filter permutator is defined only on one group, we introduce the more general paradigm of Mixed Filter Permutator (MFP) in Section 6.1, and a stream cipher design named Margrethe in Section 6.2. We propose an instance, Margrethe-$18-4$, and analyze its security in Section 6.3.

6.1 Mixed Filter Permutator paradigm

The Mixed Filter Permutator (MFP) paradigm differs from the GFP by the use of two (potentially) different groups. Therefore, the nature of the inputs and the outputs can be different, and the evaluation of the cipher can be performed over different structures. Mixing operations from two different structures to get a secure cryptographic primitive is a principle that has been used implicitly for example in AES [DR20], and explicitly more recently [BIP$^+$18, DMMS21, DGH$^+$21, HMM$^+$23]. We give the paradigm description in the following:

Paradigm. The MFP is defined by two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with operation noted $+_1$ and $+_2$, a forward secure PRNG, a key size N, a subset size n, and a filtering function f from $\mathbb{G}_1^n$ to $\mathbb{G}_2$. To encrypt m elements of $\mathbb{G}_2$ under a secret key $K \in \mathbb{G}_1^N$, the public parameters of the PRNG are chosen and then the following process is executed for each key stream s_i (for $i \in [m]$):

- The PRNG is updated, its output determines a subset, a permutation, and a length-n vector of $\mathbb{G}_1$.
- the n-element subset S_i is chosen over N-element key,
- the n to n permutation P_i is chosen,
- the vector, called whitening and denoted w_i, from $\mathbb{G}_1^n$ is chosen,
- the key stream element s_i is computed as $s_i = f(P_i(S_i(K)) +_1 w_i)$, where $+_1$ denotes the element-wise addition of $\mathbb{G}_1$.

Security. Due to the strong similarity with the GFP paradigm, we consider the same attacks detailed in Section 2.3, namely algebraic attacks, [GBJR23]'s attack, correlation attacks, and their variants using guess and determine strategies. Since in the MFP the operations are performed over two groups, the attacks have to be considered relative to these two representations. We remark that it is very similar to the two perspectives considered for the GFP, where attacks are considered on $\mathbb{G}$ and on $\mathbb{G}'$, such as $\mathbb{Z}_{16}$ and $\mathbb{F}_2$ for the instance Elisabeth-4. Thereafter, we study the same kind of attacks for the security of stream ciphers following the MFP paradigm, focusing on particular algebraic and correlation attacks depending on the particular choice of $\mathbb{G}_1$ and $\mathbb{G}_2$.

6.2 Margrethe

We define the family of stream cipher Margrethe and give an instance, Margrethe-18-4, analyzed in Section 6.3. Margrethe's design is characterized by the use of $\mathbb{F}_2$ for $\mathbb{G}_1$, and the group $\mathbb{Z}_{2^\ell}$ with $\ell \in \mathbb{N}^*$ for $\mathbb{G}_2$. The filter function is obtained by the direct sum over $\mathbb{G}_2$ of t times a function obtained by a look-up table following the general principle of her cousin Elisabeth. Using look-up tables from a bits to b bits, we denote by Margrethe-a-b the instance associated with this LUT size.

We propose the following instance, Margrethe-18-4 is the MFP paradigm instantiated with:

- $\mathbb{G}_1 = \mathbb{F}_2$,
- $\mathbb{G}_2 = \mathbb{Z}_{16}$,
- $N = 2048$,
- the filter function is the function f from $\mathbb{F}_2^n$ to $\mathbb{Z}_{16}$, that uses the inner function g, a function from $\mathbb{F}_2^{18}$ to $\mathbb{Z}_{16}$.
  $g : \mathbb{F}_2^{18} \mapsto \mathbb{Z}_{16}$ is given by a LUT.
  $f : \mathbb{F}_2^{308} \mapsto \mathbb{Z}_{16}$ is defined by:

$$f(x_1, \ldots, x_{308}) = \sum_{i=0}^{13} g(x_{22i+1}, \ldots, x_{22i+18}) + \mathbb{Z}_{16}\left(\sum_{k=0}^{3} 2^k x_{22i+19+k}\right),$$

  where the symbols Σ and $+$ denote the addition modulo 16 and $\mathbb{Z}_{16}(x_1, x_2, x_3, x_4)$ denotes the element of $\mathbb{Z}_{16}$ with binary representation (x_1, x_2, x_3, x_4).

Note that the key of Margrethe-18-4 has the same size as the one of Elisabeth-b4. Indeed, key elements of Margrethe-18-4 are elements of $\mathbb{F}_2$ while Elisabeth-b4 key elements are in $\mathbb{Z}_{16}$.

The LUT generation is similar to the NLUTs generation for Elisabeth-b (Section 4). In Python, the protocol starts by hashing the character string "Welcome to Margrethe" with SHA256 (in the scripts, using the hashlib module). This hash is then used as the seed of a PRNG, which is then used to generate hex numbers. Each number encodes an image of a LUT, so in order to generate the 18-to-4 LUT we need 2^{18} hex numbers. The script used for generating the LUT can be found in the git repository.

6.3 Security analysis

For the security of Margrethe-18-4 we consider the attacks only coming from the representation over $\mathbb{F}_2$ for two reasons. First, the function g acts at the bit level therefore the result of f cannot be written in terms of key elements of $\mathbb{Z}_{16}$ as for Elisabeth-4. Then, as proven in Proposition 2, the probability for a function taken at random to be close (in Hamming distance) to a polyfunction over $\mathbb{Z}_{16}$ is negligible, making it difficult for an adversary to take advantage of the representation over $\mathbb{Z}_{16}$.

Consequently, we analyze the security of Margrethe-18-4 by considering the different attacks based on its representation as a vectorial Boolean function. Similarly to the analysis of Section 4, we study the properties of a simplification of Margrethe-18-4 that we call Mag-18-4 where the additions modulo 16 are replaced by a length-4 bitwise XOR, allowing to study the property of the filter function from the properties of direct sums. As previously, we conjecture that Margrethe-18-4, is at least as secure as Mag-18-4. More formally, the filter function of Mag-4 is F from $\mathbb{F}_2^{308}$ to $\mathbb{F}_2^4$ is defined by:

$$F(x_1, \ldots, x_{308}) = \bigoplus_{i=1}^{13} G(x_{22i+1}, \ldots, x_{22i+18}) \bigoplus_{i=1}^{13} (x_{22i+19}, x_{22i+20}, x_{22i+21}, x_{22i+22}),$$

where $\bigoplus$ denotes the (length-4) bitwise XOR and G the interpretation of g of a vectorial Boolean function rather than a function from $\mathbb{F}_2^{18}$ to $\mathbb{Z}_{16}$.

First, we determine the properties of the 15 component functions of G relative to the Boolean cryptographic criteria of resilience, degree and nonlinearity. We also study the same properties for their sub-functions up to guessing two variables. Additionally, we compute the number of monomials of the 15 component functions. Then, we give lower bounds on the properties of F using the direct sum properties (see Lemma 1, Lemma 2 and Lemma 3). Finally, we derive the complexity bounds on the different attacks, following the analysis of Section 2.3. We give the parameters of G in Table 4 and the ones of F and the security bounds in Table 5. For the complexity of the algebraic and fast algebraic attacks, we follow the same strategy as in Section 4. Without any guess, the AI of the components of F is at least 14 using Lemma 3, the same bound is valid when no more than 2 guesses are in the same function. If at least 3 variables are guessed in one function, this one cannot be considered in the chain of Lemma 3, accordingly the bound gives $\mathsf{AI} \geq 14 - \lfloor \ell/3 \rfloor$. If we assume the components of G have AI at least $18/3 = 6$, if one function did not receive guesses and 8 others received less than 3 guesses applying Lemma 2 already guarantees an AI of at least 14. In the end, for an AI value of 12 and up to 28 variables guessed, following the formulas of Section 2.3, the bound on the algebraic attacks complexity is 2^{288} and 2^{128} for the one on the fast algebraic attack complexity.

The minimal number of monomials observed in the ANF for a component of G is $2^{16.9983}$, approximately half of the monomials as for a random 18-variable Boolean function. Accordingly, the number of monomials to consider for a linearization attack is the standard one from Section 2.3 rather than Section 3.

ℓ	0	1	2
deg	17	16	15
res	-1	-1	-1
NL	129757	64516	32013

Table 4. Minimum cryptographic parameters for the function G seen as a vectorial Boolean function after fixing up to ℓ binary inputs

Number of guesses ℓ	0	$[1, 13]$	$[14, 27]$
Resiliency	13	$13 - \ell$	-1
$\mathsf{NL}/2^{4n-\ell 7}$	$\geq 0.5 - 10^{-28}$	$\geq 0.5 - 10^{-25}$	$\geq 0.5 - 10^{-23}$
Correlation attack complexity (bits)	$>> 128$	$>> 128$	$>> 128$

Table 5. Mag-18-4 minimal parameter bounds up to ℓ fixed binary inputs and complexity estimations (in bits). The complexity estimations come from Section 2.3, when ℓ lies in an interval the complexity given is the minimal one for this interval of guesses.

7 Conclusion and open question

To sum up, we studied different patches to the Elisabeth-4 cryptosystem. Elisabeth-b4 brings minimal changes by simply modifying the filter inner function by combining more elements. Gabriel-4 reduces the performance overhead by integrating two different branches in the direct sum of the filter. Leaving the GFP paradigm, Margrethe-18-4 fixes the root of the issue raised by [GBJR23] by mixing the key at the bit level.

On the one hand Elisabeth-b4 and Gabriel-4 use the same operations as Elisabeth-4, with the same size of negacyclic LUT. Their homomorphic implementations would therefore be similar to the one of Elisabeth-4, with slightly different homomorphic parameters. On the other hand, the homomorphic evaluation of instances from Margrethe is left as an open problem. We believe that the structure of the mixed filter permutator can exploit recent advancements in FHE research using full domain, bigger, LUTs (e.g. [KS23, CZB+22, MHWW23]) or by mixing different representations (e.g. [CDPP22]).

8 Acknowledgments

Pierrick Méaux was supported by the ERC Advanced Grant no. 787390.

References

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, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 139–150. ACM, 2022.

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, *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 430–454. Springer, 2015.

BIP+18. Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729. Springer, 2018.

BY03. Mihir Bellare and Bennet S. Yee. Forward-security in private-key cryptography. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2003.

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

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.

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, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 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.

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, *Advances in Cryptology – ASIACRYPT 2021*, pages 640–669, Cham, 2021. Springer International Publishing.

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, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 32–67. Springer, 2022.

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.

CM13. Claude Carlet and Brahim Merabet. Asymptotic lower bound on the algebraic immunity of random balanced multi-output boolean functions. *Adv. Math. Commun.*, 7:197–217, 2013.

CMR17. Claude Carlet, Pierrick Méaux, and Yann Rotella. Boolean functions with restricted input and their robustness; application to the FLIP cipher. *IACR Trans. Symmetric Cryptol.*, 2017(3), 2017.

Cop94. Don Coppersmith. Solving homogeneous linear equations over gf(2) via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

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.

CZB+22. Pierre-Emmanuel Clet, Martin Zuber, Aymen Boudguiga, Renaud Sirdey, and Cédric Gouy-Pailler. Putting up the swiss army knife of homomorphic calculations by means of TFHE functional bootstrapping. *IACR Cryptol. ePrint Arch.*, page 149, 2022.

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 *CRYPTO 2018*, pages 662–692, 2018.

DGH+21. Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. Mpc-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 517–547. Springer, 2021.

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 Transactions on Information Theory*, 52(10):4496–4503, 2006.

DLR16. Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP family of stream ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2016.

DMMS21. Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring crypto-physical dark matter and learning with physical rounding towards secure and efficient fresh re-keying. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):373–401, 2021.

DMR23. Aurélien Dupin, Pierrick Méaux, and Mélissa Rossi. On the algebraic immunity - resiliency trade-off, implications for goldreich's pseudorandom generator. *Des. Codes Cryptogr.*, 91(9):3035–3079, 2023.

DR20. Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition*. Information Security and Cryptography. Springer, 2020.

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.

FY53. Ronald Aylmer Fisher and Frank Yates. *Statistical tables for biological, agricultural and medical research*. Hafner Publishing Company, 1953.

GBJR23. Henri Gilbert, Rachelle Heim Boissier, Jérémy Jean, and Jean-René Reinhard. Cryptanalysis of elisabeth-4. *IACR Cryptol. ePrint Arch.*, page 1436, 2023.

GIKV23. Robin Geelen, Ilia Iliashenko, Jiayi Kang, and Frederik Vercauteren. On polynomial functions modulo p^e and faster bootstrapping for 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 257–286. Springer, 2023.

Gol00. Oded Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(90), 2000.

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, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 581–610. Springer, 2022.

HMM+23. Clément Hoffmann, Pierrick Méaux, Charles Momin, Yann Rotella, François-Xavier Standaert, and Balazs Udvarhelyi. Learning with physical rounding for linear and quadratic leakage functions. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 410–439. Springer, 2023.

Knu97. Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley Professional, third edition, November 1997.

KS23. Kamil Kluczniak and Leonard Schild. FDFB: full domain functional bootstrapping towards practical fully homomorphic encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(1):501–537, 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.

Méa22. Pierrick Méaux. On the algebraic immunity of direct sum constructions. *Discret. Appl. Math.*, 320:223–234, 2022.

MHWW23. Shihe Ma, Tairong Huang, Anyu Wang, and Xiaoyun Wang. Fast and accurate: Efficient full-domain functional bootstrap and digit decomposition for homomorphic computation. *IACR Cryptol. ePrint Arch.*, page 645, 2023.

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.

MPP23. Pierrick Méaux, Jeongeun Park, and Hilder V. L. Pereira. Towards practical transciphering for FHE with setup independent of the plaintext space. *IACR Cryptol. ePrint Arch.*, page 1531, 2023.

SHW23. Ernst Specker, Norbert Hungerbühler, and Micha Wasem. The ring of polyfunctions over z/nz. *Communications in Algebra*, 51(1):116–134, 2023.