Cryptanalysis of a Theorem: Decomposing the Only Known Solution to the Big APN Problem*

Léo Perrin[1(✉)], Aleksei Udovenko[1(✉)], and Alex Biryukov[1,2(✉)]

[1] SnT, University of Luxembourg, Luxembourg City, Luxembourg
{leo.perrin,aleksei.udovenko}@uni.lu
[2] CSC, University of Luxembourg, Luxembourg City, Luxembourg
alex.biryukov@uni.lu

Abstract. The existence of Almost Perfect Non-linear (APN) permutations operating on an even number of bits has been a long standing open question until Dillon et al., who work for the NSA, provided an example on 6 bits in 2009.
In this paper, we apply methods intended to reverse-engineer S-Boxes with unknown structure to this permutation and find a simple decomposition relying on the cube function over $GF(2^3)$. More precisely, we show that it is a particular case of a permutation structure we introduce, the *butterfly*. Such butterflies are $2n$-bit mappings with two CCZ-equivalent representations: one is a quadratic non-bijective function and one is a degree $n+1$ permutation. We show that these structures always have differential uniformity at most 4 when n is odd. A particular case of this structure is actually a 3-round Feistel Network with similar differential and linear properties. These functions also share an excellent non-linearity for $n = 3, 5, 7$.
Furthermore, we deduce a bitsliced implementation and significantly reduce the hardware cost of a 6-bit APN permutation using this decomposition, thus simplifying the use of such a permutation as building block for a cryptographic primitive.

Keywords: Boolean functions, APN, Butterfly structure, S-Box decomposition, CCZ-equivalence, Feistel Network, Bitsliced implementation.

1 Introduction

When designing a symmetric primitive, it is common to use functions operating on a small part of the internal state to provide non-linearity. These are called *S-Boxes* and their properties can be leveraged to justify security against differential [1] and linear [2] attacks using for example a wide-trail argument, as was done for the AES [3].

* The work of Léo Perrin is supported by the CORE ACRYPT project (ID C12-15-4009992) funded by the *Fonds National de la Recherche* (Luxembourg). The work of Aleksei Udovenko is supported by the *Fonds National de la Recherche*, Luxembourg (project reference 9037104)

A popular strategy for choosing S-Boxes with desirable cryptographic properties is to use mathematical construction based for example on the inverse in a finite field [4]. A function with optimal differential property (in a sense that we will define later) is called *Almost Perfect Non-linear* or *APN*. While it is easy to find functions with this property, permutations are more rare. Many monomials are known to be APN permutations in finite fields of size 2^n for n odd (for example the cube function), but whether there even exists APN permutations operating on an even number of bits is still an important research area.

In this context, the 6-bit APN permutation described by a team of mathematicians from the NSA (Dillon *et al.*) in [5] is of great theoretical importance: it is the only known APN permutation for even n so far. Furthermore, it has already been used to design an authenticated cipher: Fides [6]. However, the method used by the Dillon et al. to find it relies on sophisticated considerations related to error correcting codes and no generalization of their results has been published to the best of our knowledge. In their paper, the authors state the "big APN problem" and it is, 6 years later, still as much of an open question:

> **(STILL) The Big APN Problem:** Does there exist an APN permutation on $GF(2^m)$ if m is EVEN and GREATER THAN 6?

Our Contribution By applying methods designed by Biryukov *et al.* to reverse-engineer the S-Box of the last Russian cryptographic standards [7], we show the existence of a much simpler expression of the 6-bit APN permutation. This is stated in Theorem 3 which we reproduce here.

Main Theorem (A Family of 6-bit APN Permutations). The 6-bit permutation described by Dillon *et al.* in [5] is affine equivalent to any involution built using the structure described in Figure 1, where $\odot$ denotes multiplication in the finite field $GF(2^3)$, $\alpha \neq 0$ is such that $\mathrm{Tr}(\alpha) = 0$ and $\mathcal{A}$ denotes any 3-bit APN permutation.

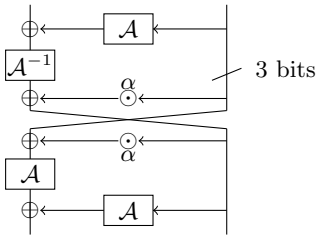


Fig. 1: Some S-Boxes affine-equivalent to the Dillon APN permutation.

We study extensively this structure, both experimentally and mathematically, and derive in particular new families of differentially 4-uniform permutations of $2n$ bits for n odd.

Outline This paper is devoted to first deriving this theorem and then exploring its consequences. Section 2 describes how the cryptanalysis strategy described in [7] can be successfully applied to the 6-bit APN permutation to identify a highly structured decomposition. We then study this structure in Section 3. Next, we show in Section 4 that the same structure can be used to build differentially 4-uniform permutations with algebraic degree at least n in fields of size $2n$ for odd n. Finally, we use our results on the decomposition of 6-bit APN permutations to describe efficient bit-sliced and hardware implementation of some of them in Section 5.

Notations and Definitions

We use common definitions and notations throughout this paper. For the sake of clarity, we list them here. First, we describe the notations related to finite field:

- $\mathbb{F}_{2^n}$ is a finite field of size 2^n,
- for any x in $\mathbb{F}_{2^n}$, the trace of x is $\mathrm{Tr}(x) = \sum_{i=0}^{n-1} x^{2^i}$,

The differential properties of an S-Box $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ are studied using its Difference Distribution Table (DDT), the $2^n \times 2^m$ matrix $\mathcal{D}(f)$ such that $\mathcal{D}(f)[\delta, \Delta] = \#\{x \in \mathbb{F}_{2^n}, f(x+\delta) + f(x) = \Delta\}$. The maximum coefficient[3] in $\mathcal{D}(f)$ is the differential uniformity of f and, if it is equal to u, then we say that f is differentially u-uniform. A differentially 2-uniform function is called Almost Perfect Non-linear (APN).

Similarly, security against linear attacks can be justified using the Linear Approximation Table (LAT)[4] of f. It is the $2^n \times 2^m$ matrix $\mathcal{L}(f)$ such that $\mathcal{L}(f)[a, b] = \#\{x \in \mathbb{F}_{2^n}, a \cdot x = b \cdot y\} - 2^{n-1}$ (where "$\cdot$" denotes the scalar product). The *non-linearity* of a $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is $\mathcal{NL}(f) = 2^{n-1} - \max(|\mathcal{L}(f)[a, b]|)$ where the maximum is taken over all non-zero line and column indices a and b.

Finally, we also consider algebraic decompositions of the functions we study using the following tools:

- if x and u are vectors of $\mathbb{F}_2^n$, then $x^u = \prod_{i=0}^{n-1} x_i^{u_i}$ so that $x^u = 1$ if and only if $x_i = 1$ for all i such that $u_i = 1$,
- the Algebraic Normal Form (ANF) of a Boolean function f is its unique expression $f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u x^u$ where all a_u are in $\{0, 1\}$,
- the algebraic degree of a Boolean function f is denoted $\deg(f)$ and is equal to the maximum Hamming weight of u such that $a_u = 1$ in the ANF of f,
- the field polynomial representation of f mapping $\mathbb{F}_{2^n}$ to itself is its unique expression as a univariate polynomial of $\mathbb{F}_{2^n}$, so that $f(x) = \sum_{i=0}^{2^n-1} c_i x^i$ with c_i in $\mathbb{F}_{2^n}$. It can be obtained using Lagrange interpolation.

[3] The maximum is taken over all non-zero line indices.

[4] This object is also sometimes referred to as the "correlation matrix". Up to a multiplication by a constant factor, the coefficients in the LAT of a function also form its Walsh Spectrum.

Note that the algebraic degree of a polynomial of $\mathbb{F}_{2^n}$ is equal to the maximum Hamming weight of the binary expansions of the exponents in its field polynomial representation. For example, the algebraic degree of the cube function $x \mapsto x^3$ in $\mathbb{F}_{2^n}$ is equal to 2.

Two functions f and g are *affine equivalent* if there exist affine permutations A and B such that $g = B \circ f \circ A$. If we also add an affine function C to the output, that is, $g = B \circ f \circ A + C$, then f and g are *extended affine-equivalent (EA-equivalent)*.

Finally, we denote the concatenation of two binary variables using the symbol "||". In particular, we will often interpret bit-strings of length $2n$ as $x||y$, where x and y are in $\mathbb{F}_2^n$.

2 A Decomposition of the 6-bit APN Permutation

In this section, we identify a decomposition of the Dillon APN permutation. We denote this permutation $S_0 : \mathbb{F}_2^6 \to \mathbb{F}_2^6$ and give its look-up table in Table 1. As we are interested only in its being an APN permutation, we allow ourselves to compose it with affine permutations as such transformations preserve this property. We will omit the respective inverse permutations to simplify our description.

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	00	36	30	0d	0f	12	35	23	19	3f	2d	34	03	14	29	21
1.	3b	24	02	22	0a	08	39	25	3c	13	2a	0e	32	1a	3a	18
2.	27	1b	15	11	10	1d	01	3e	2f	28	33	38	07	2b	2c	26
3.	1f	0b	04	1c	3d	2e	05	31	09	06	17	20	1e	0c	37	16

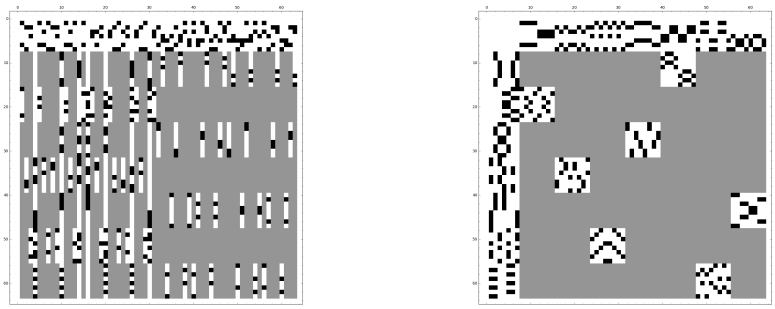
Table 1: The Dillon permutation S_0 in hexadecimal (e.g. $S_0(\texttt{0x10}) = \texttt{0x3b}$).

Our strategy is identical to the one used to recover the structure of the S-Box of the last Russian cryptographic standards described in [7]. First, we obtain a high level decomposition of the permutation relying on two distinct but closely related 3-bit keyed permutations (the "TU-decomposition") in Section 2.1. Then, we decompose these keyed permutations in Sections 2.2. Finally, we provide the complete decomposition of an S-Box affine-equivalent to S_0 in Section 2.3.

2.1 High-Level TU-Decomposition

As suggested in [8] and [7], we looked at the "Jackson Pollock" representation of the absolute value of the LAT of the S-Box (see Figure 2a). We can see some patterns, namely columns and aligned short vertical segments of black and white colors within a grey rectangle (white is 0, grey is 4 and black is 8). The black-and-white columns also have the 8 topmost coefficients equal to zero. Moreover, their horizontal coordinates form a linear subspace of $\mathbb{F}_2^6$.

Therefore, as was done in [7], we compose the S-Box with a particular linear permutation chosen so that these particular columns are clustered to the left of the picture, i.e their abscissa become $[0, 7]$. The black-and-white columns have coordinates $\{0, 4, 10, 14, 16, 20, 26, 30\}$ and the binary expansion of these numbers form a linear subspace of $\mathbb{F}_2^6$ spanned by the binary expansions of $\{4, 10, 16\}$. We thus construct a permutation η, linear over $GF(2)$, such that $\eta : 1 \mapsto 4, 2 \mapsto 10, 4 \mapsto 16$ and then we complete it by setting $\eta : 8 \mapsto 1, 16 \mapsto 2, 32 \mapsto 32$ so that η is a permutation. By Theorem 1 from [7], the composition $\eta^t \circ S_0$ of such mapping with the S-Box will group the black-and-white columns in the LAT. The Jackson Pollock representation of $\eta^t \circ S_0$ is given in Figure 2b.



(a) LAT of S_0.

(b) LAT of $\eta^t \circ S_0$.

Fig. 2: The Jackson Pollock representation of the LAT of two permutations (absolute value). Row/column indices correspond to input/output linear approximation masks respectively. White pixels correspond to 0, grey to 4 and black to 8.

As we can see the columns are now aligned, as was our goal, and the short segments became grouped into small squares, thus making the whole picture more structured. Doing this also caused the appearance of a "white-square" in the top-left square $[0, 7] \times [0, 7]$. This last pattern is a known side effect of the existence of specific integral properties (see Lemma 2 of [7] which is itself derived from [9]). Hence, we checked for integral/multiset properties as defined in [10] and identified the following property: fixing the last 3 bits of the input and letting the first 3 take all possible values leads to the last 3 bits of the output taking all possible values.

We keep following the blueprint laid out in [7] and investigate the consequences of this integral distinguisher. In fact we generalize their next step, which consists in providing a high level decomposition of the S-Box, by describing the *TU-decomposition*.

Lemma 1. *Let f be a function mapping $\mathbb{F}_2^n \times \mathbb{F}_2^n$ to itself such that fixing the right input to any value and letting the left one take all 2^n possible values leads to the left output taking all 2^n possible values. Then f can be decomposed using*

a keyed n-bit permutation T and a keyed n-bit function U (see Figure 3a):

$$f(x,y) \;=\; \big(T_y(x), U_{T_y(x)}(y)\big),$$

Besides, if f is a permutation then U is a keyed permutation.

(a) Basic TU-decomposition. (b) TU-decomposition composed with a swap.

Fig. 3: Principle of the TU-decomposition.

Proof. We simply define $T_y(x)$ to be the left side of $f(x,y)$. Because of the multiset property, T_y is a permutation for all y. We then define U to be such that $U_k(y)$ is the right side of $f\big(T_y^{-1}(k), y\big)$.

If f is a permutation then $(x,y) \mapsto f\big(T_y^{-1}(x), y\big)$ is a permutation equal to $(x,y) \mapsto (x, U_x(y))$. In particular, it holds that U_x is a permutation for all x, making it a keyed permutation. □

We apply Lemma 1 to $\eta^t \circ S_0$ and deduce its TU-decomposition. We actually have the output halves swapped so we may draw the structure in a more symmetric fashion (see Figure 3b). The corresponding keyed permutations T and U are given in Table 2.

	0	1	2	3	4	5	6	7
T_0	0	6	4	7	3	1	5	2
T_1	7	5	1	6	4	2	0	3
T_2	4	3	2	0	5	6	1	7
T_3	3	5	2	1	4	6	7	0
T_4	1	2	0	6	4	3	7	5
T_5	6	5	2	4	7	0	1	3
T_6	5	2	6	4	0	3	1	7
T_7	2	0	1	6	5	3	4	7

(a) T.

	0	1	2	3	4	5	6	7
U_0	0	3	6	4	2	7	1	5
U_1	7	4	0	2	3	6	1	5
U_2	1	4	2	6	3	0	5	7
U_3	7	2	5	1	3	0	4	6
U_4	7	3	4	1	0	2	6	5
U_5	3	7	1	4	2	0	5	6
U_6	1	3	7	4	6	2	5	0
U_7	4	6	3	0	5	1	7	2

(b) U.

Table 2: The keyed permutations T and U. T_k and U_k denote the permutations corresponding to the key k.

The degree of T as a 6-bit permutation is equal to 3 and that of U is equal to 2. However the degree of T^{-1} is equal to 2 as well. One may think that T^{-1} and U

are somehow related and we indeed found that T^{-1} and U are linearly equivalent using the algorithm by Biryukov *et al.* from [11]. The linear equivalence of T^{-1} and U is given by:

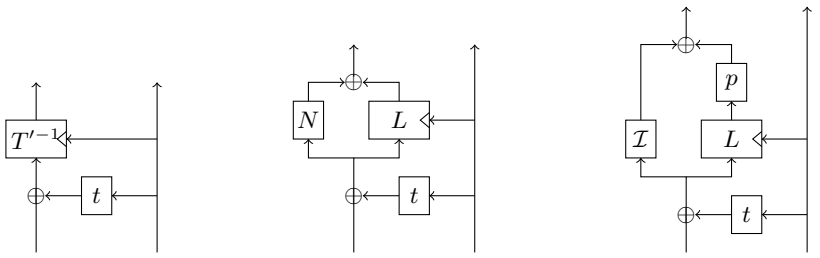
$$U(x) = M'_U \circ T^{-1} \circ M_U(x),$$

where T and U are considered as 6-bit permutations and the linear permutations M_U and M'_U are given as the following binary matrices:

$$M_U = \begin{bmatrix} 0&1&1&0&1&0 \\ 1&0&0&0&1&0 \\ 0&0&1&0&1&1 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&1&1 \end{bmatrix}, M'_U = \begin{bmatrix} 1&1&0&0&0&0 \\ 0&1&0&0&1&0 \\ 1&0&1&1&1&1 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&1&1 \end{bmatrix}.$$

2.2 Decomposing T

As we applied a linear mapping on the output of the S-Box, we might have scrambled the initial structure of U. Hence, we choose the decomposition of T^{-1} as our main target. We start by composing it with a Feistel round to ensure that 0 is mapped to itself for all keys. Again, this simplification was performed while reverse-engineering the GOST S-Box. If we apply such an appropriate Feistel round before or after T^{-1}, the corresponding Feistel function is always a permutation. Moreover, in the case when the Feistel function is used between T and U, the Feistel function is linear[5] so we choose this side. We define $t(k) = T_k(0)$ and $T'_k(x) = T_k(x) \oplus t(k)$ so that $T'_k(0) = T'^{-1}_k(0) = 0$ for all k (see Figure 4a). The linear permutation t is given by $t(x) = (0, 7, 4, 3, 1, 6, 5, 2)$.



(a) Detaching a linear Feistel round.

(b) Splitting T'^{-1} into N and L.

(c) Simplifying N into $\mathcal{I}$ and linear functions.

Fig. 4: Simplifying the keyed permutation T'^{-1}.

We then check the existence of particular algebraic structure in T'. We choose the irreducible polynomial $X^3 + X + 1$ to represent elements of $\mathbb{F}_{2^3}$ as binary

[5] If we had attacked U instead of T^{-1}, then detaching a Feistel function in this way leads only to a nonlinear Feistel function (regardless of the side), which supports our choice of T'^{-1} as an easier target.

strings and, furthermore, we represent these binary strings as integers. In equations we represent such constants in italic. Note that this representation was motivated by convenience reasons for working in Sage [12] and we are using it only in this section for describing the decomposition process.

Now we use Lagrange interpolation to represent each $T_k'^{-1}$ as a polynomial over $\mathbb{F}_{2^3}$. The result is given in Table 3. Interestingly, the coefficients of the non-linear terms x^6, x^5, x^3 are key-independent. We therefore decompose T'^{-1} as a sum of its non-linear part N and its key-dependent linear part L_k so that $T_k'^{-1}(x) = N(x) + L_k(x)$, where $N(x) = 3x^6 + 2x^5 + 5x^3$ and $L_k(x)$ is linear for any k (see Figure 4b).

	0	1	2	3	4	5	6	7	Interpolation polynomial
$T_0'^{-1}$	0	5	7	4	2	6	1	3	$3x^6 + 2x^5 + 3x^4 + 5x^3 + 2x^2 + 0x$
$T_1'^{-1}$	0	3	1	4	7	5	2	6	$3x^6 + 2x^5 + 1x^4 + 5x^3 + 4x^2 + 2x$
$T_2'^{-1}$	0	4	5	7	3	6	2	1	$3x^6 + 2x^5 + 0x^4 + 5x^3 + 0x^2 + 0x$
$T_3'^{-1}$	0	2	3	7	6	5	1	4	$3x^6 + 2x^5 + 2x^4 + 5x^3 + 6x^2 + 2x$
$T_4'^{-1}$	0	2	5	1	7	4	6	3	$3x^6 + 2x^5 + 3x^4 + 5x^3 + 0x^2 + 5x$
$T_5'^{-1}$	0	4	3	1	2	7	5	6	$3x^6 + 2x^5 + 1x^4 + 5x^3 + 6x^2 + 7x$
$T_6'^{-1}$	0	3	7	2	6	4	5	1	$3x^6 + 2x^5 + 0x^4 + 5x^3 + 2x^2 + 5x$
$T_7'^{-1}$	0	5	1	2	3	7	6	4	$3x^6 + 2x^5 + 2x^4 + 5x^3 + 4x^2 + 7x$

Table 3: The values and polynomial interpolation of each $T_k'^{-1}$.

We now simplify N by applying a linear function of our choice after T'^{-1} (see Figure 4c). We allow ourselves to do this because this side corresponds to the input of the S-Box on which, as we said before, we may apply any affine layer as those would preserve the differential uniformity of the whole permutation. Choosing this side also prevents the need for a corresponding modification of U. We choose $p(x) = 4x^4 + x^2 + x$ because $(p \circ N)(x) = x^6$ is the inverse function in $\mathbb{F}_{2^3}$, denoted $\mathcal{I}$.

We further remark that $p \circ L_k$ is simpler than L_k too: there are nonzero coefficients only at x^2 and x^4 (see Table 4). Note also that $p \circ L_2 = 0$ so we add 2 to k to obtain these linear layers:

$$(p \circ L_k)(x) = l_2(k+2)x^2 + l_4(k+2)x^4,$$

where $l_2(x) = 2x^4 + 4x^2 + x$ and $l_4(x) = x^4 + 6x^2 + 2x$ are obtained from the Lagrange interpolations of $p \circ L_k$ given in Table 4.

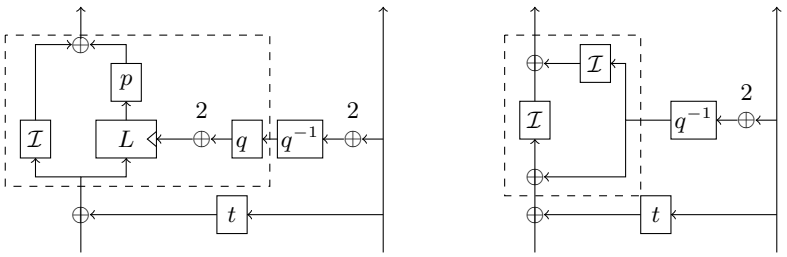
In our effort to simplify the structure, we search for a linear permutation q such that both $l_2 \circ q$ and $l_4 \circ q$ have a simpler form and find that $q(x) = 3x^4 + 7x^2 + 3x$ is such that $(l_2 \circ q)(x) = x^4$ and $(l_4 \circ q)(x) = x^2$. Therefore, we can write $(p \circ L_k)(x) = k'^4 x^2 + k'^2 x^4$, where $k' = q^{-1}(k+2)$. We deduce a representation of the whole structure of $p \circ T'^{-1}$ depending only on linear functions and the inverse function which we describe in Equation (1) and Figure 5.

$$(p \circ T_k'^{-1})(x) = x^6 + x^2 k'^4 + x^4 k'^2 = (x + k')^6 + k'^6, \text{ with } k' = q^{-1}(k+2). \quad (1)$$

Function	Polynomial
$p \circ L_0$	$7x^4 + 3x^2$
$p \circ L_1$	$2x^4 + 4x^2$
$p \circ L_2$	$0x^4 + 0x^2$
$p \circ L_3$	$5x^4 + 7x^2$

Function	Polynomial
$p \circ L_4$	$4x^4 + 6x^2$
$p \circ L_5$	$1x^4 + 1x^2$
$p \circ L_6$	$3x^4 + 5x^2$
$p \circ L_7$	$6x^4 + 2x^2$

Table 4: The interpolation polynomials of each $p \circ L_k$.



(a) Using $k' = q(k) \oplus 2$. (b) Using Equation (1).

Fig. 5: Simplifying $p \circ L$ and thus T'^{-1}. The dashed area corresponds to the equivalence given by Equation 1.

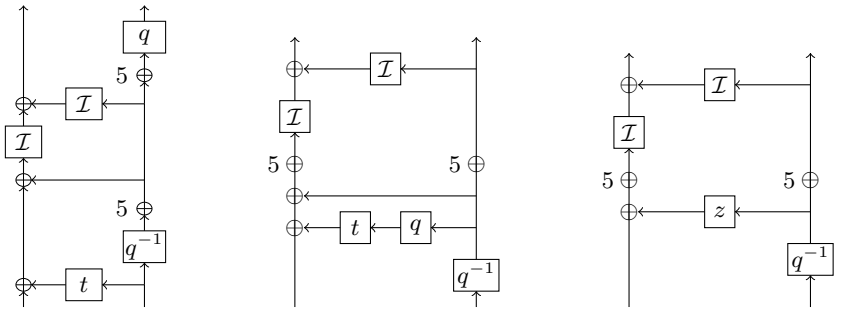
Then, we replace the application of $x \mapsto q^{-1}(x + 2)$ on the horizontal branch in Figure 5b by its application on the right vertical branch followed by its inverse (see Figure 6a; note that $q^{-1}(2) = 5$). By then discarding the affine permutation applied on the top of the right branch (we omit the affine layers applied to the outside of the complete permutation), we obtain the equivalent structure shown in Figure 6b. Finally, we merge the two linear Feistel functions into $z(x) = t(q(x)) \oplus x$ to obtain our final decomposition of $\mathcal{T}^{-1}$:

$$\mathcal{T}^{-1}(\ell||r) \;=\; \mathcal{I}\big(\ell + z\left(q^{-1}(r)\right) + 5\big) + \mathcal{I}\big(q^{-1}(r) + 5\big) \;||\; (q^{-1}(r) + 5),$$

which is also is described in Figure 6c. Now that we have found a decomposition of T, we shall use it to express a whole permutation affine-equivalent to S_0.

2.3 Joining the decompositions of T and U.

Let us now join the decomposition of T and U together, that of U being obtained using that $\mathcal{U}(x) = M'_U \circ \mathcal{T}^{-1} \circ M_U(x)$. The affine transformations applied on the top of T'^{-1} make the relation between T^{-1} and U affine instead of linear on one side. This side corresponds to the output of the S-Box and we omit this transformation. The other linear mapping M_U connecting T^{-1} and U merges with the linear part of T^{-1} and its symmetric copy from U into the linear mapping M (see Figure 7a and 7b). The linear permutation M is given by the



(a) (b) (c)

Fig. 6: Finishing the decomposition of T^{-1}: moving q, q^{-1} and $x \mapsto x+2$ around, removing the outer affine layer and merging the Feistel linear rounds.

following matrix over $\mathbb{F}_2$:

$$M = \begin{bmatrix} 1&0&1&1&1&1 \\ 1&1&0&0&1&0 \\ 0&0&1&1&1&0 \\ 1&1&0&1&0&1 \\ 1&1&1&1&1&0 \\ 1&0&1&0&0&1 \end{bmatrix}.$$

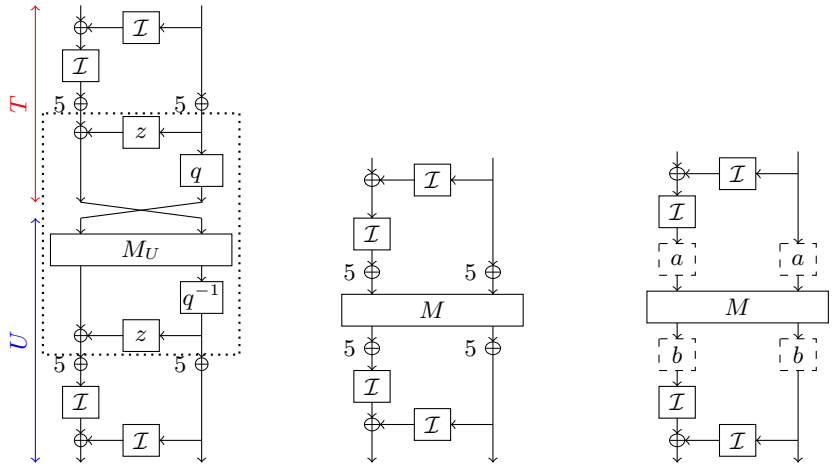
In order to further improve our decomposition, we studied how each component of this structure could be modified so as to preserve the APN property of the permutation. We investigated both the replacement of the linear and non-linear permutations used and describe our findings in Section 3.3. In particular, we found that we could modify the central affine layer in the following fashions while still keeping the APN property of the permutation (see Theorem 2):

- changing the xor constants to any value, in particular 0;
- inserting two arbitrary 3-bit linear permutations a and b as shown in Figure 7c.

Thus, we remove the xors from the structure and exhaustively check all linear permutations a, b such that the resulting linear layer from Figure 7c has the simplest form. We found that for $a(x) = 2x^4+2x^2+4x$ and $b(x) = 2x^4+3x^2+2x$ the resulting matrix can be represented as the following matrix M' over $\mathbb{F}_{2^3}$:

$$M' = \begin{bmatrix} 2 & 5 \\ 1 & 2 \end{bmatrix}.$$

Interestingly, M' is an involution which, because of the symmetry of our decomposition, makes the whole S-Box involutive too! The matrix M' can moreover be decomposed into a 2-round Feistel Network with finite field multiplications by 2 as Feistel functions. We deduce the final decomposition from this final observation and describe it in the following theorem.



(a) Joining the decompositions of T and U.

(b) Merging linear layers.

(c) Allowed transformations.

Fig. 7: Simplifying the middle affine layer. The linear mappings in the dotted area in Figure 7a form the linear layer M.

Theorem 1. *There exist linear bijections A and B such that the Dillon 6-bit permutation is equal to*

$$S_0(x) = B(S_{\mathcal{I}}(A(x) \oplus 9) \oplus 4,$$

where the output of $S_{\mathcal{I}}(\ell||r)$ is the concatenation of two bivariate polynomials of $\mathbb{F}_2[X]/(X^3 + X + 1)$, namely $S^L_{\mathcal{I}}(\ell, r)$ and $S^R_{\mathcal{I}}(\ell, r)$. These are equal to

$$\begin{cases} S^R_{\mathcal{I}}(\ell||r) = (r^6 + \ell)^6 + 2r, \\ S^L_{\mathcal{I}}(\ell||r) = \left(r + 2S^R_{\mathcal{I}}(\ell||r)\right)^6 + S^R_{\mathcal{I}}(\ell||r)^6. \end{cases}$$

A picture representing a circuit computing $S_{\mathcal{I}}$ is given Figure 8.

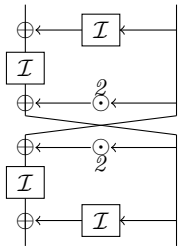


Fig. 8: The APN involution $S_{\mathcal{I}}$, where $\mathcal{I}$ denotes the inverse in the finite field $\mathbb{F}_{2^3}$ with the irreducible polynomial $X^3 + X + 1$, i.e. the monomial $x \mapsto x^6$.

For the sake of completeness, we give the matrices of the linear permutations A and B:

$$A = \begin{bmatrix} 1&1&0&1&0&1 \\ 1&1&1&1&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&1&0&1 \\ 0&0&0&1&0&0 \\ 0&0&0&1&1&0 \end{bmatrix}, \quad B = \begin{bmatrix} 0&1&1&1&0&1 \\ 0&0&0&0&0&1 \\ 0&0&1&1&1&0 \\ 0&0&0&1&1&1 \\ 0&0&1&0&1&0 \\ 1&0&1&1&0&1 \end{bmatrix}.$$

3 Analysing Our Decomposition

In this section, we study the structure of the 6-bit APN permutation we derived from the Dillon S-Box in Section 2. We start with a description of its cryptographic properties in Section 3.1. Then, we generalize this structure into the *Butterfly structure* (see Section 3.2). We investigate how 3-bit affine permutations propagate through the different components of our decomposition in Section 3.3 and then we use this information to deduce how much freedom we have when choosing the different components of the permutation (see Section 3.4).

We discover some new relations between the APN permutation, the Kim function and the cube mapping over $\mathbb{F}_{2^6}$ in Section 3.5. Furthermore, we describe some simple univariate representations of the structure in Section 3.6. We have also noticed that $S_{\mathcal{I}}$ is CCZ-equivalent to the concatenation of two bent functions. However, because it could not produce any new 6-bit APN permutations, we discuss this in the full version of this paper [13].

3.1 Cryptographic Properties

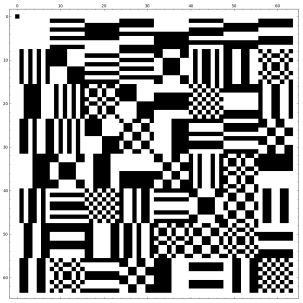
The first consequence of our decomposition is the surprising observation that the 6-bit APN permutation is affine-equivalent to an involution. To the best of our knowledge, this was not known.

The permutation $S_{\mathcal{I}}$ is obviously APN due to how it was obtained, so that the highest differential probability is equal to $2/64 = 2^{-5}$. The Jackson Pollock representation of the DDT of $\mathsf{Swap} \circ S_{\mathcal{I}} \circ \mathsf{Swap}$, where Swap is a simple branch swap, is provided in Figure 9a. The LAT of $S_{\mathcal{I}}$ contains[6], in absolute value, only 3 different coefficients: 945 occurrences of 0, 2688 occurrences of 4 and 336 occurrences of 8 (see Figure 9b). Its maximum linear bias is thus $8/32 = 2^{-2}$. The left half of its output bits have algebraic degree 4 and those on the right half have algebraic degree 3.

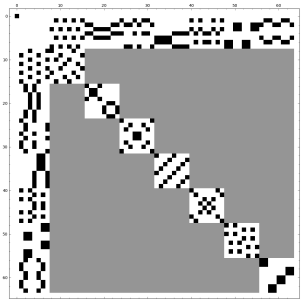
3.2 The Butterfly Structure

As described above, the output of our 6-bit APN permutation $S_{\mathcal{I}}$ is the concatenation of two bivariate polynomials of $\mathbb{F}_{2^3}$. We define the keyed permutation R_k

[6] As $S_{\mathcal{I}}$ is a permutation, we ignore the first line and the first column of its LAT.



(a) DDT of $\mathsf{Swap} \circ S_{\mathcal{I}} \circ \mathsf{Swap}$ (white: 0, black: 2).



(b) LAT of $S_{\mathcal{I}}$ (white: 0, grey: 4, black: 8).

Fig. 9: The Jackson Pollock representation of the DDT and LAT of $S_{\mathcal{I}}$.

of $\mathbb{F}_{2^3}$ with a key in $\mathbb{F}_{2^3}$ as

$$R_k(x) = (x + 2k)^6 + k^6,$$

where R_k is indeed a permutation affine equivalent to the inverse function $x \mapsto x^6$. In fact, its inverse R_k^{-1} such that $R_k^{-1}(R_k(x)) = x$ is equal to $R_k^{-1} = (x + k^6)^6 + 2k$. Using this keyed permutation and its inverse, it is easy to express $S_{\mathcal{I}}$ (see also Figure 10a):

$$S_{\mathcal{I}}(\ell||r) \;=\; R_{R_r^{-1}(\ell)}(r) \;||\; R_r^{-1}(\ell).$$

Using this representation, we show that $S_{\mathcal{I}}$ is CCZ-equivalent to a quadratic function with a very similar structure. First, we recall the definition of CCZ-equivalence (where CCZ stands for Carlet-Charpin-Zinoviev [14]) as it is defined e.g. in [15].

Definition 1 (CCZ-equivalence). *Let f and g be two functions mapping $\mathbb{F}_{2^n}$ to itself. They are said to be CCZ-equivalent if the sets $\{(x, f(x)) \mid x \in \mathbb{F}_{2^n}\}$ and $\{(x, g(x)) \mid x \in \mathbb{F}_{2^n}\}$ are affine equivalent. In other words, they are CCZ-equivalent if and only if there exists a linear permutation L of $(\mathbb{F}_{2^n})^2$ such that*

$$\left\{(x, f(x)), \forall x \in \mathbb{F}_{2^n}\right\} = \left\{L\left(x, g(x)\right), \forall x \in \mathbb{F}_{2^n}\right\}.$$

For example, a permutation is CCZ-equivalent to its inverse. As is shown in Proposition 2 of [16], CCZ-equivalence preserves both the differential uniformity and the Walsh spectrum (i.e. the distribution of the coefficients in the LAT).

Lemma 2. *The permutation $S_{\mathcal{I}}$ is CCZ-equivalent to the quadratic function $Q_{\mathcal{I}} : \mathbb{F}_2^6 \to \mathbb{F}_2^6$ obtained by concatenating two bivariate polynomials of $\mathbb{F}_{2^3}$:*

$$Q_{\mathcal{I}}(\ell||r) \;=\; R_r(\ell)||R_\ell(r).$$

A representation of $Q_{\mathcal{I}}$ is given Figure 10b.

(a) The permutation $S_{\mathcal{I}}$. (b) The function $Q_{\mathcal{I}}$.

Fig. 10: Two CCZ-equivalent APN functions of $\mathbb{F}_2^6$.

Proof. The functional graph of the function $Q_{\mathcal{I}}$ is the following set:

$$\{\big(x||y,\ R_y(x)||R_x(y)\big),\ \forall x||y \in \mathbb{F}_2^6\},$$

in which we can replace the variable x by $z = R_y(x)$ so that $x = R_y^{-1}(z)$ as R_k is invertible for all k. We obtain a new description of the same set:

$$\{\big(R_y^{-1}(z)||y,\ z||R_{R_y^{-1}(z)}(y)\big),\ \forall z||y \in \mathbb{F}_2^6\}.$$

As the function $\mu : (\mathbb{F}_2^6)^2 \to (\mathbb{F}_2^6)^2$ with $\mu(x||y, a||b) = (a||y, b||x)$ is linear, this graph is linearly equivalent to the following one:

$$\{\big(z||y,\ R_{R_y^{-1}(z)}(y))||R_y^{-1}(z),\ \forall z||y \in \mathbb{F}_2^6\},$$

which is the functional graph of $S_{\mathcal{I}}$: the two functions are CCZ-equivalent. □

Definition 2 (Butterfly Structure). *Let α be in $\mathbb{F}_{2^n}$, e be an integer such that $x \mapsto x^e$ is a permutation of $\mathbb{F}_{2^n}$ and $R_k[e, \alpha]$ be the keyed permutation*

$$R_k[e, \alpha](x) = (x + \alpha k)^e + k^e.$$

We call Butterfly Structures *the functions of $(\mathbb{F}_{2^n})^2$ defined as follows:*

- *the* Open Butterfly *with branch size n, exponent e and coefficient α is the permutation denoted H_e^α defined by:*

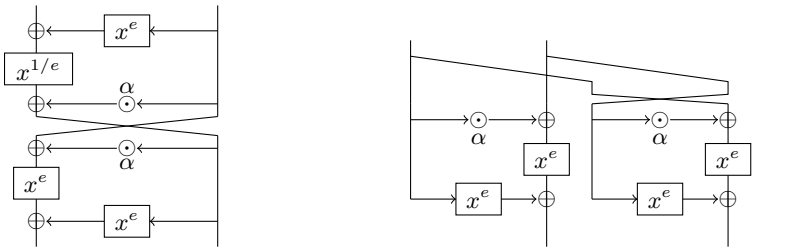
$$\mathsf{H}_e^\alpha(x, y) = \big(R^{-1}_{R_y[e,\alpha](x)}(y),\ R_y[e, \alpha](x)\big),$$

- *the* Closed Butterfly *with branch size n, exponent e and coefficient α is the function denoted V_e^α defined by:*

$$\mathsf{V}_e^\alpha(x, y) = \big(R_y[e, \alpha](x),\ R_x[e, \alpha](y)\big).$$

Furthermore, the permutation H_e^α and the function V_e^α are CCZ-equivalent.

Pictures representing such functions are given in Figure 11. Our decomposition of the 6-bit APN permutation and its CCZ-equivalent function have butterfly structures: $S_{\mathcal{I}} = \mathsf{H}_6^2$ and $Q_{\mathcal{I}} = \mathsf{V}_6^2$. In fact, the proof of the CCZ-equivalence of open and closed butterfly is identical to that of Lemma 2. The properties of such structures for $n > 3$ are studied in Section 4.1, in particular in Theorem 4. In this section, we focus on the case $n = 3$.



(a) Open (bijective) butterfly H_e^α. (b) Closed (non-bijective) butterfly V_e^α.

Fig. 11: The two types of butterfly structure with coefficient α and exponent e.

3.3 Propagation of Affine Mappings through the Components

As we have seen, affine-equivalence and CCZ-equivalence are key concepts in our analysis of $S_{\mathcal{I}}$. In this context, it is natural to extend our analysis not only to outer affine layers applied before and after the permutation but also to the inner affine permutation itself: what modifications can we make to this function while preserving the APN property of the structure? In this section, we study the "propagation" of affine layers in the sense defined below. Our study will show some interesting properties of the structure and why changing some components can lead to an affine equivalent structure.

Definition 3 (Propagation of Affine Layers). *We say that an affine transformation A propagates through a component C if there exists an affine transformation A' such that $C \circ A = A' \circ C$.*

Note that this definition is another way of looking at self-equivalence: indeed, $C \circ A = A' \circ C$ is equivalent to $C = A'^{-1} \circ C \circ A$.

Theorem 2. *Consider the two permutations of $\mathbb{F}_2^6$ with structures shown in Figure 12, where $A, B : \mathbb{F}_2^3 \to \mathbb{F}_2^3$ are some linear bijections,*

$$M = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$$

is an invertible matrix operating on column-vectors, p, q, r, s are 3×3 submatrices over $\mathbb{F}_2$ and a, b, c, d are constants of $\mathbb{F}_{2^3}$. Assume also that q is invertible. Then both structures are affine-equivalent for any choice of M (with q invertible) and constants. As a consequence, all such structures are in the same affine-equivalence class.

Proof. We start by proving that adding constants a, b, c, d as described in Figure 12 leads to affine-equivalent permutations. For now, we assume that A and B are the identity. First, we modify the constants without modifying the function to move them to the right branches only. To do this, we move a through

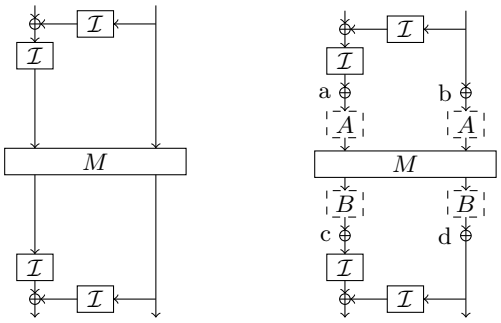


Fig. 12: Affine equivalent structures.

the linear layer M and modify b in such a way that c cancels out. The difference required, $x = b' \oplus b$, is a solution to the equation $p(a) \oplus q(x) = c$, so that $x = q^{-1}(p(a) \oplus c)$ and x always exists since q is invertible. Thus, for

$$b' = b \oplus x = b \oplus q^{-1}(p(a) \oplus c),$$
$$d' = d \oplus r(a) \oplus s(x) = d \oplus r(a) \oplus s(q^{-1}(p(a) \oplus c)),$$

constructions with the structure described in Figure 13a and 13b are functionally equivalent.

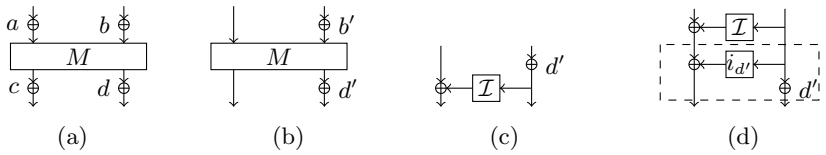


Fig. 13: How the xors around the central linear layer are affine equivalent to outer linear layers.

The xors remaining on the right branches propagate through the Feistel function $\mathcal{I}$ and are equivalent to particular outer affine transformations. Note that in $\mathbb{F}_{2^3}$ we have[7]

$$\mathcal{I}(x + d') = (x + d')^6 = x^6 + d'^2x^4 + d'^4x^2 + d'^6 = \mathcal{I}(x) + i_{d'}(x),$$

where $i_{d'}(x) = d'^2x^4 + d'^4x^2 + d'^6$ is an affine function and can be seen as an additional Feistel round. The propagation of the xor with d' is illustrated on

[7] For larger fields the inverse function does not satisfy the property and therefore such propagation is impossible. An anonymous reviewer pointed out that this works in $\mathbb{F}_{2^3}$ because the inverse function there has boolean algebraic degree 2 and therefore its derivative is linear.

Figure 13c and 13d: the functions described on both figures are functionally equivalent. The case with b' is symmetrical.

We have now showed that the xors a, b, c, d can be removed and the resulting S-Box stays in the same affine equivalence class. Since the equivalence relation is symmetric, we can also modify the constants to arbitrary values. We now move on to studying the impact of branch-wise affine permutations.

It is sufficient to show how the two applications of B propagate through the bottom field inverses, the case of A being symmetric. We start by analyzing propagation through a single inverse function (see Figure 14).

In the case when the input transformation is linear (when $c = 0$), it is easy to see that if the equivalent output transformation is affine, then it is actually linear, since $B(0) = \mathcal{I}(0) = 0$. By exhaustively checking all linear 3-bit permutations B we found that the only functions which propagate in such way are 21 functions of the form $x \mapsto \lambda x^{2^e}$, where $e \in \{0, 1, 2\}, \lambda \in \mathbb{F}_{2^3}, \lambda \neq 0$. This propagation is quite obvious since $(\lambda x^{2^e})^6 = \lambda^6 (x^6)^{2^e}$.

The more interesting case is when the input transformation is affine. By exhaustive search we found that *any* linear bijection B propagates through the field inverse in $\mathbb{F}_{2^3}$, but only *together* with a particular B-dependent xor constant. That is, for any linear bijection B there exists a constant c such that $\mathcal{I}(B(x) + c) = B'(\mathcal{I}(x)) + c'$ for some linear bijection B' and constant c', i.e. the affine function $B(x) + c$ propagates through the inverse function in the affine way (see Figure 14b).

(a) Linear. (b) Affine.

Fig. 14: Propagation of linear/affine permutation through the field inverse.

Note that after applying the linear bijections A and B the top right submatrix of M becomes $B \times q \times A$ and is still invertible, therefore the part of theorem about constant addition, which we already proved, is still applicable. Hence for any linear mappings A, B we can add the xor constants required for propagation of A, B. Let x, y be the values on the left and right branches respectively after applying the linear layer M. Then the left half of the output is equal to

$$x' = \mathcal{I}(B(x)+c)+\mathcal{I}(B(y)+c) = B'(\mathcal{I}(x))+c'+B'(\mathcal{I}(y))+c' = B'(\mathcal{I}(x)+\mathcal{I}(y)),$$

and the right half is simply $y' = B(x) + c$. The procedure is shown in Figure 15. □

Theorem 2 shows an interesting property of the field inverse in $\mathbb{F}_{2^3}$: all linear bijections propagate through it together with some xor constant. We have

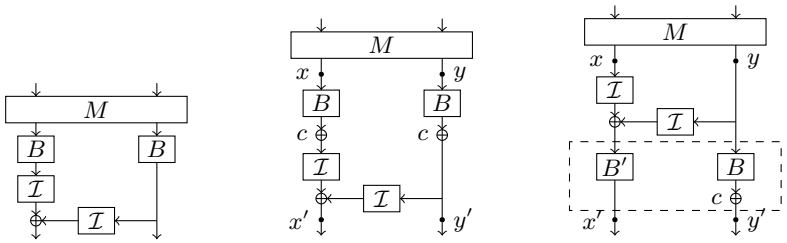


Fig. 15: Propagation of affine mappings through the inverses. The dashed area contains the outer affine parts.

checked all nonlinear exponent functions in $\mathbb{F}_{2^n}$ for $n = 4, 5, 6, 7$ and none of them has this property. By using self-equivalence algorithm from [11] we found that in these fields the only affine transformations which propagate through such nonlinear monomial functions are the linear mappings of the form $x \mapsto \lambda x^{2^e}$, where $e \in [0, n-1], \lambda \in \mathbb{F}_{2^n}, \lambda \neq 0$.

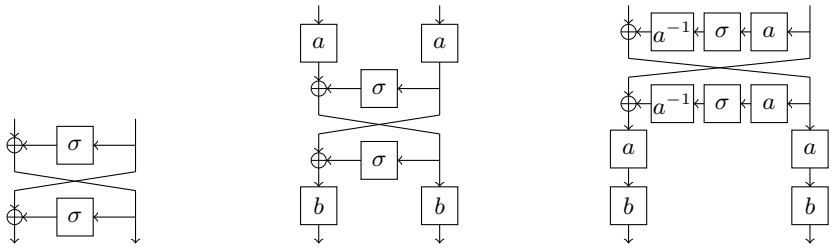
In our decomposition the central linear layer is a 2-round Feistel Network where the round function σ is multiplication by 2 in the finite field defined by a particular polynomial (see Figure 16a). By applying linear transformations around as in Theorem 2 we obtain an affine equivalent S-Box. We can move the linear functions a through the linear Feistel network, such that the round functions are modified and the linear functions a merge with the linear functions b as shown in Figures 16b and 16c. Since by Theorem 2 the outer linear function $b \circ a$ can be omitted, we conclude that σ may be replaced by $a^{-1} \circ \sigma \circ a$ for arbitrary linear permutation a. By exhaustively checking $a^{-1} \circ \sigma \circ a$ for all a we found that there are 24 unique variants of σ. In particular, in the field defined by the irreducible polynomial $X^3 + X + 1$ the allowed multiplications by a constant α are when $\alpha \in \{2, 4, 6\}$, where the latter two are obtained from $\sigma(x) = 2x$ by setting $a(x) = x^2$ and $a(x) = x^4$. In the field defined by the other irreducible polynomial $X^3 + X^2 + 1$ such constants become $\alpha \in \{3, 5, 6\}$. We note that all these elements can be unambiguously defined by the conditions $\mathrm{Tr}(\alpha) = 0, \alpha \neq 0$ in both fields.

3.4 Replacing Components

It is natural to ask how unique are the components of the decomposition; can we get a different APN permutation by changing the central linear layer or the inverse functions?

We made an exhaustive[8] search for an invertible matrix such that when it is used as the middle linear layer in our decomposition, the resulting S-Box is an APN permutation. All the APN permutations we found are CCZ-equivalent to the original S-Box. However not all of them are affine-equivalent to it. By

[8] Actually we optimized the search by utilizing the equivalence classes given by Theorem 2.



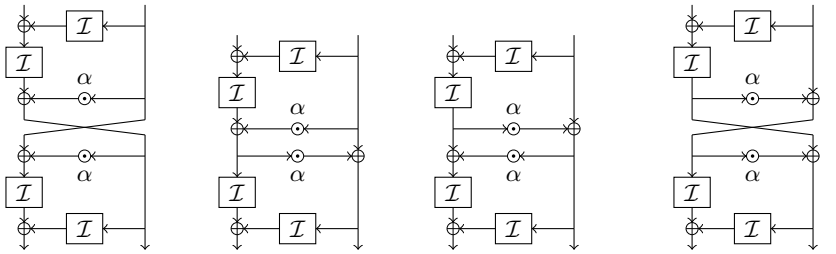
(a) The linear layer from the decomposition

(b) Applying arbitrary linear bijections a and b.

(c) Moving the linear functions a down.

Fig. 16: Propagation of the linear function a through the middle linear layer.

studying the new matrices we found that all of them can be obtained by using transformations from Theorem 2 together with swaps applied before and/or after the linear layer. All four different combinations of swaps result in four S-Boxes from distinct affine-equivalence classes (see Figure 17). However they form two pairs of EA-equivalent S-Boxes: Figure 17a and 17c, Figure 17b and 17d. The proof for EA-equivalence is given in the full version of this paper [13]. Note that the function shown in Figure 17c is the inverse of the function from Figure 17b and both functions from Figures 17a and 17d are involutions. Whether all four functions are EA-equivalent remains an open question.



(a) No swaps.

(b) Swap after.

(c) Swap before.

(d) Both swaps.

Fig. 17: Four APN permutations from different affine-equivalence classes, obtained by adding swaps before and/or after the central linear layer.

We also made an exhaustive search of all 3-bit permutations and tried to use them instead of the field inverses. A non-involutive function has to be inverted in one of the places, as in the butterfly construction we introduced in Section 3.2. It turns out that the set of all 3-bit permutations for which the respective S-Box is an APN permutation is exactly the set of all 3-bit APN permutations. It is not surprising because all 3-bit APN permutations are in the same affine equivalence class. By using Theorem 2 and by applying some outer affine transformations

we can easily replace the field inverses with arbitrary affine-equivalent functions and therefore with arbitrary 3-bit APN permutation. It follows that the two APN permutations at the top and the two APN permutations at the bottom may be different and the resulting S-Box will still be an APN permutation. We also note that one of the 3-bit APN permutations is such that its DDT and LAT are identical up to the signs in the LAT. It is the S-Box used in the block cipher 3-way [17].

As a summary of our observations we give the following theorem:

Theorem 3 (A Family of 6-bit APN Permutations). *The 6-bit permutation described by Dillon et al. in [5] is affine equivalent to the involution built using the structure described in Figure 1, where $\odot$ denotes multiplication in the finite field $GF(2^3)$, $\alpha \neq 0$ is such that $Tr(\alpha) = 0$ and $\mathcal{A}$ denotes any 3-bit APN permutation.*

3.5 Relations with the Kim and the Cube functions

It is suggested in [11] to count the number of pairs of affine permutations A, B such that $S_I = B \circ S_I \circ A$ as a measure of the symmetries inside $S_{\mathcal{I}}$. An algorithm performing this task is also provided. Using it, we have found that there are only 7 such pairs (including the pair of identity mappings). This property is preserved by affine transformations and the number could therefore be obtained without our decomposition. However, for the S-Box $S_{\mathcal{I}}$, these 7 pairs of transformations have a simple description:

$$S_{\mathcal{I}}(\lambda x, \lambda^{-1} y) = (\lambda, \lambda^{-1}) \otimes S_{\mathcal{I}}(x, y) \text{ for all } \lambda \in \mathbb{F}_{2^3}^*, \tag{2}$$

where "$\otimes$" is such that $(a, b) \otimes (c, d) = (ac, bd)$. In other words, multiplying the inputs by λ and λ^{-1} is equivalent to multiplying the outputs by the same values. As we have shown in Section 3.3, there are more symmetries *inside* the structure.

An anonymous reviewer pointed out that the observed property is quite similar to that of "Kim mapping", a non-bijective quadratic APN function from which Dillon *et al.* [5] obtained the APN permutation by applying transformations preserving CCZ-equivalence. The Kim function is defined over $\mathbb{F}_{2^6}$ as $k(x) = x^3 + x^{10} + ux^{24}$, where u is some primitive element of $\mathbb{F}_{2^6}$. It is pointed in [5] that the following holds:

$$k(\lambda x) = \lambda^3 k(x) \text{ for all } \lambda \in \mathbb{F}_{2^3}. \tag{3}$$

We found experimentally that the Kim mapping is actually affine-equivalent to all Closed Butterflies V_e^α with $n = 3, e \in \{3, 5, 6\}, Tr(\alpha) = 0$ and $\alpha \neq 0$. In particular, it is affine-equivalent to the function $Q_{\mathcal{I}} = \mathsf{V}_6^2$ described before.

The property that $k(\lambda x) = \lambda^3 k(x)$ for all $\lambda \in \mathbb{F}_{2^3}$ can be nicely translated to V_e^α structure (when $\alpha \neq 0$). Indeed, it is easy to see that the following holds:

$$\mathsf{V}_e^\alpha(\lambda x, \lambda y) = (\lambda^e, \lambda^e) \otimes \mathsf{V}_e^\alpha(x, y) \text{ for all } \lambda \in \mathbb{F}_{2^3}. \tag{4}$$

In particular, setting $e = 3$ and α such that V_e^α is affine-equivalent to the Kim mapping leads to a branch-wise variant of the property from Equation 3.

Similarly, the Open Butterflies H_e^α exhibit the following property:

$$\mathsf{H}_e^\alpha(\lambda^e x, \lambda y) = (\lambda^e, \lambda) \otimes \mathsf{H}_e^\alpha(x, y) \text{ for all } \lambda \in \mathbb{F}_{2^3}. \tag{5}$$

While V_e^α is an interesting decomposition of the Kim function (when $Tr(\alpha) = 0, \alpha \neq 0$), we also found a very similar decomposition for the cube function over $\mathbb{F}_{2^6}$, which is also a quadratic APN function. Recall that the closed butterfly V_3^α maps (x, y) to $R_{kim}(x, y)||R_{kim}(y, x)$, where $R_{kim}(x, y) = (x + \alpha y)^3 + y^3$. We have found that changing R_{kim} to $R_{cube}(x, y) = (x + \alpha y)^3 + x^3 + \alpha y^3$ leads to a function affine-equivalent to the cube function over $\mathbb{F}_{2^6}$. We describe the way we found this decomposition in the full version of this paper [13].

3.6 Univariate Polynomial Representations

In this section we describe several univariate polynomial representations of APN permutations from the affine-equivalence classes described in Section 3.4. We obtained them by interpolating the structures from previous sections in various bases relying on the field decomposition $\mathbb{F}_{2^6} \simeq (\mathbb{F}_{2^3})^2$. All polynomials described in this section are specified over $\mathbb{F}_{2^6}$ and w is a primitive element such that $w = X$ in $\mathbb{F}_2[X]/(X^6 + X^4 + X^3 + X + 1)$.

In [5], Dillon *et al.* represented the APN permutation as a univariate polynomial over $\mathbb{F}_{2^6}$ with 52 nonzero coefficients. Using our decomposition, we managed to find an APN permutation whose univariate polynomial has only 25 terms. Due to lack of space we give the polynomial in the full version of this paper [13].

Originally, the APN permutation was obtained as a composition $g = f_2 \circ f_1^{-1}$, where $f_1(x)$ and $f_2(x)$ contain 18 monomials each (as given in [5]). We have found a variant with much simpler polynomials. The function g is still an APN permutation if f_1 and f_2 as defined in [5] are replaced by the following two functions:

$$f_1(x) = w^{11}x^{34} + w^{53}x^{20} + x^8 + x,$$
$$f_2(x) = w^{28}x^{48} + w^{61}x^{34} + w^{12}x^{20} + w^{16}x^8 + x^6 + w^2x.$$

Additionally, we found a few other simple representations relying on a composition of simple polynomials. Let $g(x) = i \circ m \circ i^{-1}(x)$, then g is an APN permutation when

$$i(x) = w^{21}x^{34} + x^{20} + x^8 + x, \quad m(x) = w^{52}x^8 + w^{36}x$$

or when

$$i(x) = w^{37}x^{48} + x^{34} + w^{49}x^{20} + w^{21}x^8 + w^{30}x^6 + x, \quad m(x) = x^8.$$

In these representations, i corresponds to the sum of the two inverse functions $\mathcal{I}$ so that i and i^{-1} are the non-linear parts of the open butterfly. The function m corresponds to the central linear layer (including possible branch swaps).

4 Differentially 4-Uniform Permutations of Larger Blocks

An up to date overview of known APN functions can be found in [15]. As APN functions operating on an even number of bits are still to be found for even block sizes larger than 6, differentially 4-uniform permutations have received a lot of attention from researchers. An obvious example is the inverse function $x \mapsto x^{2^n-2}$ of $\mathbb{F}_{2^n}$ studied in the seminal work of Nyberg [4].

However, security against differential cryptanalysis is not sufficient and linear attack need to be taken into account too. The search can thus be focused on differentially 4-uniform permutations of $2n$ bits with non-linearity $2^{2n-1} - 2^n$ which is, as far as we know, the best that can be achieved. Whether there exists functions improving this bound is an open problem (Open Problem 2 in [18]). The same paper also states Open Problem 1: we must find other highly non-linear differentially 4-uniform functions operating on fields of even degree. Several papers have then presented constructions for such permutations, for example using binomials [19] or an APN permutation on $\mathbb{F}_{2^{n+1}}$ for even n [20].

In this section, we study the butterfly structure. In Section 4.1, we study butterflies with $\alpha \neq 0, 1$ and, in Section 4.2, the case $\alpha = 1$ in which the open butterfly is functionally equivalent to a 3-round Feistel Network. We show that these structures are always differentially 4-uniform for block sizes $2n$ (n odd) and have algebraic degree $n+1$ (when $\alpha \neq 1$) and n (when $\alpha = 1$) in the bijective case, 2 otherwise. While we could not prove it in the general case, we conjecture that they both have non-linearity $2^{2n-1} - 2^n$.

4.1 Butterfly with Non-Trivial α

Theorem 4 (Properties of the Butterfly Structure). *Let V_e^α and H_e^α respectively be the closed and open $2n$-bit butterflies with exponent $e = 3 \times 2^t$ for some t, coefficient α not in $\{0, 1\}$ and n odd. Then:*

- *the differential uniformity of both H_e^α and V_e^α is at most 4,*
- *V_e^α is quadratic, and*
- *half of the coordinates of H_e^α have algebraic degree n, the other half have algebraic degree $n+1$.*

Proof. In this proof, we rely a lot on the *univariate degree* of a polynomial of $\mathbb{F}_2^n$. It is different from the algebraic degree: the cube function has *univariate degree* 3 and *algebraic degree* 2.

Differential Properties. As V_e^α and H_e^α are CCZ-equivalent, they have the same differential uniformity. It is thus sufficient to prove that the one of V_e^α is at most 4. First, note that the functions V_e^α with exponent 3×2^t is affine equivalent to V_3^α which uses the exponent 3 as V_3^α can be obtained simply by applying the linear permutation $x \mapsto x^{2^{n-t}}$ on each half of the output of V_e^α. Thus, it is sufficient to study the case where the exponent is equal to 3.

Let T_α be the linear permutation of $\mathbb{F}_2^n \times \mathbb{F}_2^n$ defined by the matrix

$$T_\alpha = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix}.$$

As affine equivalence preserves differential uniformity, we will prove that the differential uniformity of $P = T_\alpha \circ \mathsf{V}_3^\alpha$ is at most equal to 4 and deduce that V_3^α has the same property. The left side of the output of P is equal to

$$\begin{aligned} P_L(x,y) &= R(x,y) + \alpha R(y,x) \\ &= (x+\alpha y)^3 + y^3 + \alpha\big((y+\alpha x)^3 + x^3\big) \\ &= x^3(1+\alpha+\alpha^4) + y^3(1+\alpha+\alpha^3) + x^2y(\alpha+\alpha^3) \end{aligned}$$

and the right side to

$$\begin{aligned} P_R(x,y) &= R(y,x) + \alpha R(x,y) \\ &= y^3(1+\alpha+\alpha^4) + x^3(1+\alpha+\alpha^3) + xy^2(\alpha+\alpha^3). \end{aligned}$$

To simplify expressions, we use the notation $\beta = \alpha^3 + \alpha$. Note that for the values of α we are interested in, namely $\alpha \neq 0, 1$, it holds that $\beta \neq 0$.

By definition of differential uniformity, the differential uniformity of P is at most 4 if and only if the following system has at most 4 solutions for any a, b, c, d (unless $a = b = 0$):

$$\begin{cases} P_L(x,y) + P_L(x+a, y+b) = c \\ P_R(x,y) + P_R(x+a, y+b) = d, \end{cases}$$

which is equivalent to

$$\begin{cases} (ax^2 + a^2x)(1+\alpha+\alpha^4) + (by^2 + b^2y)(1+\beta) + (bx^2 + a^2y)\beta = c + P_L(a,b) \\ (by^2 + b^2y)(1+\alpha+\alpha^4) + (ax^2 + a^2x)(1+\beta) + (b^2x + ay^2)\beta = d + P_R(a,b). \end{cases}$$

If $a = 0$ then the second line of the system yields the sum of a univariate degree 2 polynomial in y with $b^2\beta x$. As $b \neq 0$ (recall that $a = b = 0$ is impossible), we deduce that x is equal to a univariate degree 2 polynomial in y and replace it by this expression in the first equation. We obtain an equation with univariate degree 4 only in y with at most 4 solutions, for each of which we deduce a unique value x. Hence, the system has at most 4 solutions. The case $b = 0$ is treated similarly.

We now suppose $a \neq 0$ and $b \neq 0$. We replace the left side of the first line ℓ_1 by a linear combination of the left sides of the two equations: $\ell_1 := ab^2\ell_1 + a^2b\ell_2$. This quantity is a degree one bivariate polynomial with variables $X = ax^2 + a^2x$ and $Y = by^2 + b^2y$ so that we can write $\ell_1 = \gamma_0 X + \gamma_1 Y = \epsilon$, where ϵ is obtained by computing the same linear combination on the right side of the equations. If $\gamma_0 = 0$ then ℓ_1 actually is a degree 2 equation in y. For each of its at most 2 solutions, we obtain a degree 2 equation in x in ℓ_2 with at most 2 solutions. Hence, the total number of solutions is at most equal to 4. The case $\gamma_1 = 0$ is identical.

We now suppose $\gamma_0 \neq 0$ and $\gamma_1 \neq 0$. Using that $\gamma_0 X + \gamma_1 Y = \epsilon$, we deduce that $(ax^2 + a^2x) = \big(\epsilon + (by^2 + b^2y)\gamma_1\big)/\gamma_0$. We can therefore replace $(ax^2 + a^2x)$ by this quantity in the second equation which becomes the sum of a degree 2

equation in y with a degree 1 term in x. As before, we deduce an expression of x as a degree 2 polynomial in y and replace it by this polynomial in the other equation. Hence, the initial system has as many solutions as an equation with univariate degree 4, i.e. at most 4.

Therefore, $P(x,y)+P(x+a,y+b)=(c,d)$ has at most 4 solutions, meaning that the differential uniformity of P is at most 4.

Algebraic Degrees. As the left and right side of $\mathsf{V}_e^\alpha(x,y)$ are equal to, respectively, $(x+\alpha y)^3+y^3$ and $(y+\alpha x)^3+x^3$, it is obvious that it is quadratic (recall that the algebraic degree of the univariate polynomial $x\mapsto x^e$ of $\mathbb{F}_2^n$ is the Hamming weight of the binary expansion of e).

Consider now the open butterfly H_e^α. For the sake of simplicity, we treat the case $e=3$; other cases yield identical proofs. The right side of the output of such an open butterfly is equal to $(x+\alpha y^3)^{1/3}+\alpha y$, where $x||y$ is the input. We deduce from Theorem 1 of [21] (or equivalently from Proposition 5 of [4]) that the inverse of 3 modulo 2^n-1 for odd n is

$$1/3 \equiv \sum_{i=0}^{(n-1)/2} 2^{2i} \mod 2^n-1,$$

which implies in particular why the algebraic degree of $x\mapsto x^{1/3}$ is equal to $(n+1)/2$. We deduce from this expression that $(x+\alpha y^3)^{1/3}$ is equal to $\prod_{i=0}^{(n-1)/2}(x+\alpha y^3)^{2^{2k}}$. This sum can be developed as follows:

$$(x+\alpha y^3)^{1/3} = \sum_{J\subseteq[0,(n-1)/2]} \underbrace{\prod_{j\in J} \alpha^{2^{2j}} y^{3 2^{2j}}}_{\deg<2|J|} \underbrace{\prod_{j\in \overline{J}} x^{2^{2j}}}_{\deg=(n+1)/2-|J|},$$

where $\overline{J}$ is the complement of J in $[0,(n-1)/2]$, i.e. $J\cap\overline{J}=\emptyset$ and $J\cup\overline{J}=[0,(n-1)/2]$. The algebraic degree of each term in this sum is at most equal to $|J|+(n+1)/2$. If $\overline{J}=\emptyset$ then x is absent from the term so that the maximum algebraic degree is n. If $\overline{J}=\{j\}$ for some j, then the term is equal to $(xy^{-1})^{2^{2j}}$ (we omit the constant factor) which has algebraic degree $1+(n-1)=n$. If $|J|<(n-1)/2$, then the whole degree is smaller than n. Thus, the right side of the output has an algebraic degree equal to n.

The left side is equal to

$$\Big(y+\alpha\big((x+\alpha y^3)^{1/3}+\alpha y\big)\Big)^3+\big((x+\alpha y^3)^{1/3}+\alpha y\big)^3.$$

The terms of highest algebraic degree in this equation are of the shape $y^2(x+\alpha y^3)^{1/3}$ and $y(x+\alpha y^3)^{2\times 1/3}$. Because of what we established above, we have:

$$y^2(x+\alpha y^3)^{1/3} = \sum_{J\subseteq[0,(n-1)/2]} \underbrace{y^2\times\prod_{j\in J} \alpha^{2^{2j}} y^{3\times 2^{2j}}}_{\deg<2|J|+1} \underbrace{\prod_{j\in \overline{J}} x^{2^{2j}}}_{\deg=(n+1)/2-|J|},$$

so that the algebraic degree of this term is at most equal to $|J|+(n+1)/2+1 \leq n+1$. If $J = [0,(n-1)/2] \setminus \{j\}$ for some j, then the algebraic degree of the expression is $(1+(n-1)/2)+(n+1)/2 = n+1$, meaning that this bound is reached. The terms $y(x+\alpha y^3)^{2/3}$ are treated similarly. Hence, the left side of the output has algebraic degree $n+1$. □

This proof lead us to some interesting observations.

Remark 1. The proof relies on the study of $T_\alpha \circ \mathsf{V}_e^\alpha$ which, for $n = 3$, has as its output the concatenation of $b(x,y)$ and $b(y,x)$ for a bent function b with a Maiorana-MacFarland structure. We provide further analysis for this observation in the full version of this paper [13]. We also note that the idea of building APN or differentially 4-uniform functions by concatenating two functions, at least one of which is bent, was discussed by Carlet in [22].

We have also studied the butterfly structure experimentally. While we could not find a pair (e,α) for which a butterfly is APN for $n > 3$, we did notice a variation in the distribution of 0, 2 and 4 in their DDT. It is therefore possible that APN butterflies exist but not for $n = 5, 7$. Moreover, butterflies are never differentially 4-uniform for $n = 4, 8, 10$. However, the case $n = 6$ yields the following proposition.

Proposition 1. *If $n = 6$, then there exists α such that H_5^α is a 12-bit permutation that is differentially 4-uniform. In fact, all of the coefficients in its DDT are in $\{0,4\}$. Its non-linearity is $1920 = 2^{2n-1} - 2^{n+1}$.*

A natural generalization would be to have the same result for $e = 5$ whenever $x \mapsto x^5$ is a permutation. However, we found experimentally that this result does not hold for $n = 10$, although $x \mapsto x^5$ is a permutation of $\mathbb{F}_{2^{10}}$. We note also that, unlike in Theorem 4, Proposition 1 does not hold for all values of α but only for few of those.

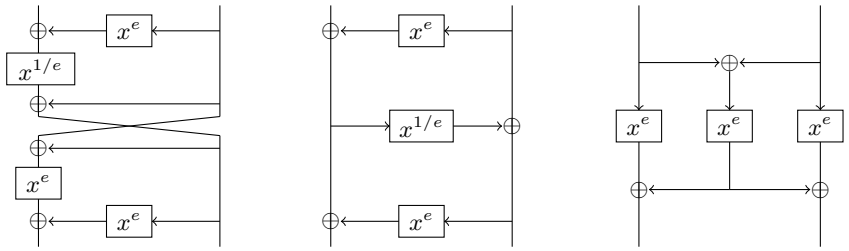
We also found experimentally that the maximum LAT coefficient of a butterfly structure operating on $2n$ bits is equal to 2^n for $n = 3, 5, 7$. This implies that the non-linearity of the butterfly structure is "optimal" in the sense that no known permutations of a field of size $2n$ have a non-linearity higher than $2^{2n-1} - 2^n$. It is however not known if this bound holds for all permutations (see Open Problem 2 in [18]).

Proposition 2. *The non-linearity of a butterfly structure operating on $2n$ bits is equal to $2^{2n-1} - 2^n$ for $n = 3, 5, 7$.*

We conjecture that this proposition is true for every odd n.

4.2 Feistel Network ($\alpha = 1$)

If we set $\alpha = 1$ in an open butterfly structure, the resulting permutation is functionally equivalent to a 3-round Feistel Network with round functions $x \mapsto x^e$, $x \mapsto x^{1/e}$ and $x \mapsto x^e$, as described in Figure 18. We denote such a Feistel



(a) Open butterfly H_e^1. (b) F^e (note $\mathsf{F}^e = \mathsf{H}_e^1$). (c) Closed butterfly V_e^1.

Fig. 18: The equivalence between H_e^1 and F^e.

Network F^e. We note that the closed butterfly V_e^1 has a structure reminiscent of a Lai-Massey round (see Figure 18c).

In [23], Li and Wang proved that the $2n$-bit Feistel Networks F^e with $e = 2^k+1$ and odd n such that $gcd(n,k) = 1$ have very good cryptographic properties:

1. the differential spectrum of F^e is equal to $\{0, 4\}$;
2. the non-linearity of F^e is the best known and is equal to $2^{2n-1} - 2^n$;
3. the algebraic degree of F^e is equal to n.

Note that the butterfly structures from Theorem 4 have degree $n+1$ on half of the coordinates. We have proved that F^3 has degree n on all coordinates. The proof is given in the full version of this paper [13].

Remark 2. The proof for the algebraic degree of the left side of F^3 relies on particular cancellation occurring in the sum $y^2(x+y^3)^{1/3} + y(x+y^3)^{2/3}$. Such cancellations do not occur when $\alpha \neq 1$ as the terms in the corresponding sum are preceded by different coefficients which are both functions of α. This explains why the algebraic degree of F^3 and the open butterfly structure with $\alpha \neq 1$ are different.

We also note that the monomial $x \mapsto x^5$ in $\mathbb{F}_{2^{2n}}$ shares the same differential and linear properties. In [23] it is mentioned that for $n = 3$ the Feistel Network F^3 is CCZ-equivalent to the monomial $x \mapsto x^5$. We observe that the closed butterfly V_5^1, which is CCZ-equivalent to F^5, is actually linear-equivalent to the monomial $x \mapsto x^5$ over $\mathbb{F}_{2^{2n}}$ for all odd $n \geq 3$. We state the generalized result in the following theorem.

Theorem 5. *Let $n \geq 3$ be an odd integer and $e = 2^{2k} + 1$ for some positive integer k. Then the closed $2n$-bit butterfly V_e^1 is linear-equivalent to the monomial $x \mapsto x^e$ of $\mathbb{F}_{2^{2n}}$.*

Corollary 1. *Let $n \geq 3$ be an odd integer and $e = 2^{2k} + 1$ for some positive integer k, such that the monomial $x \mapsto x^e$ defines a permutation of $\mathbb{F}_{2^{2n}}$. Then the $2n$-bit Feistel Network F^e is CCZ-equivalent to this permutation.*

The proof is based on the field decomposition $\mathbb{F}_{2^{2n}} \simeq (\mathbb{F}_{2^n})^2$ and is given it in the full version of this paper [13].

5 Implementing 6-bit APN Permutations

We can use the open butterfly structure to efficiently implement 6-bit APN permutations in both a bit-sliced fashion for use in software and in hardware. In this section, we explore this idea and provide an S-Box A_o which is affine equivalent to H_3^2 and for which there exists such efficient implementation.

5.1 Efficient Bit-Sliced Implementations

Starting from the algebraic normal forms of the operations used to compute H_3^2, it is easy to write a first naive bitsliced implementation (see full version [13]).

This implementation can be optimized by using Boolean algebra and removing the linear component of $x \mapsto x^3$ in the first and last steps. Doing this is equivalent to applying an affine permutation before and after the H_3^2 to obtain a new permutation A_o. This operation preserves the differential and linear property of the permutation while also keeping the property that $A_o^{-1} = \mathsf{Swap}_6 \circ A_o \circ \mathsf{Swap}_6$, where Swap_6 simply swaps the two 3-bit branches. The bitsliced implementation of this simplified S-Box is given in Algorithm 1 and its look-up table in Table 5.

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	0	1d	6	3f	3c	3b	31	12	22	35	17	2c	16	33	30	39
1.	2d	a	38	2b	1	4	2f	1e	3	34	2e	25	27	1a	29	28
2.	2a	7	14	3d	36	19	b	20	3e	d	37	8	1b	2	9	1c
3.	10	1f	21	3a	26	13	24	5	c	f	11	e	23	32	15	18

Table 5: The look-up table of A_o in hexadecimal, e.g. $A_o(\texttt{0x32}) = \texttt{0x21}$.

Algorithm 1 An optimised bitsliced implementation of an S-Box affine-equivalent to the open butterfly with $\alpha = 2$, $e = 3$.

function $A_o(X_0, ..., X_5)$
1. $t = (X_5 \wedge X_3)$
2. $X_0 \oplus= t \oplus (X_5 \wedge X_4)$
3. $X_1 \oplus= t$
4. $X_2 \oplus= (X_4 \vee X_3)$
5. $t = (X_1 \vee X_0)$
6. $X_0 \oplus= (X_2 \wedge X_1) \oplus X_4$
7. $X_1 \oplus= (X_2 \wedge X_0) \oplus X_5 \oplus X_3$
8. $X_2 \oplus= t \oplus X_3$
9. $X_3 \oplus= X_1$
10. $X_4 \oplus= X_2 \oplus X_0$
11. $X_5 \oplus= X_0$
12. $u = X_3$
13. $t = X_4$
14. $X_3 \oplus= t$
15. $X_3 = X_3 \wedge X_5 \oplus t$
16. $X_4 \oplus= ((\neg X_5) \wedge u)$
17. $X_5 \oplus= (t \vee u)$
18. $t = (X_2 \wedge X_0)$
19. $X_3 \oplus= t \oplus (X_2 \wedge X_1)$
20. $X_4 \oplus= t$
21. $X_5 \oplus= (X_1 \vee X_0)$

end function

5.2 Hardware Implementation

Our decompositions also eases the hardware implementation of these S-Boxes. To illustrate this, we simulated the circuit computing these functions in three different ways. First, we simply gave the look-up table to the software[9] and let it find the best implementation it could (*no decomposition* case). Then, we fed it our decomposition of the different structures (*decomposed* case).

The optimization performed by the software is done for two competing criteria. The first is the area which simply corresponds to the physical space needed to implement the circuit using the logical gates available. The second is the propagation time, i.e. the delay necessary for the electronic signal to go through the circuit implementing the S-Box and to stabilize itself to the output value.[10]

For each function, we repeated the experience several times using different periods for the clock cycles: when the period is maximum, priority is given to optimizing the area and, as the period decreases, the priority shifts toward the propagation time. The results are given in Table 6.

		Base			**Decomposed**		
S-Box	Period (ns)	a	d	$a \times d$	a	d	$a \times d$
H_3^2	100	799	56.42	45 079.58	414	39.31	16 274.34
	20	827	19.75	16 333.25	404	18.7	7554.8
	10	928	9.81	9103.68	431	9.76	4206.56
	5	1062	4.81	5108.22	569	4.81	2736.89
A_o	100	774	53.13	41 122.62	384	42.01	16 131.84
	20	812	19.3	15 671.6	384	15.43	5925.12
	10	869	9.63	8368.47	382	9.77	3732.14
	6	1041	5.8	6037.8	464	5.8	2691.2

Table 6: Results on the hardware implementation of our S-Boxes. The area a is in $(\mu m)^2$, the delay d is in ns and $a \times d$ is their product.

As we can see, the knowledge of the decompositions always allows a more efficient implementation: regardless of what the main optimisation criteria is, both the area and the delay are decreased.

6 Conclusion

We have identified a decomposition of the 6-bit APN permutation published by Dillon *et al.* [5] and found it to be affine equivalent to an involution. We

[9] We used the digital cell library `SAED90n-1P9M` in the "normal V_t, high temperature, nominal voltage" corner.

[10] We also considered implementing the cube function using finite field arithmetic but could not easily improve our results.

generalized the structure found to larger block sizes, although we could only prove its being differentially 4-uniform in those cases. We also deduced efficient implementation of 6-bit APN permutations in both a bit-sliced fashion and in hardware.

Our work also raised the following open questions.

Open Problems (On the properties of Butterfly Structures).

1. Is there a tuple n, e, α where $n > 3$ and e are integers, and α is a finite field element such that H_e^α operating on $(\mathbb{F}_{2^n})^2$ is APN?
2. Is it true that the non-linearity of a butterfly structure on $2n$ bits with $\alpha \neq 0, 1$ and n odd is always $2^{2n-1} - 2^n$?

7 Acknowledgements

We thank the anonymous reviewers for their helpful comments. We also thank Yann Le Corre for studying the hardware implementation of the permutation. The work of Léo Perrin is supported by the CORE ACRYPT project (ID C12-15-4009992) funded by the *Fonds National de la Recherche* (Luxembourg). The work of Aleksei Udovenko is supported by the *Fonds National de la Recherche*, Luxembourg (project reference 9037104).

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY **4**(1) (1991) 3–72
2. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Advances in Cryptology – EUROCRYPT'93, Springer (1994) 386–397
3. Daemen, J., Rijmen, V.: The design of Rijndael: AES-the advanced encryption standard. Springer (2002)
4. Nyberg, K.: Differentially uniform mappings for cryptography. In: Advances in cryptology — Eurocrypt'93, Springer (1994) 55–64
5. Browning, K., Dillon, J., McQuistan, M., Wolfe, A.: An APN permutation in dimension six. Finite Fields: theory and applications **518** (2010) 33–42
6. Bilgin, B., Bogdanov, A., Knežević, M., Mendel, F., Wang, Q.: Fides: Lightweight Authenticated Cipher with Side-Channel Resistance for Constrained Hardware. In: Cryptographic Hardware and Embedded Systems - CHES 2013: 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 142–158
7. Biryukov, A., Perrin, L., Udovenko, A.: Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1. In: Advances in Cryptology-Eurocrypt 2016, Springer (2016) 372–402
8. Biryukov, A., Perrin, L.: On Reverse-Engineering S-Boxes with Hidden Design Criteria or Structure. In Gennaro, R., Robshaw, M., eds.: Advances in Cryptology – CRYPTO 2015. Volume 9215 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2015) 116–140

9. Bogdanov, A., Leander, G., Nyberg, K., Wang, M.: Integral and multidimensional linear distinguishers with correlation zero. In Wang, X., Sako, K., eds.: Advances in Cryptology – ASIACRYPT 2012. Volume 7658 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 244–261
10. Biryukov, A., Shamir, A.: Structural cryptanalysis of SASAS. In Pfitzmann, B., ed.: Advances in Cryptology – EUROCRYPT 2001. Volume 2045 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 395–405
11. Biryukov, A., De Cannière, C., Braeken, A., Preneel, B.: A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms. In Biham, E., ed.: Advances in Cryptology — EUROCRYPT 2003. Volume 2656 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 33–50
12. Developers, T.S.: SageMath, the Sage Mathematics Software System (Version 7.1). (2016) `http://www.sagemath.org`.
13. Perrin, L., Udovenko, A., Biryukov, A.: Cryptanalysis of a Theorem: Decomposing the Only Known Solution to the Big APN Problem (Full Version). Cryptology ePrint Archive, Report 2016/539 (2016) http://eprint.iacr.org/.
14. Carlet, C., Charpin, P., Zinoviev, V.: Codes, bent functions and permutations suitable for DES-like cryptosystems. Designs, Codes and Cryptography **15**(2) (1998) 125–156
15. Blondeau, C., Nyberg, K.: Perfect nonlinear functions and cryptography. Finite Fields and Their Applications **32** (2015) 120 – 147 Special Issue : Second Decade of {FFA}.
16. Budaghyan, L., Carlet, C., Pott, A.: New classes of almost bent and almost perfect nonlinear polynomials. IEEE Transactions on Information Theory **52**(3) (2006) 1141–1152
17. Daemen, J., Govaerts, R., Vandewalle, J.: A new approach to block cipher design. In: Fast Software Encryption: Cambridge Security Workshop Cambridge, U. K., December 9–11,1993 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1994) 18–32
18. Bracken, C., Leander, G.: A highly nonlinear differentially 4 uniform power mapping that permutes fields of even degree. Finite Fields and Their Applications **16**(4) (2010) 231–242
19. Bracken, C., Tan, C.H., Tan, Y.: Binomial differentially 4 uniform permutations with high nonlinearity. Finite Fields and Their Applications **18**(3) (2012) 537–546
20. Li, Y., Wang, M.: Constructing differentially 4-uniform permutations over GF(2^{2m}) from quadratic APN permutations over GF(2^{2m+1}). Designs, Codes and Cryptography **72**(2) (2014) 249–264
21. Kyureghyan, G.M., Suder, V.: On inverses of APN exponents. In: Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on, IEEE (2012) 1207–1211
22. Carlet, C.: Relating three nonlinearity parameters of vectorial functions and building APN functions from bent functions. Designs, Codes and Cryptography **59**(1) (2011) 89–109
23. Li, Y., Wang, M.: Constructing S-boxes for Lightweight Cryptography with Feistel Structure. In: CHES, Springer (2014) 127–146